

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ УЗБЕКИСТАН
НАМАНГАНСКИЙ ИНЖЕНЕРНО-
СТРОИТЕЛЬНЫЙ ИНСТИТУТ

О.С. Абдуллаева

**ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ
В СТРОИТЕЛЬСТВЕ
(ЧАСТЬ 2)**

Учебник

*Ответственный редактор:
доктор педагогических наук (DSc), профессор У. Аскарова*

RU
science
RU-SCIENCE.COM

Москва
2022

УДК
ББК

И74

*Рекомендовано к изданию решением Совета
Наманганского инженерно-строительного института от 31.05.2022 №12*

Рецензенты:

М. Якубов профессор ТУИТ, д-р техн. наук (DSc),

М.Т. Тухтасинов, доцент кафедры «Информационные технологии»
НамИСИ, канд. техн. наук

Ответственный редактор:

У. Аскарлова, д-р пед. наук (DSc), проф.

И74 Информационные технологии в строительстве (Ч. 2) :
учебник / О.С. Абдуллаева ; отв. ред. У. Аскарлова. — Москва :
РУСАЙНС, 2022. — 190 с.

ISBN 978-5-466-02022-9

Основная цель учебника оказать помощь обучающимся в приобретении знаний и выработке рефлексивных умений по курсу «Информационные технологии в строительстве». В нём приводятся теоретические и практические основы интегрированных информационных технологий.

Особое внимание посвящено актуальным проблемам основ предмета информационные технологии, вопросам безопасности информации и способам защиты информации; алгоритмизации и программированию информационных процессов; компьютерной графики, трехмерного моделирования.

Учебник может представлять интерес для широкого круга специалистов, преподавателей.

Ключевые слова: политическая кампания; политические технологии; политическая коммуникация; интернет-технологии; информационно-коммуникационное пространство; политическая стабильность.

УДК
ББК

ISBN 978-5-466-02022-9

© Абдуллаева О.С., 2022

© ООО «РУСАЙНС», 2022



Абдуллаева Озода Сафибуллаевна является профессором кафедры «Информационные технологии в технических системах», доктор философии по педагогическим наукам (PhD). На сегодняшний день ею опубликовано более 170 научных работ. За последние 3 года 25 из 51 научной статьи опубликованы в зарубежных изданиях, в том числе 6 компьютерных программ, 7 монографий, 4 учебника, 24 учебных пособий. В 2018 году стала победителем конкурса «Лучший педагог высшего образовательного учреждения». Является победителем конкурса «Лучший молодой ученый 2020», организованный Ассоциацией молодых ученых (Казахстан, г. Нур-Султан) и награждена медалью. В 2020 году награждена медалью Альфреда Нобеля со стороны Академии естественных и точных наук Российской Федерации. Награждена золотой медалью на Международной выставке женщин-изобретателей KIWIE-2020 в Южной Корее. Награждена бронзовой медалью на Международной выставке женщин-изобретателей KIWIE-2021 в Южной Корее.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
ГЛАВА I. БЕЗОПАСНОСТЬ ИНФОРМАЦИИ И СПОСОБЫ ЗАЩИТЫ ИНФОРМАЦИИ	7
Тема 1.1: Безопасность информации и способы защиты информации в сфере архитектуры и строительства	7
Тема 1.2. Алгоритмизация и программирование информационных процессов в сфере архитектуры и строительства.....	19
ГЛАВА II. ИНФОРМАЦИОННЫЕ СИСТЕМЫ	40
Тема 2.1. Информационные системы в архетиктуре и строительстве.....	40
Тема 2.2. Методы проектирования и построения информационных систем.....	44
Тема 2.3. Система электронного документооборота.....	56
ГЛАВА III. КОМПЬЮТЕРНАЯ ГРАФИКА.....	63
Тема 3.1. Понятие компьютерной графики и их виды	63
Тема 3.2. Введение в программу Corel DRAW. Возможности программы и её интерфейс	70
Тема 3.3. Автоматизированные системы проектирования в строительстве. Общие сведения программы AutoCAD.....	77
Тема 3.4. Строение трехмерных объектов, команды для их изменения	102
Тема 3.5. Автоматизация строительных конструкций (Lira).....	130
ГЛАВА IV. ТРЕХМЕРНОЕ МОДЕЛИРОВАНИЕ.....	143
Тема 4.1. Способы трехмерного моделирования на компьютере. Введение программы 3D Studio MAX.....	143
Тема 4.2. Редактор материалов. Источник света. Установка камер. Основы анимации. Рендер	167
Тема 4.3. Системы проектирования информационных моделей строительных объектов	179
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	189

ВВЕДЕНИЕ

Характерной чертой нашего времени являются интенсивно развивающиеся процессы информатизации практически во всех сферах человеческой деятельности. Они привели к формированию новой информационной инфраструктуры, которая связана с новым типом общественных отношений, с новой реальностью, с новыми информационными технологиями различных видов деятельности.

В данном учебнике рассматриваются основные теоретические и практические аспекты дисциплины «Информационные технологии в строительстве». Целью обучения данного курса является освоение обучающимися технологического подхода к информационной деятельности как способа её теоретического осмысления и практического внедрения информационных технологий, как в сферы строительства, так и в общественной жизни.

Основными задачами при изучении дисциплины являются формирование следующих профессиональных компетенций:

- способность к проектированию базовых и прикладных информационных технологий;
- способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, технические и программные).

Учебник разработан в полном соответствии с учебной программой дисциплины «Информационные технологии в строительстве».

Структура издания ориентирована на системное изложение учебного материала 2 семестра обучения данного курса. В первой главе «Безопасность информации. Алгоритмизация и программирование информационных процессов» излагается информация о безопасности информации и способах защиты информации; алгоритмизации и программировании информационных процессов; современные технологии программирования; основные элементы языков программирования; технологии визуального программирования в архитектуре и строительстве. Во второй главе «Информационные системы» представлены сведения об информационных системах в архитектуре и строительстве, способах проектирования и построения информационных систем, системе электронного документооборота. В третьей главе «Компьютерная графика» рассмотрены основные понятия компьютерной графики и их видах; введение в программу Corel DRAW, возможности программы; общие сведения программы AutoCAD. В четвёртой главе «Трёхмерное моделирование» изложены сведения о строении трёхмерных объектов, команды для их изменения; рендер; этапы проектирования чер-

тежей строительных объектов; способы трехмерного моделирования на компьютере; введение в программу 3D Studio MAX; редактор материалов; источник света, установка камер, основы анимации. В конце каждого параграфа представлены вопросы для самоконтроля. В издании имеется список использованной литературы, на которую в тексте имеются ссылки. В конце каждого параграфа представлены вопросы для самоконтроля. В издании имеется список использованной литературы, на которую в тексте имеются ссылки.

ГЛАВА I. БЕЗОПАСНОСТЬ ИНФОРМАЦИИ И СПОСОБЫ ЗАЩИТЫ ИНФОРМАЦИИ

Тема 1.1: Безопасность информации и способы защиты информации в сфере архитектуры и строительства

Основные модули

- Понятие безопасности информации.
- Политика информационной безопасности.
- Технические и программные средства защиты информации.
- Способы защиты информации. Идентификация и аутентификация. Несанкционированное подключение к компьютерной сети, корыстные действия и нарушение правил работы в сетях.
- Компьютерные вирусы и их виды. Защита от компьютерных вирусов
- Защита от компьютерных вирусов

Информационной безопасностью называют комплекс организационных, технических и технологических мер по защите информации от неавторизованного доступа, разрушения, модификации, раскрытия и задержек в доступе.

Информационная безопасность дает гарантию того, что достигаются следующие цели:

- **конфиденциальность** информации (свойство информационных ресурсов, в том числе информации, связанное с тем, что они не станут доступными и не будут раскрыты для неуполномоченных лиц);
- **целостность** информации и связанных с ней процессов (неизменность информации в процессе ее передачи или хранения);
- **доступность** информации, когда она нужна (свойство информационных ресурсов, в том числе информации, определяющее возможность их получения и использования по требованию уполномоченных лиц);
- **учет** всех процессов, связанных с информацией.

Обеспечение **безопасности информации** складывается из трех составляющих:

- Конфиденциальности,
- Целостности,

- Доступности.

Точками приложения процесса защиты информации к информационной системе являются:

- аппаратное обеспечение,
- программное обеспечение
- обеспечение связи (коммуникации).

Сами процедуры (механизмы) защиты разделяются на

- защиту физического уровня,
- защиту персонала
- организационный уровень.

Угроза безопасности компьютерной системы - это потенциально возможное происшествие (преднамеренное или нет), которое может оказать нежелательное воздействие на саму систему, а также на информацию, хранящуюся в ней.

Политика безопасности - это комплекс мер и активных действий по управлению и совершенствованию систем и технологий безопасности, включая информационную безопасность.

Виды информационных угроз

Организационная защита

- *организация режима и охраны.*
- *организация работы с сотрудниками* (подбор и расстановка персонала, включая ознакомление с сотрудниками, их изучение, обучение правилам работы с конфиденциальной информацией, ознакомление с мерами ответственности за нарушение правил защиты информации и др.)
- *организация работы с документами* и документированной информацией (разработка, использование, учет, исполнение, возврат, хранение и уничтожение документов и носителей конфиденциальной информации)
- *организация использования технических средств* сбора, обработки, накопления и хранения конфиденциальной информации;
- *организация работы по анализу внутренних и внешних угроз* конфиденциальной информации и выработке мер по обеспечению ее защиты;
- *организация работы по проведению систематического контроля за работой персонала* с конфиденциальной информацией, порядком учета, хранения и уничтожения документов и технических носителей.

Технические средства защиты информации

Для защиты периметра информационной системы создаются:

- системы охранной и пожарной сигнализации;
- системы цифрового видео наблюдения;
- системы контроля и управления доступом (СКУД).

Защита информации от ее утечки техническими каналами связи обеспечивается следующими средствами и мероприятиями:

- использованием экранированного кабеля и прокладка проводов и кабелей в экранированных конструкциях;
- установкой на линиях связи высокочастотных фильтров;
- построение экранированных помещений («капсул»);
- использование экранированного оборудования;
- установка активных систем шумления;
- создание контролируемых зон.

Аппаратные средства защиты информации

- Специальные регистры для хранения реквизитов защиты: паролей, идентифицирующих кодов, грифов или уровней секретности;
- Устройства измерения индивидуальных характеристик человека (голоса, отпечатков) с целью его идентификации;
- Схемы прерывания передачи информации в линии связи с целью периодической проверки адреса выдачи данных.
- Устройства для шифрования информации (криптографические методы).
- Системы бесперебойного питания:
 - Источники бесперебойного питания;
 - Резервирование нагрузки;
 - Генераторы напряжения.

Программные средства защиты информации

- Межсетевые экраны.
- Криптографические средства:
 - Шифрование;
 - Цифровая подпись.
- Системы резервного копирования.
- Системы аутентификации:
 - Пароль;
 - Ключ доступа (физический или электронный);
 - Сертификат;
 - Биометрия.

- Инструментальные средства анализа систем защиты:
 - Мониторинговый программный продукт (рис 1.1.1.).

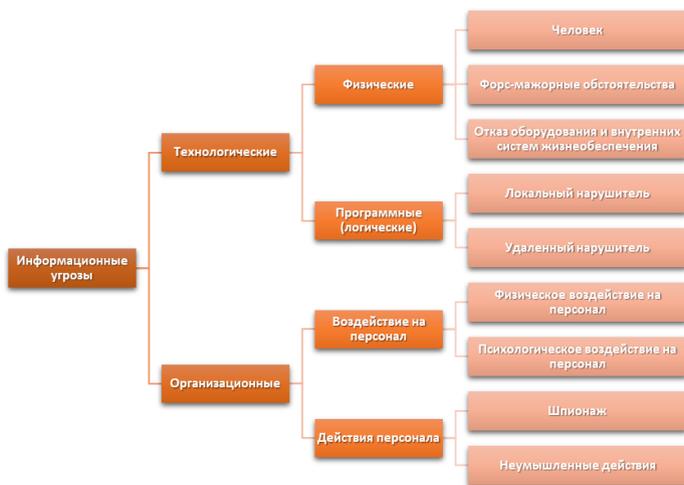


Рис 1.1.1. Виды информационных угроз

Понятие компьютерного вируса

Компьютерный вирус – это специально написанная, как правило, небольшая по размерам программа, которая может записывать (внедрять) свои копии (возможно, измененные) в компьютерные программы, расположенные в исполнимых файлах, системных областях дисков, драйверах, документах и т.д., причем эти копии сохраняют возможность к «размножению».

Процесс внедрения вирусом своей копии в другую программу (системную область диска и т.д.) называется заражением, а программа или объект, содержащий вирус – зараженным.

Компьютерные вирусы являются программами, которые могут «размножаться» и скрытно внедрять свои копии в файлы, загрузочные секторы дисков и документы.

Обязательным свойством компьютерного вируса является способность к размножению (самокопированию) и незаметному для пользователя внедрению в файлы, загрузочные секторы дисков и документы. Название «вирус» по отношению к компьютерным программам пришло из биологии именно по признаку способности к саморазмножению.

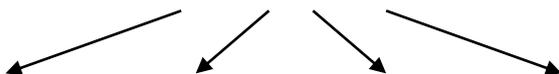
Классификация компьютерных вирусов

- по среде обитания;
- по операционным системам;
- по алгоритму работы;
- по деструктивным возможностям.

По операционным системам

Для каждой операционной системы создаются свои вирусы, которые будут «работать» только в ней. Но существуют и универсальные вирусы, которые способны внедряться в различные операционные системы.

По среде обитания



Файловые вирусы

Наносят вред файлам.
Создают файл-двойник с именем оригинала.

Загрузочные вирусы

внедряются в загрузочный сектор диска.
Операционная система при этом загружается с ошибками и сбоями

Макро-вирусы

«Портят» документы Word, Excel и других прикладных программ операционной системы Windows.

Сетевые вирусы

Распространяются по Internet через электронные письма или после посещения сомнительных сайтов.

По алгоритму работы



Резидентность

Вирусы, обладающие этим свойством действуют постоянно пока компьютер включен.

Самошифрование и полиморфизм

Вирусы-полиморфики изменяют свой код или тело программы, что их трудно обнаружить.

Стелс-алгоритм

Вирусы-невидимки «прячутся» в оперативной памяти и антивирусная программа их не может обнаружить.

Нестандартные приемы

Принципиально новые методы воздействия вируса на компьютер.

По деструктивным возможностям



Безвредные

не наносят никакого вреда

Неопасные

наносит моральный ущерб пользователю

Опасные

уничтожают информацию в файлах.

Очень опасные

сбивают процесс загрузки ОС, после чего требуется ее переустановка; или «портят» винчестер, что его требуется форматировать

Вредоносные программы

- *Троянский конь* - это программа, содержащая в себе некоторую разрушающую функцию, которая активизируется при наступлении некоторого условия срабатывания. Обычно такие программы маскируются под какие-нибудь полезные утилиты. Виды деструктивных действий:
 - Уничтожение информации. (Конкретный выбор объектов и способов уничтожения зависит только от фантазии автора такой программы и возможностей ОС. Эта функция является общей для троянских коней и закладок).
 - Перехват и передача информации. (паролей, набираемых на клавиатуре).
 - Целенаправленное изменение программы.
- *Червями* называют вирусы, которые распространяются по глобальным сетям, поражая целые системы, а не отдельные программы. Это самый опасный вид вирусов, так как объектами нападения в этом случае становятся информационные системы государственного масштаба. С появлением глобальной сети Internet этот вид нарушения безопасности представляет наибольшую угрозу, т.к. ему в любой момент может подвергнуться любой из компьютеров, подключенных к этой сети. Основная функция вирусов данного типа – взлом атакуемой системы, т.е. преодоление защиты с целью нарушения безопасности и целостности.

Защита от компьютерных вирусов

Для обнаружения, удаления и защиты от компьютерных вирусов разработано несколько видов специальных программ, которые позволяют обнаруживать и уничтожать вирусы. Такие программы называются антивирусными.

Виды антивирусных программ

- *Детекторы* позволяют обнаруживать файлы, заражённые одним из нескольких известных вирусов. Некоторые программы-детекторы также выполняют эвристический анализ файлов и системных областей дисков, что часто (но отнюдь не всегда) позволяет обнаруживать новые, не известные программедетектору, вирусы.

- **Фильтры** - это резидентные программы, которые оповещают пользователя о всех попытках какой-либо программы записаться на диск, а уж тем более отформатировать его, а также о других подозрительных действиях.
- **Программы-доктора или фаги** не только находят зараженные вирусами файлы, но и «лечат» их, т.е. удаляют из файла тело программы-вируса, возвращая файлы в исходное состояние.
- **Ревизоры** запоминают сведения о состоянии файлов и системных областей дисков, а при последующих запусках – сравнивают их состояние исходным. При выявлении несоответствий об этом сообщается пользователю.
- **Сторожа или фильтры** располагаются резидентно в оперативной памяти компьютера и проверяют на наличие вирусов запускаемые файлы и вставляемые USB-накопители.
- **Программы-вакцины или иммунизаторы** модифицируют программы и диски таким образом, что это не отражается на работе программ, но тот вирус, от которого производится вакцинация, считает эти программы или диски уже заражёнными.

Недостатки антивирусных программ

- Ни одна из существующих антивирусных технологий не может обеспечить полной защиты от вирусов.
- Антивирусная программа забирает часть вычислительных ресурсов системы, нагружая центральный процессор и жёсткий диск. Особенно это может быть заметно на слабых компьютерах.
- Антивирусные программы могут видеть угрозу там, где её нет (ложные срабатывания).
- Антивирусные программы загружают обновления из Интернета, тем самым расходуя трафик.
- Различные методы шифрования и упаковки вредоносных программ делают даже известные вирусы не обнаруживаемыми антивирусным программным обеспечением. Для обнаружения этих «замаскированных» вирусов требуется мощный механизм распаковки, который может дешифровать файлы перед их проверкой. Однако во многих антивирусных программах эта возможность отсутствует и, в связи с этим, часто невозможно обнаружить зашифрованные вирусы.

Установка, обновление базы ESET NOD32, проверка памяти с ее помощью, очистка вирусов.

Для установки антивируса ESET NOD32 необходимо нажать на exe файл. Для соглашения лицензионного соглашения необходимо нажать на галочку на «Я принимаю условия лицензионного соглашения» и нажать на кнопку «Далее» (рис.1.3.1.-1.3.4.).

В данному диалоговом окне необходимо ввести «Имя пользователя» и «Пароль» для автоматического обновления, или нажать на «Установить параметры обновления позже» затем нажать на кнопку «Далее». В результате установится программа и появится ее логотип.

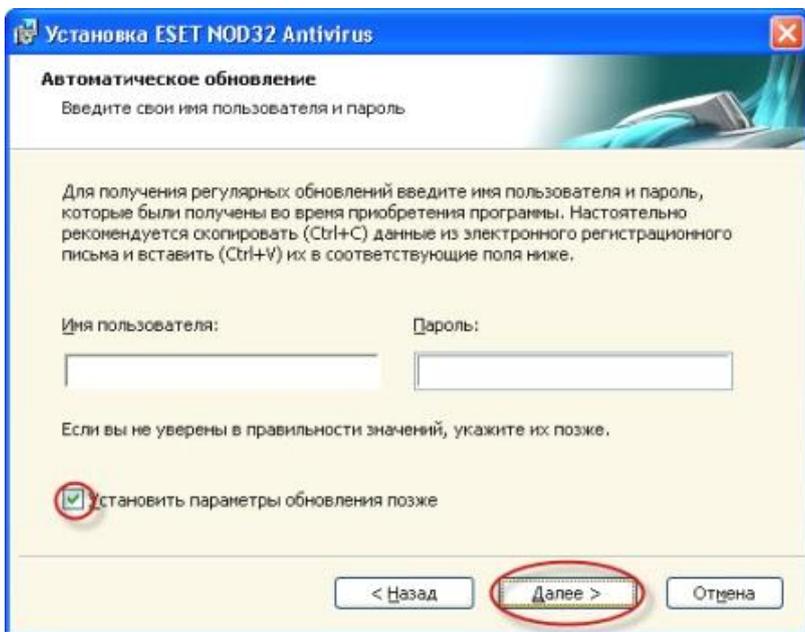


Рис.1.3.1. Установка ESET NOD32

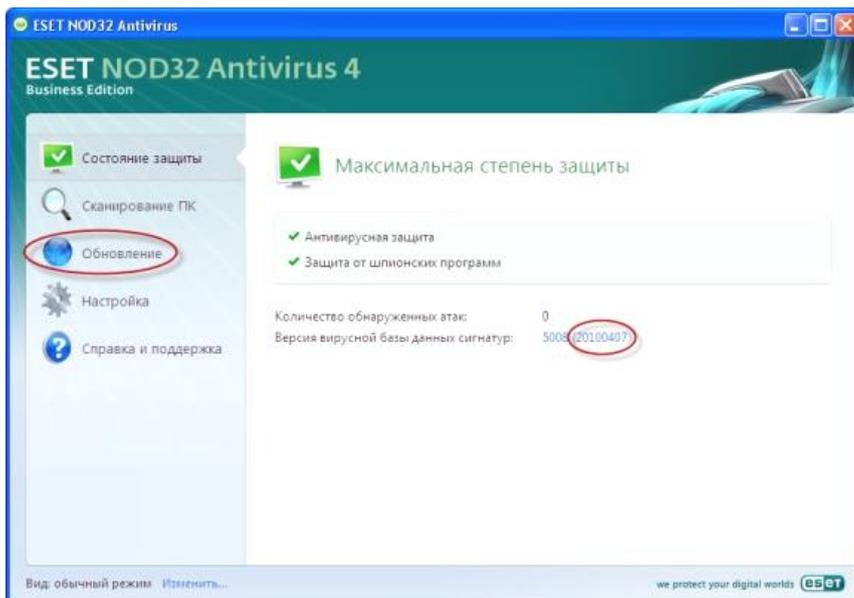


Рис.1.3.2. Установка ESET NOD32

Для обновления базы необходимо на рабочем столе нажать на логотип ESET NOD32 или выполнить действия «Пуск»-«Все программы»-«ESET»-«ESET NOD32 Antivirus»-«ESET NOD32 Antivirus».

В открывшемся окне в разделе “Обновлении” появятся данные о последнем обновлении программы.

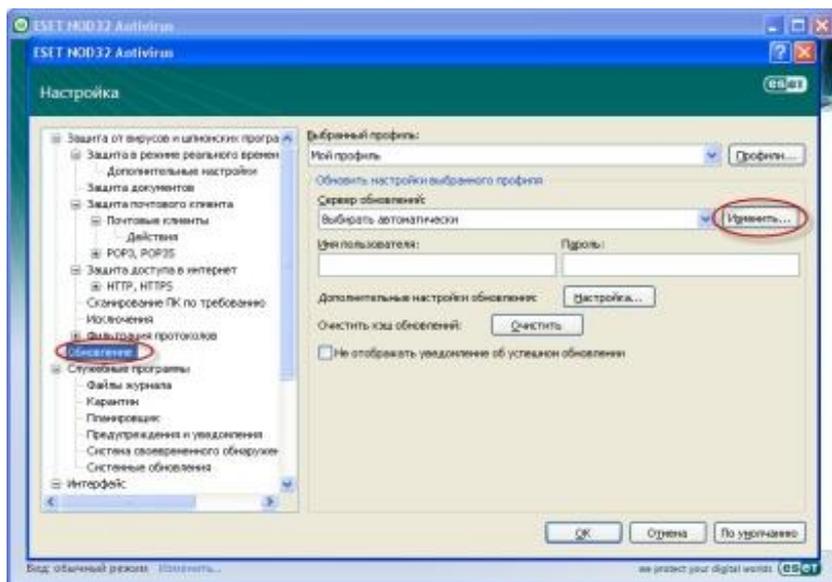


Рис.1.3.3. Установка ESET NOD32

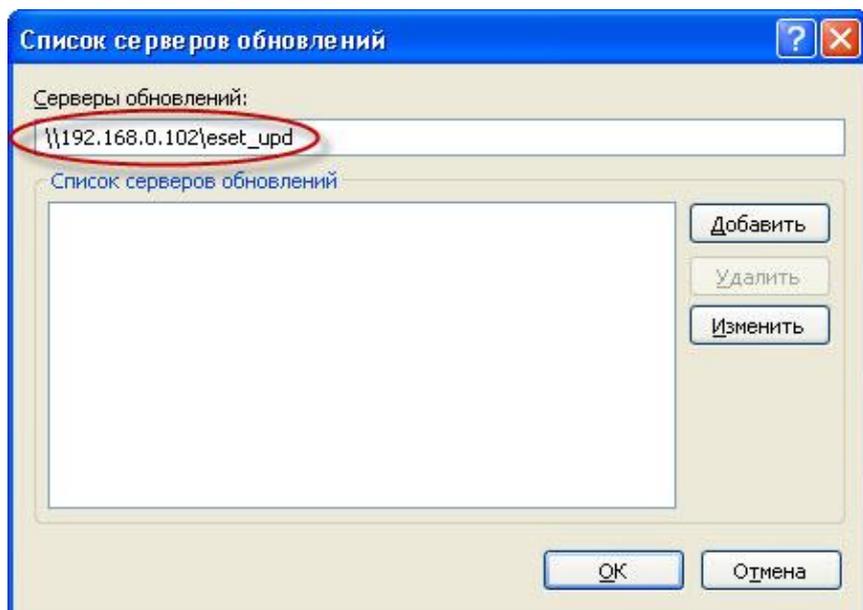


Рис.1.3.4. Установка ESET NOD32

Нажмите на кнопку F5. В открывшемся окне необходимо указать расположение базы для антивируса. Для этого необходимо нажать на кнопку «Изменить».

Для проверки (fleshки, жесткого диска, чипов) необходимо нажать на правую кнопку мыши и выбрать «Расширенные функции/очистка файлов».

В данному диалоговом окне необходимо ввести “Имя пользователя” и “Пароль” для автоматического обновления, или нажать на “Установить параметры обновления позже” затем нажать на кнопку “Далее”. В результате установится программа и появится ее логотип.

Вопросы для самоконтроля

1. Понятие безопасности информации.
2. Политика информационной безопасности.
3. Технические и программные средства защиты информации.
4. Способы защиты информации. Идентификация и аутентификация. Несанкционированное подключение к компьютерной сети, корыстные действия и нарушение правил работы в сетях.
5. Компьютерные вирусы и их виды.
6. Защита от компьютерных вирусов.
7. Средства обеспечения безопасности информации в операционных системах
8. Средства обеспечения безопасности компьютерных сетей.
9. Внутренние и внешние воздействия в информатизированных отраслях

Тема 1.2. Алгоритмизация и программирование информационных процессов в сфере архитектуры и строительства

Основные модули

- Виды и методы создания алгоритма.
- Графический способ записи алгоритмов.
- Математическая модель
- Этапы решения задач на компьютере.

Понятие алгоритма такое же основополагающее для информатики, как и понятие информации. Именно поэтому важно в нем разобраться.

Название "**алгоритм**" произошло от латинской формы имени величайшего среднеазиатского математика **Мухаммеда ибн Муса ал-Хорезми** (Algorithmi), жившего в 783—850 гг.

Человек ежедневно встречается с необходимостью следовать тем или иным правилам, выполнять различные инструкции и указания. Например, переходя через дорогу на перекрестке без светофора надо сначала посмотреть направо. Если машин нет, то перейти полдороги, а если машины есть, ждать, пока они пройдут, затем перейти полдороги. После этого посмотреть налево и, если машин нет, то перейти дорогу до конца, а если машины есть, ждать, пока они пройдут, а затем перейти дорогу до конца.

В математике для решения типовых задач мы используем определенные правила, описывающие последовательности действий. Например, правила сложения дробных чисел, решения квадратных уравнений и т. д. Обычно любые инструкции и правила представляют собой последовательность действий, которые необходимо выполнить в определенном порядке. Для решения задачи надо знать, что дано, что следует получить и какие действия и в каком порядке следует для этого выполнить.

Предписание, определяющее порядок выполнения действий над данными с целью получения искомого результата, и есть алгоритм.

Алгоритм – заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Какими свойствами обладают алгоритмы?

Основные свойства алгоритмов следующие:

Понятность для исполнителя – исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.

Дискретность (прерывность, раздельность) – алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).

Определенность – каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

Результативность (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для выполнения алгоритма, с выдачей промежуточных результатов.

Массовость означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным.

При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

1. Такое графическое представление называется схемой алгоритма или **блок-схемой**. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде **блочного символа**. Блочные символы соединяются **линиями переходов**, определяющими очередность выполнения действий. В таблице приведены наиболее часто употребляемые символы (таблица 1.2.1).

Таблица 1.2.1

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

2. Блок **"процесс"** применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.
3. Блок **"решение"** используется для обозначения переходов управления по условию. В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.
4. Блок **"модификация"** используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.
5. Блок **"предопределенный процесс"** используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

Способы представления алгоритма, вспомогательные алгоритмы.

Этапы решения задач на компьютере.

Базовые структуры алгоритмов – это определенный набор блоков и стандартных способов их соединения для выполнения типичных последовательностей действий.

К основным структурам относятся следующие:

- линейные
- разветвляющиеся
- циклические

Линейными называются алгоритмы, в которых действия осуществляются последовательно друг за другом. Стандартная блок-схема линейного алгоритма приводится ниже (рис.1.2.2.):

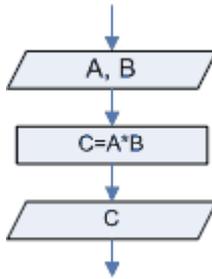


Рис.1.2.2. Блок-схема линейного процесса

Разветвляющимся называется алгоритм, в котором действие выполняется по одной из возможных ветвей решения задачи, в зависимости от выполнения условий. В отличие от линейных алгоритмов, в которых команды выполняются последовательно одна за другой, в разветвляющихся алгоритмах входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (действий).

В качестве условия в разветвляющемся алгоритме может быть использовано любое понятное исполнителю утверждение, которое может соблюдаться (быть истинно) или не соблюдаться (быть ложно). Такое утверждение может быть выражено как словами, так и формулой. Таким образом, алгоритм ветвления состоит из условия и двух последовательностей команд.

В зависимости от того, в обоих ветвях решения задачи находится последовательность команд или только в одной разветвляющиеся алгоритмы делятся на полные и не полные (сокращенные).

Стандартные блок-схемы разветвляющегося алгоритма приведены ниже (рис.1.2.3.):

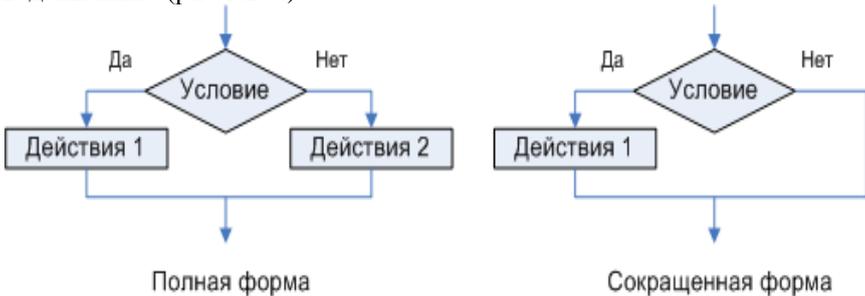


Рис.1.2.3. Блок-схема разветвленного процесса

Циклическим называется алгоритм, в котором некоторая часть операций (тело цикла - последовательность команд) выполняется многократно. Однако слово «многократно» не значит «до бесконечности». Организация циклов, никогда не приводящая к остановке в выполнении алгоритма, является нарушением требования его результативности - получения результата за конечное число шагов.

Перед операцией цикла осуществляются операции присвоения начальных значений тем объектам, которые используются в теле цикла. В цикл входят в качестве базовых следующие структуры:

1. блок проверки условия
2. блок, называемый телом цикла

Существуют три типа циклов:

1. Цикл с предусловием
2. Цикл с постусловием
3. Цикл с параметром (разновидность цикла с предусловием)

Если тело цикла расположено после проверки условий, то может случиться, что при определенных условиях тело цикла не выполнится ни разу. Такой вариант организации цикла, управляемый предусловием, называется циклом с предусловием.

Возможен другой случай, когда тело цикла выполняется по крайней мере один раз и будет повторяться до тех пор, пока не станет ложным условие. Такая организация цикла, когда его тело расположено перед проверкой условия, носит название цикла с постусловием.

Цикл с параметром является разновидностью цикла с предусловием. Особенностью данного типа цикла является то, что в нем имеется параметр, начальное значение которого задается в заголовке цикла, там же задается условие продолжения цикла и закон изменения параметра цикла. Механизм работы полностью соответствует циклу с предусловием, за исключением того, что после выполнения тела цикла происходит изменение параметра по указанному закону и только потом переход на проверку условия.

Стандартные блок-схемы циклических алгоритмов приведены ниже (рис.1.2.4.):

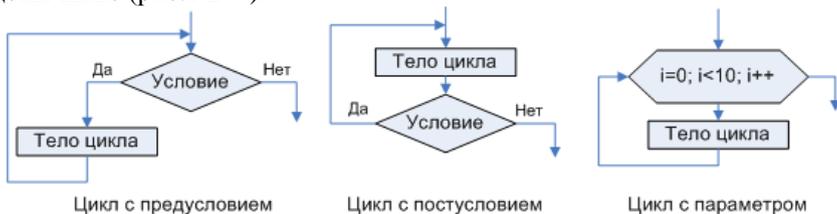


Рис.1.2.4. Блок-схема циклического процесса

Вопросы для самоконтроля

1. Виды и методы создания алгоритма.
2. Этапы решения задач в компьютере.
3. Графический способ записи алгоритмов.
4. Математическая модель

Тема 1.3. Современные технологии программирования в сфере архитектуры и строительства.

Основные модули

- Современные технологии программирования;
- Объектно-ориентированный язык программирования;
- Основные особенности языка Python

Современные технологии программирования.

Основной компонент программного обеспечения - программа - упорядоченная в соответствии с некоторым алгоритмом последовательность команд (инструкций) компьютера для решения задачи пользователя. Чаще всего образ программы хранится в виде исполняемого модуля (отдельного файла или группы файлов).

Пользователь – лицо, заинтересованное в решении некоторой задачи средствами вычислительной техники. По отношению к программному обеспечению компьютерные пользователи делятся на следующие группы:

- системные программисты, занимающиеся разработкой, эксплуатацией и сопровождением системного программного обеспечения;
- прикладные программисты. Выполняют разработку и отладку программ решения задач из различных прикладных сфер деятельности пользователей;
- конечные пользователи. Используют прикладное программное обеспечение для решения задач в своей повседневной деятельности. Различаются по уровню своей подготовки в части знания и использования компьютерной техники;
- администраторы. Как правило, это высококвалифицированные компьютерные специалисты, отвечающие за работу вычислительной сети, баз данных, корпоративной информационной системы в целом, безопасность и защиту данных. Могут иметь определенную специализацию: управление сетевым каталогом, политикой учетных записей, политикой аудита и т.п.

Задача (*problem, task*) – проблема, подлежащая решению в интересах пользователя.

Термин "задача" в программировании означает единицу работы вычислительной системы, требующую выделения вычислительных ресурсов (процессорного времени, оперативной и внешней памяти, файлов и т.п.).

Приложение (*application*) – *программная реализация* решения задачи на компьютере. Приложение может состоять из одной или нескольких взаимосвязанных и взаимодействующих программ.

Принято (весьма условно) делить программы на небольшие (простые), *средней сложности* и большие.

Программа считается небольшой как по размерам, так и по другим признакам, если она удовлетворяет следующим признакам:

- решает одну четко поставленную задачу в хорошо известных ограничениях, к тому же, не очень существенную для какой-либо практической или исследовательской деятельности;
- неважно, насколько быстро она работает;
- ущерб от неправильной работы программы – практически нулевой (за исключением возможности обрушения ею системы, в которой выполняются и другие, более важные задачи);
- не требуется дополнять программу новыми возможностями, практически никому не нужно разрабатывать ее новые версии или исправлять найденные ошибки;
- в связи со сказанным выше не очень нужно прилагать к программе подробную и понятную документацию – для человека, который ею заинтересуется, не составит большого труда понять, как ею пользоваться, просто по исходному коду.

Сложные, или большие, программы, называемые также программными системами, программными комплексами, программными продуктами, отличаются от небольших не столько по размерам (хотя обычно они значительно больше), сколько наличием дополнительных факторов. Эти факторы связаны с их востребованностью и готовностью пользователей платить деньги, как за приобретение самой программы, так и за ее сопровождение и даже за специальное обучение работе с ней.

Примером большой программы может служить *стандартная библиотека* классов *Java*, *C#* или *Python*, соответствующих систем программирования.

Строго говоря, ни одно из указанных свойств не является обязательным для того, чтобы программу можно было считать большой, но при наличии двух-трех из них достаточно уверенно можно утверждать, что она большая. На основании некоторых из перечисленных свойств

можно сделать *вывод*, что большая *программа* или программная система чаще всего представляет собой не просто код или *исполняемый файл*, а включает еще и набор проектной и пользовательской документации.

Программирование (*programming*) - теоретическая и практическая *деятельность*, связанная с созданием программ.

Разработка программных систем (ПС), т.е. *программирование*, имеет ряд специфических особенностей. Прежде всего, следует отметить некоторое противостояние: неформальный характер требований к ПС (постановки задачи) и понятия ошибки в нем, но формализованный основной *объект* разработки – программы ПС. Тем самым разработка ПС содержит определенные этапы формализации, а переход от неформального к формальному существенно неформален.

Разработка ПС носит творческий характер (на каждом шаге приходится делать какой-либо выбор, принимать какое-либо решение), а не сводится к выполнению некоей последовательности регламентированных действий. Тем самым эта разработка ближе к процессу проектирования сложных устройств, но никак не к их массовому производству. Этот творческий характер разработки ПС сохраняется до самого ее конца.

Технологии программирования

В процессе разработки программных систем используются различные технологии программирования. В соответствии с обычным значением слова "технология" под технологией программирования (*programming technology*) понимается совокупность производственных процессов, приводящая к созданию требуемой ПС, а также описание этой совокупности процессов. Другими словами, технология программирования понимается здесь в широком смысле как технология разработки программных средств, включая в нее все процессы, начиная с момента зарождения идеи этого средства до создания необходимой программной документации. Каждый процесс этой совокупности базируется на использовании каких-либо методов и средств, например, компьютера (в этом случае речь идет о компьютерной технологии программирования).

Решение задач с помощью компьютера включает в себя следующие основные этапы, часть из которых осуществляется без участия компьютера.

1. Постановка задачи:

- сбор информации о задаче;
- формулировка условия задачи;

- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных (их типов, диапазонов величин, структуры и т.п.).

Постановка задачи (problem definition) – это точная формулировка требований (функциональных и нефункциональных), предъявляемых к работе программы, с описанием *входной* и *выходной* информации, и, возможно, описание подходов к решению задачи.

2. Анализ и исследование задачи, модели:

- анализ существующих аналогов;
- анализ технических и программных средств;
- разработка математической модели;
- разработка структур данных.

3. Разработка алгоритма:

- выбор метода проектирования алгоритма;
- выбор формы записи алгоритма (блок-схемы, псевдокод и др.);
- выбор тестов и метода тестирования;
- проектирование алгоритма.

4. Программирование:

- выбор языка программирования;
- уточнение способов организации данных;
- запись алгоритма на выбранном языке программирования.

5. Тестирование и отладка:

- синтаксическая отладка;
- отладка семантики и логической структуры;
- тестовые расчеты и анализ результатов тестирования;
- совершенствование программы.

6. Анализ результатов решения задачи и уточнение в случае необходимости математической модели с повторным выполнением этапов 2 - 5.

7. Сопровождение программы:

- доработка программы для решения конкретных задач;
- составление документации к решенной задаче, к математической модели, к алгоритму, к программе, к набору тестов, к использованию.

Итак, создавая математическую модель для решения задачи, нужно:

- выделить предположения, на которых будет основываться математическая модель;
- определить, что считать исходными данными и результатами;

- записать математические соотношения, связывающие результаты с исходными данными.

При построении математических моделей далеко не всегда удается найти формулы, явно выражающие искомые величины через данные. В таких случаях используются математические методы, позволяющие дать ответы той или иной степени точности.

Основные особенности языка Python

Python – интерпретируемый язык программирования. Это значит, что исходный код частями преобразуется в машинный в процессе его чтения специальной программой – интерпретатором.

Python характеризуется ясным синтаксисом. Читать код на нем легче, чем на других языках программирования, так как в Питоне мало используются такие вспомогательные синтаксические элементы как скобки, точки с запятыми. С другой стороны, правила языка заставляют программистов делать отступы для обозначения вложенных конструкций. Понятно, что хорошо оформленный текст с малым количеством отвлекающих элементов читать и понимать легче.

Python – это полноценный во многом универсальный язык программирования, используемый в различных сферах. Основная, но не единственная, поддерживаемая им парадигма, – объектно-ориентированное программирование. Однако в данном курсе мы только упомянем об объектах, а будем изучать структурное программирование, так как оно является базой. Без знания основных типов данных, ветвлений, циклов, функций нет смысла изучать более сложные парадигмы, так как в них все это используется.

Типы данных. Переменные

В реальной жизни мы совершаем различные действия над окружающими нас предметами, или объектами. Мы меняем их свойства, наделяем новыми функциями. По аналогии с этим компьютерные программы также управляют объектами, только виртуальными, цифровыми. Пока не дойдем до уровня объектно-ориентированного программирования, будем называть такие объекты **данными**.

Очевидно, данные бывают разными. Часто компьютерной программе приходится работать с числами и строками. Так на прошлом уроке мы работали с числами, выполняя над ними арифметические операции. Операция сложения выполняла изменение первого числа на величину второго, а умножение увеличивало одно число в количество раз, соответствующее второму.

Здесь следует для себя отметить, что то, что делает оператор в операции, зависит не только от него, но и от типов данных, которыми он оперирует. Молоток в случае нападения на вас крокодила перестанет играть роль строительного инструмента. Однако в большинстве случаев операторы не универсальны. Например, знак плюса неприменим, если операндами являются, с одной стороны, число, а с другой – строка.

```
>>> 1 + 'a'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +:
'int' and 'str'
```

Здесь в строке `TypeError: unsupported operand type(s) for +: 'int' and 'str'`, интерпретатор сообщает, что произошла ошибка типа – неподдерживаемый операнд для типов `int` и `str`.

Изменение типов данных

Приведенную выше операцию все-таки можно выполнить, если превратить число 1 в строку "1". Для изменения одних типов данных в другие в языке Python предусмотрен ряд встроенных в него функций (что такое функция в принципе, вы узнаете в других уроках). Поскольку мы пока работаем только с тремя типами (`int`, `float` и `str`), рассмотрим вызовы соответствующих им функций – `int()`, `float()`, `str()`.

```
>>> str(1) + 'a'
'1a'
>>> int('3') + 4
7
>>> float('3.2') + int('2')
5.2
>>> str(4) + str(1.2)
'41.2'
```

Эти функции преобразуют то, что помещается в их скобки соответственно в целое число, вещественное число или строку. Однако преобразовать можно не все:

```
>>> int('hi')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with
base 10: 'hi'
```

Здесь возникла ошибка значения (`ValueError`), так как передан литерал (в данном случае строка с буквенными символами), который

нельзя преобразовать к числу с основанием 10. Однако функция `int` не такая простая:

```
>>> int('101', 2)
5
>>> int('F', 16)
15
```

Если вы знаете о различных системах счисления, то поймете, что здесь произошло.

Обратим внимание еще на одно. Данные могут называться **значениями**, а также **литералами**. Эти три понятия ("данные", "значение", "литерал") не обозначают одно и то же, но близки и нередко употребляются как синонимы. Чтобы понять различие между ними, места их употребления, надо изучить программирование глубже.

Переменные

Данные хранятся в ячейках памяти компьютера. Когда мы вводим число, оно помещается в какую-то ячейку памяти. Но как потом узнать, куда именно? Как впоследствии обращаться к этим данными? Нужно как-то запомнить, пометить соответствующую ячейку.

Раньше, при написании программ на машинном языке, обращение к ячейкам памяти осуществляли с помощью указания их регистров, то есть конкретно сообщали, куда положить данные и откуда их взять. Однако с появлением ассемблеров при обращении к данным стали использовать словесные **переменные**, что куда удобней для человека.

Механизм связи между переменными и данными может различаться в зависимости от языка программирования и типов данных. Пока достаточно запомнить, что в программе данные связываются с каким-либо именем и в дальнейшем обращение к ним возможно по этому имени-переменной.

Слово "переменная" обозначает, что сущность может меняться, она непостоянна. Действительно, вы увидите это в дальнейшем, одна и та же переменная может быть связана сначала с одними данными, а потом – с другими. То есть ее значение может меняться, она переменчива.

В программе на языке Python, как и на большинстве других языков, связь между данными и переменными устанавливается с помощью знака `=`. Такая операция называется присваивание (также говорят "присвоение"). Например, выражение `sq = 4` означает, что на объект, представляющий собой число 4, находящееся в определенной области памяти, теперь ссылается переменная `sq`, и обращаться к этому объекту следует по имени `sq` (рис.1.3.2).



Рис.1.3.2. Присвоение переменных

Имена переменных могут быть любыми. Однако есть несколько общих правил их написания:

Желательно давать переменным осмысленные имена, говорящие о назначении данных, на которые они ссылаются.

Имя переменной не должно совпадать с командами языка (зарезервированными ключевыми словами).

Имя переменной должно начинаться с буквы или символа подчеркивания (`_`), но не с цифры.

Имя переменной не должно содержать пробелы.

Чтобы узнать значение, на которое ссылается переменная, находясь в режиме интерпретатора, достаточно ее вызвать, то есть написать имя и нажать Enter.

```
>>> sq = 4
>>> sq
4
```

Вот более сложный пример работы с переменными в интерактивном режиме:

```
>>> apples = 100
>>> eat_day = 5
>>> day = 7
>>> apples = apples - eat_day * day
>>> apples
65
```

Здесь фигурируют три переменные: *apples*, *eat_day* и *day*. Каждой из них присваивается свое значение. Выражение `apples = apples - eat_day * day` сложное. Сначала выполняется подвыражение, стоящее справа от знака равенства. После этого его результат присваивается переменной *apples*, в результате чего ее старое значение (100) теряется. В подвыражении `apples - eat_day * day` вместо имен переменных на самом деле используются их значения, то есть числа 100, 5 и 7.

Ввод и вывод данных

Функцией `print()` отвечает за вывод данных, по умолчанию на экран. Если код содержится в файле, то без нее не обойтись. В интерактивном режиме в ряде случаев можно обойтись без нее.

Ввод данных в программу и их вывод важны в программировании. Без ввода программы делали бы одно и то же, исключая случаи, когда в них самих генерируются случайные значения. Вывод позволяет увидеть, использовать, передать дальше результат работы программы.

Обычно требуется, чтобы программа обрабатывала какой-то диапазон различных входных данных, которые поступают в нее из внешних источников. В качестве последних могут выступать файлы, клавиатура, сеть, выходные данные из другой программы. Вывод данных также возможен в файлы и др. Однако во многих случаях это происходит на экран монитора.

Можно сказать, что программа – это открытая система, которая обменивается чем-либо с внешней для нее средой. Если живой организм в основном обменивается веществом и энергией, то программа – данными, информацией.

Вывод данных. Функция `print()`

Что такое функция в программировании, узнаем позже. Пока будем считать, что `print()` – это такая команда языка Python, которая выводит то, что в ее скобках на экран.

```
>>> print(1032)
1032
>>> print(2.34)
2.34
>>> print("Hello")
Hello
```

В скобках могут быть любые типы данных. Кроме того, количество данных может быть различным:

```
>>> print("a:", 1)
a: 1
>>> one = 1
>>> two = 2
>>> three = 3
>>> print(one, two, three)
1 2 3
```

Можно передавать в функцию `print()` как непосредственно литералы (в данном случае "a:" и 1), так и переменные, вместо которых будут выведены их значения. Аргументы функции (то, что в скобках),

разделяются между собой запятыми. В выводе вместо запятых значения разделены пробелом.

Если в скобках стоит выражение, то сначала оно выполняется, после чего `print()` уже выводит результат данного выражения:

```
>>> print("hello" + " " + "world")
hello world
>>> print(10 - 2.5/2)
8.75
```

В `print()` предусмотрены дополнительные параметры. Например, через параметр `sep` можно указать отличный от пробела разделитель строк:

```
>>> print("Mon", "Tue", "Wed", "Thu",
... "Fri", "Sat", "Sun", sep="-")
Mon-Tue-Wed-Thu-Fri-Sat-Sun
>>> print(1, 2, 3, sep="//")
1//2//3
```

Параметр `end` позволяет указывать, что делать, после вывода строки. По-умолчанию происходит переход на новую строку. Однако это действие можно отменить, указав любой другой символ или строку:

```
>>> print(10, end="")
10>>>
```

Обычно `end` используется не в интерактивном режиме, а в скриптах, когда несколько выводов подряд надо разделить не переходом на новую строку, а, скажем, запятыми. Сам переход на новую строку обозначается символом `'\n'`. Если присвоить это значение параметру `end`, то никаких изменений в работе функции `print` вы не увидите, так как это значение и так присвоено по умолчанию:

```
>>> print(10, end='\n')
10
>>>
```

Однако, если надо отступить на одну дополнительную строку после вывода, то можно сделать так:

```
>>> print(10, end='\n\n')
10
>>>
```

Следующее, что стоит рассказать о функции `print` – это использование форматирования строк. На самом деле оно никакого отношения к `print` не имеет, а применяется к строкам. Однако часто форматирование используется в сочетании с вызовом функции `print()`.

Форматирование может выполняться в так называемом старом стиле или с помощью строкового метода `format`. Старый стиль также называют Си-стилем, так как он схож с тем, как происходит вывод на экран в языке С. Рассмотрим пример:

```
>>> pupil = "Ben"
>>> old = 16
>>> grade = 9.2
>>> print("It's %s, %d. Level: %f" %
... (pupil, old, grade))
It's Ben, 16. Level: 9.200000
```

Здесь вместо трех комбинаций символов `%s`, `%d`, `%f` подставляются значения переменных `pupil`, `old`, `grade`. Буквы `s`, `d`, `f` обозначают типы данных – строку, целое число, вещественное число. Если бы требовалось подставить три строки, то во всех случаях использовалось бы сочетание `%s`.

Хотя в качестве значения переменной `grade` было указано число 9.2, на экран оно вывелось с дополнительными нулями. Однако мы можем указать, сколько требуется знаков после запятой, записав перед буквой `f` точку с желаемым числом знаков в дробной части:

```
>>> print("It's %s, %d. Level: %.1f"
... % (pupil, old, grade))
It's Ben, 16. Level: 9.2
Теперь посмотрим на метод format():
>>> print("This is a {0}. It's {1}."
... .format("ball", "red"))
This is a ball. It's red.
>>> print("This is a {0}. It's {1}."
... .format("cat", "white"))
This is a cat. It's white.
>>> print("This is a {0}. It's {1} {2}."
... .format(1, "a", "number"))
This is a 1. It's a number.
```

В строке в фигурных скобках указаны номера данных, которые будут сюда подставлены. Далее к строке применяется метод `format()`. В его скобках указываются сами данные (можно использовать переменные). На нулевое место подставится первый аргумент метода `format()`, на место с номером 1 – второй и т. д.

На самом деле возможности метода `format()` существенно шире, и для их изучения понадобился бы отдельный урок. Нам пока будет достаточно этого.

Ввод данных. Функция input()

За ввод в программу данных с клавиатуры в Python отвечает функция `input`. Когда вызывается эта функция, программа останавливает свое выполнение и ждет, когда пользователь введет текст. После этого, когда он нажмет `Enter`, функция `input()` заберет введенный текст и передаст его программе, которая уже будет обрабатывать его согласно своим алгоритмам.

Если в интерактивном режиме ввести команду `input()`, то ничего интересного вы не увидите. Компьютер будет ждать, когда вы что-нибудь введете и нажмете `Enter` или просто нажмете `Enter`. Если вы что-то ввели, это сразу же отобразится на экране:

```
>>> input()
Yes!
'Yes!'
```

Функция `input()` передает введенные данные в программу. Их можно присвоить переменной. В этом случае интерпретатор не выводит строку сразу же:

```
>>> answer = input()
No, it is not.
```

В данном случае строка сохраняется в переменной `answer`, и при желании мы можем вывести ее значение на экран:

```
>>> answer
'No, it is not.'
```

При использовании функции `print()` кавычки в выводе опускаются:

```
>>> print(answer)
No, it is not.
```

Куда интересней использовать функцию `input()` в скриптах – файлах с кодом. Рассмотрим такую программу (рис.1.3.3).

При запуске программы, компьютер ждет, когда будет введена сначала одна строка, потом вторая. Они будут присвоены переменным `nameUser` и `cityUser`. После этого значения этих переменных выводятся на экран с помощью форматированного вывода.

```
test.py ✖
1 nameUser = input()
2 cityUser = input()
3 print("Вас зовут {0}. Ваш город {1}.".format(nameUser, cityUser))
4
```

Статус	pl@pl-desk:~\$ python3 test.py
Компилятор	Лев
Сообщения	Саванна Вас зовут Лев. Ваш город Саванна.
Заметки	pl@pl-desk:~\$
Терминал	

Рис.1.3.3. Использование функции input()

Вышеприведенный скрипт далек от совершенства. Откуда пользователю знать, что хочет от него программа? Чтобы не вводить человека в замешательство, для функции input предусмотрен специальный параметр-приглашение. Это приглашение выводится на экран при вызове input(). Усовершенствованная программа может выглядеть так (рис.1.3.4):

```
test.py ✖
1 nameUser = input("Ваше имя: ")
2 cityUser = input("Ваш город: ")
3 print("Вас зовут {0}. Ваш город {1}.".format(nameUser, cityUser))
4
```

Статус	pl@pl-desk:~\$ python3 test.py
Компилятор	Ваше имя: Сикама
Сообщения	Ваш город: где-то в Африке Вас зовут Сикама. Ваш город где-то в Африке.
Заметки	pl@pl-desk:~\$
Терминал	

Рис.1.3.4. Программирование с использованием функции input

Обратите внимание, что в программу поступает строка. Даже если ввести число, функция input() все равно вернет его строковое представление. Но что делать, если надо получить число? Ответ: использовать функции преобразования типов (рис.1.3.5).

```
test.py ✖
1 qtyOranges = input("Сколько апельсинов? ")
2 priceOrange = input("Цена одного апельсина? ")
3
4 qtyOranges = int(qtyOranges)
5 priceOrange = float(priceOrange)
6
7 sumOranges = qtyOranges * priceOrange
8
9 print("Заплатите", sumOranges, "руб.")
10
```

Статус	pl@pl-desk:~\$ python3 test.py Сколько апельсинов? 5
Компилятор	Цена одного апельсина? 21.50
Сообщения	Заплатите 107.5 руб.
Заметки	
Терминал	pl@pl-desk:~\$

Рис.1.3.5. Программирование с использованием функции input

В данном случае с помощью функций `int()` и `float()` строковые значения переменных `qtyOranges` и `priceOrange` преобразуются соответственно в целое число и вещественное число. После этого новые численные значения присваиваются тем же переменным.

Программный код можно сократить, если преобразование типов выполнить в тех же строках кода, где вызывается функция `input()`:

```
qtyOranges = int(input("Сколько апельсинов? "))
priceOrange = float(input("Цена одного? "))
sumOranges = qtyOranges * priceOrange
print("Заплатите", sumOranges, "руб.")
```

Сначала выполняется функция `input()`. Она возвращает строку, которую функция `int()` или `float()` сразу преобразует в число. Только после этого происходит присваивание переменной, то есть она сразу получает численное значение.

Вопросы для самоконтроля

1. Основные особенности языка Python
2. Типы данных. Переменные
3. Операции в программировании
4. Изменение типов данных
5. Ввод и вывод данных
6. Ввод данных. Функция `input()`

ГЛАВА II. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Тема 2.1. Информационные системы в архитектуре и строительстве

Основные модули

- Информационные системы, их значение, применение и задачи.
- Строение информационных систем и их характеристика.
- Основные процессы информационных систем.
- Составные части обеспечивающие информационные системы: техническое, программное, математическое, информационное и правовое обеспечение.

Информационная система (ИС)– взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели

Типы обеспечивающих подсистем

Структуру информационной системы составляет совокупность отдельных ее частей, называемых подсистемами.

Подсистема - это часть системы, выделенная по какому-либо признаку.

Общую структуру информационной системы можно рассматривать как совокупность подсистем независимо от сферы применения. В этом случае говорят о структурном признаке классификации, а подсистемы называют обеспечивающими. Таким образом, структура любой информационной системы может быть представлена совокупностью обеспечивающих подсистем.

Среди обеспечивающих подсистем обычно выделяют информационное, техническое, математическое, программное, организационное и правовое обеспечение.

Информационное обеспечение

Назначение подсистемы информационного обеспечения состоит в современном формировании и выдаче достоверной информации для принятия управленческих решений.

Информационное обеспечение - совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных.

Техническое обеспечение

Техническое обеспечение - комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы

Комплекс технических средств составляют:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы и др.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение. Документацию можно условно разделить на три группы:

- общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению;
- специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;
- нормативно-справочную, используемую при выполнении расчетов по техническому обеспечению.

К настоящему времени сложились две основные формы организации технического обеспечения (формы использования технических средств): централизованная и частично или полностью децентрализованная.

Централизованное техническое обеспечение базируется на использовании в информационной системе больших ЭВМ и вычислительных центров.

Децентрализация технических средств предполагает реализацию функциональных подсистем на персональных компьютерах непосредственно на рабочих местах.

Математическое и программное обеспечение

Математическое и программное обеспечение - совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

К средствам математического обеспечения относятся:

- средства моделирования процессов управления;
- типовые задачи управления;

методы математического программирования, математической статистики, теории массового обслуживания и др.

В состав программного обеспечения входят общесистемные и специальные программные продукты, а также техническая документация.

К общесистемному программному обеспечению относятся комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных.

Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной информационной системы. В его состав входят пакеты прикладных программ (ППП), реализующие разработанные модели разной степени адекватности, отражающие функционирование реального объекта.

Техническая документация на разработку программных средств должна содержать описание задач, задание на алгоритмизацию, экономико-математическую модель задачи, контрольные примеры.

Организационное обеспечение

Организационное обеспечение - совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации;

- подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;

- разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Правовое обеспечение

Правовое обеспечение - совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

Главной целью правового обеспечения является укрепление законности.

В состав правового обеспечения входят законы, указы, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, организаций, местных органов власти. В правовом обеспечении можно выделить общую часть, регулирующую функционирование любой информационной системы, и локальную часть, регулирующую функционирование конкретной системы.

Правовое обеспечение этапов разработки информационной системы включает нормативные акты, связанные с договорными отношениями разработчика и заказчика и правовым регулированием отклонений от договора.

Правовое обеспечение этапов функционирования информационной системы включает:

- статус информационной системы;
- права, обязанности и ответственность персонала;
- правовые положения отдельных видов процесса управления;
- порядок создания и использования информации и др.

Вопросы для самоконтроля

1. Информационные системы, их значение, применение и задачи.
2. Строение информационных систем и их характеристика.
3. Основные процессы информационных систем.
4. Составные части обеспечивающие информационные системы

Тема 2.2. Методы проектирования и построения информационных систем

Основные модули

- Методология создания информационных систем
- Моделью жизненного цикла информационной системы
- Каскадная модель жизненного цикла информационной системы
- Спиральная модель жизненного цикла информационной системы

Методология создания информационных систем заключается в организации процесса построения информационной системы для предприятия и обеспечении управления этим процессом. Таким образом, современная методология проектирования должна поддерживать сбор данных, их анализ, проектирование, оценку проекта и оценку возможности удовлетворения техническим характеристикам разрабатываемой системы.

Методология проектирования - это процесс, требующий наличия совокупности теоретических и инженерных знаний и обеспечивающий упорядоченное создание информационных систем. Методология проектирования обычно включает три следующих аспекта:

- основные концепции и понятия, используемые при проектировании и реализации систем;
- технологию, организацию и управление процессом проектирования;
- инструментальные средства.

Современные методологии проектирования ИС должны обеспечивать представление следующей информации:

- описание объекта автоматизации, а также места разрабатываемой информационной системы и целей, которые должны быть достигнуты в процессе разработки системы;
- описание функциональных возможностей ИС, достаточное для решения вопроса о том, что поставленные цели автоматизации достижимы;
- спецификации проекта, гарантирующие достижение заданных технических характеристик системы;
- описание реализации предлагаемой системы, достаточное для оценки времени ее разработки и необходимых для этого трудозатрат;
- детальный план создания системы с оценкой сроков ее разработки.

Основными задачами, решение которых должна обеспечивать методология создания информационных систем, являются следующие:

- обеспечение создания информационных систем, отвечающих целям и задачам предприятия и соответствующих предъявляемым к ним требованиям по автоматизации деловых процессов;
- гарантия создания системы с заданными параметрами в течение заданного времени в рамках заранее оговоренного бюджета;
- простота сопровождения, модификации и расширения системы для обеспечения ее соответствия изменяющимся условиям работы предприятия;
- возможность использования в создаваемой системе разработанных ранее и уже применяемых на предприятии средств информационных технологий (программного обеспечения, баз данных, средств вычислительной техники, телекоммуникаций).

Методология проектирования ИС реализуется через конкретные технологии и поддерживающие их стандарты, методики и инструментальные средства, которые обеспечивают выполнение процессов жизненного цикла информационных систем. Основное содержание технологии проектирования составляют технологические инструкции, состоящие из описания последовательности технологических операций, условий, в зависимости от которых выполняется та или иная операция, и описаний самих этих операций.

Технология проектирования может быть представлена как совокупность трех составляющих:

- заданной последовательности выполнения технологических операций проектирования;
- критериев и правил, используемых для оценки результатов выполнения технологических операций;
- графических и текстовых средств (нотаций), используемых для описания проектируемой системы.

Каждая технологическая операция должна обеспечиваться следующими материальными и информационными ресурсами:

- данными, полученными на предыдущей операции (или исходными данными), представленными в стандартном виде;
- методическими материалами, инструкциями, нормативами и стандартами;
- программными и техническими средствами;
- исполнителями.

Результаты выполнения операции должны представляться в некотором стандартном виде, обеспечивающем их адекватное восприятие

при выполнении следующей технологической операции (на которой они будут использоваться в качестве исходных данных).

Можно сформулировать следующий ряд *общих требований*, которым должна удовлетворять технология проектирования, разработки и сопровождения информационных систем:

- поддержка полного жизненного цикла информационной системы;
- гарантированное достижение целей разработки системы с заданным качеством и в установленное время;
- возможность разделения крупных проектов на ряд подсистем - декомпозицию проекта на составные части, разрабатываемые группами исполнителей ограниченной численности, - с последующей интеграцией таких составных частей в единое целое;
- возможность ведения работ по проектированию отдельных подсистем небольшими группами (3-7 человек);
- минимальное время разработки работоспособной системы;
- возможность автоматического выпуска проектной документации и синхронизации ее версий с версиями проекта;
- независимость выполняемых проектных решений от средств реализации системы - систем управления базами данных, операционной системы, языка и системы программирования.

Понятие *жизненного цикла* является одним из базовых понятий технологии и методологии проектирования информационных систем.

Жизненный цикл информационной системы представляет собой непрерывный процесс, который начинается с момента принятия решения о создании информационной системы и заканчивается в момент полного завершения ее эксплуатации.

В целом полный жизненный цикл информационных систем включает следующие основные этапы:

- стратегическое планирование;
- системный анализ - определение потребности, назначения и основных функциональных характеристик ИС, оценка затрат и возможной эффективности ее применения;
- проектирование ИС;
- реализация (создание информационной системы);
- ввод в действие и эксплуатация;
- сопровождение ИС - эксплуатационное обслуживание, развитие функциональных возможностей и повышение эксплуатационных характеристик ИС, тиражирование и адаптация к другим ИС.

Существует международный стандарт, регламентирующий жизненный цикл информационных систем, - ISO/IEC 12207, где *ISO*

(*International Organization of Standardization*) - это международная организация по стандартизации, а *IEC (International Electrotechnical Commission)* - международная комиссия по электротехнике. Стандарт ISO/IEC 12207 определяет структуру жизненного цикла, содержащую процессы, действия и задачи, которые должны быть выполнены во время создания информационной системы.

Согласно данному стандарту, структура жизненного цикла основывается на трех группах процессов:

- основные процессы жизненного цикла (приобретение, поставка, разработка, эксплуатация, сопровождение);
- вспомогательные процессы, обеспечивающие выполнение основных процессов (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит, разрешение проблем);
- организационные процессы (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение самого жизненного цикла, обучение).

Согласно методологии, предлагаемой Rational Software, жизненный цикл информационной системы подразделяется на четыре стадии:

- начало;
- уточнение;
- конструирование;
- переход (передача в эксплуатацию).

На *начальной стадии* устанавливается область применения системы и определяются ее граничные условия. Для этого необходимо идентифицировать все внешние объекты, с которыми должна взаимодействовать разрабатываемая система, и определить характер этого взаимодействия на высоком уровне. На начальной стадии идентифицируются все функциональные возможности системы и производится описание наиболее существенных из них. В конце начальной стадии определяются и документально описываются:

- критерии успеха разработки;
- оценка риска;
- оценка ресурсов, необходимых для выполнения разработки;
- календарный план с указанием сроков завершения основных этапов.

На стадии уточнения проводится анализ прикладной области и разрабатывается архитектурная основа информационной системы. Это означает, что необходимо описать большинство функциональных возможностей системы и учесть взаимосвязи между отдельными ее составляющими. В конце стадии уточнения проводится анализ архитектурных решений и способов устранения главных элементов риска, сохраняющихся в проекте.

На стадии конструирования разрабатывается законченное изделие, готовое к передаче пользователю. По окончании этого определяется работоспособность созданного программного обеспечения.

На стадии перехода производится передача разработанного программного обеспечения пользователям. При эксплуатации системы в реальных условиях часто возникают различные проблемы, которые требуют внесения дополнительных коррективов в разработанный продукт. Это, как правило, связано с обнаружением ошибок и недоработок. В конце стадии перехода необходимо определить, были ли достигнуты цели разработки.

Для дальнейшего понимания сути процесса проектирования информационной системы необходимо определить понятие модели жизненного цикла.

Моделью жизненного цикла информационной системы мы будем называть некоторую последовательность осуществления процессов, действий и задач, выполняемых на протяжении всего ее жизненного цикла, а также взаимосвязи между этими процессами, действиями и задачами.

К настоящему времени в нашей стране и за рубежом наибольшее распространение получили две основные модели жизненного цикла: каскадная модель («водопад») и спиральная модель.

Каскадная модель демонстрирует классический подход к разработке широкого спектра различных систем в любых прикладных областях. Данная модель широко использовалась в 1970-х - 80-х гг.

Главной особенностью каскадной модели (рис. 2.2.1) является разбиение всей разработки информационной системы на этапы, когда переход с одного этапа на следующий происходит только после того, как будут полностью завершены все работы на предыдущем этапе. Причем каждый такой этап завершается выпуском полного комплекта документации, достаточной, чтобы разработка могла быть продолжена другой командой разработчиков.

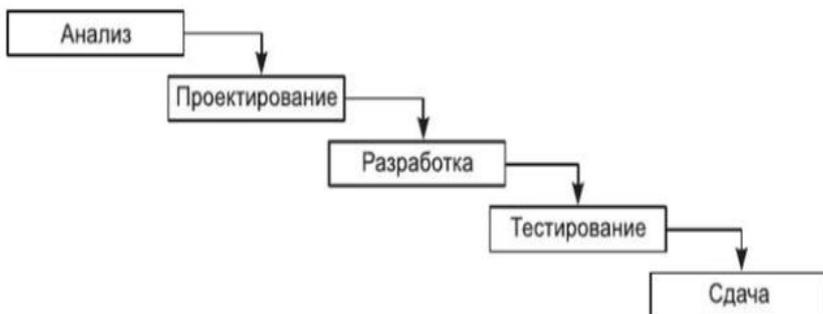


Рис. 2.2.1. Каскадная модель разработки

В каскадной модели выделяются следующие этапы разработки:

- анализ требований заказчика;
- проектирование;
- разработка информационной системы;
- тестирование и опытная эксплуатация;
- сдача готового продукта.

На первом этапе проводится исследование решаемой проблемы, четко формулируются все требования заказчика. Результатом данного этапа является *техническое задание* (задание на разработку), согласованное с разработчиком и с заказчиком. Этот документ подписывается обеими сторонами, чтобы в будущем исключить любые возможные претензии.

На втором этапе разрабатываются проектные решения, удовлетворяющие всем требованиям, сформулированным в техническом задании. Результатом данного этапа является *комплект проектной документации*, содержащей все необходимые данные для реализации проекта.

Третий этап - это собственно реализация проекта. Здесь осуществляется разработка программного обеспечения (кодирование) в соответствии с проектными решениями, полученными на предыдущем этапе. Результатом выполнения данного этапа является *готовый программный продукт*.

На четвертом этапе выполняется проверка созданного программного обеспечения на предмет его соответствия требованиям, заявленным в техническом задании. *Опытная эксплуатация* позволяет выявить различные скрытые недостатки, проявляющиеся в реальных условиях работы информационной системы.

Последний этап - *сдача готового проекта*.

Перечисленные этапы состоят из целого ряда процедур уточнения требований к системе и вариантов проектных решений. Они могут включать в себя произвольное число циклов уточнения, изменения и дополнения уже принятых и реализованных проектных решений, когда происходит развитие информационной системы и модернизация отдельных ее компонентов.

Рассмотрим основные *достоинства каскадной модели* :

- на каждом этапе формируется заверченный набор проектной документации, отвечающий критериям полноты и согласованности;
- на заключительных этапах также разрабатывается пользовательская документация, охватывающая все предусмотренные стандартами виды обеспечения информационной системы: организационное, методическое, информационное, программное, аппаратное;
- выполняемые в логичной последовательности этапы работ позволяют планировать сроки завершения и соответствующие затраты.

Следовательно, каскадный подход можно применять при построении таких информационных систем, для которых в самом начале разработки можно достаточно точно сформулировать все требования.

Однако у каскадной модели есть и *недостатки*. Например, модели автоматизируемого объекта (ранние представления будущего вида объекта) могут в силу различных причин устареть за время разработки. В целом такое положение дел естественно, особенно если учесть, что на разработку конечного программного продукта уходит несколько лет.

Однако главным недостатком каскадной схемы обычно считается задержка получения результатов. Она проявляется в основном в том, что разрабатываемая информационная система может не соответствовать требованиям пользователей (по разным причинам). Ошибки, допущенные на любом из этапов, выясняются, как правило, на последующих этапах работ, так как согласование результатов с заинтересованными сторонами производится только после завершения очередного этапа, что и приводит к необходимости возврата на предыдущие стадии разработки. Поэтому после того, как ошибки проявятся, проект возвращается на предыдущий этап, перерабатывается и снова передается на последующую стадию. Естественно, это приводит к незапланированным ранее перерасходам ресурсов (материальным, финансовым, человеческим, временным и т. д.).

Чем сложнее проект, тем больше продолжительность каждого из этапов разработки и тем сложнее взаимосвязи между отдельными ча-

стями проекта, количество которых также увеличивается. Обычно результаты реально видны лишь на этапе тестирования, т. е. после завершения анализа, проектирования и разработки - этапов, выполнение которых требует значительного времени и средств. Кроме того, недоработки предыдущего уровня могут обнаруживаться не сразу на последующем уровне, а позднее (например, на стадии опытной эксплуатации могут проявиться ошибки в описании предметной области). Это означает, что часть проекта должна быть возвращена на начальный уровень работы. Да и вообще проект может быть возвращен с любого этапа на любой предыдущий этап, поэтому в реальности каскадная схема разработки имеет вид, приведенный на рис. 2.2.2.

Одной из причин данной ситуации является то, что в качестве экспертов, участвующих в описании (анализе) предметной области, нередко выступают будущие пользователи системы, которые часто не могут четко сформулировать то, что они хотели бы получить (исполнители обычно не являются специалистами в предметной области решаемой задачи, а заказчики - далеки от программирования).

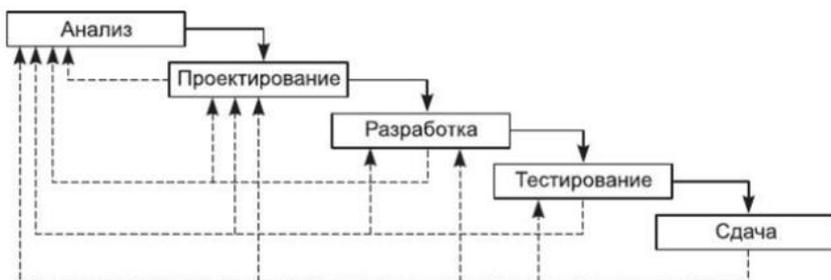


Рис. 2.2.2. Реальный процесс разработки по каскадной схеме

Спиральная модель, в отличие от каскадной, предполагает итерационный процесс разработки информационной системы (т.е. процесс последовательной доработки некоторого «сырого» варианта системы до варианта, необходимого заказчику). При этом возрастает значение начальных этапов жизненного цикла, таких как анализ и проектирование, когда проверяется и обосновывается реализуемость технических решений путем создания прототипов.

Каждая итерация (каждый виток спирали) в этой модели представляет собой законченный цикл разработки, приводящий к выпуску некоторой версии программного продукта, которая совершенствуется (уточняются цели и характеристики продукта от итерации к итерации),

чтобы стать законченной системой (рис. 2.2.3). В результате этого выработывается обоснованный вариант, который и доводится до окончательной реализации.

Использование спиральной модели позволяет осуществлять переход на следующий этап выполнения проекта, не дожидаясь полного завершения работ на текущем (окончательно этап может быть выполнен и на следующем шаге). Главная задача такого подхода к разработке программного продукта - как можно скорее создать работоспособный продукт, который можно представить пользователям для последующего внесения изменений, уточнений и дополнений. Эти изменения, как и выявленные пользователями ошибки, устраняются и тем самым пользователи, постоянно отслеживая изменения в проекте, в итоге получают законченный продукт, максимально удовлетворяющий их потребности на современном этапе.



Рис. 2.2.3. Спиральная модель жизненного цикла информационной системы

Спиральный подход к разработке программного обеспечения позволяет получить следующие *преимущества*:

- итерационная разработка существенно упрощает внесение изменений в проект при изменении требований заказчика (будущего пользователя);
- при использовании спиральной модели отдельные элементы информационной системы интегрируются в единое целое постепенно,

приближаясь к окончательному варианту, приемлемому для пользователя;

- уменьшается уровень рисков создания системы в срок и заданными параметрами;
- итерационный подход упрощает повторное использование компонентов программно-математического обеспечения (ПМО);
- анализ проекта после проведения нескольких начальных итераций позволяет выявить общие, многократно используемые компоненты, которые на последующих итерациях будут совершенствоваться;
- спиральная модель позволяет получить более надежную и устойчивую систему, так как ошибки обнаруживаются и исправляются на каждой итерации;
- итерационный подход позволяет совершенствовать процесс разработки;
- анализ, проводимый в конце каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки, и улучшить ее на следующей итерации.

В результате всего этого уровень рисков максимален в начале разработки проекта, но по мере продвижения к концу проекта ожидаемый риск уменьшается. Данное утверждение справедливо при любой модели разработки, однако при использовании спиральной модели уменьшение уровня рисков происходит с наибольшей скоростью.

Основная проблема спирального цикла - это определение момента перехода на следующий этап. Для решения этой проблемы вводятся временные ограничения на каждый этап жизненного цикла, и переход на следующий этап происходит строго в соответствии с планом, даже если не вся запланированная на текущем этапе работа завершена.

Среди основных процессов жизненного цикла в рамках спиральной модели наибольшую важность имеют три:

- разработка;
- эксплуатация;
- сопровождение.

Каждый из этих процессов характеризуется определенными задачами и методами решения, исходными данными, полученными на предыдущем этапе, и результатами.

Разработка информационной системы включает в себя все работы по созданию информационного программного обеспечения и его компонентов в соответствии с заданными требованиями. Этот процесс также включает:

- оформление проектной и эксплуатационной документации;

- подготовку материалов, необходимых для проведения тестирования разработанных программных продуктов;
- разработку материалов, необходимых для организации обучения персонала.

Разработка ИС, как правило, включает в себя стратегическое планирование, анализ, проектирование и реализацию (программирование).

Эксплуатационные работы можно разделить на подготовительные и основные.

К подготовительным эксплуатационным работам относят:

- конфигурирование базы данных и рабочих мест пользователей;
- обеспечение пользователей эксплуатационной документацией;
- обучение персонала.

Основные эксплуатационные работы включают:

- непосредственно эксплуатацию;
- локализацию проблем и устранение причин их возникновения;
- модификацию программного обеспечения;
- подготовку предложений по совершенствованию системы;
- развитие и модернизацию системы.

Основными предварительными действиями при подготовке к организации технического обслуживания информационной системы являются следующие:

- выделение наиболее ответственных узлов системы и определение для них критичности простоя. Это позволяет выделить наиболее критичные составляющие информационной системы и оптимизировать распределение ресурсов для ее технического обслуживания;

- определение задач технического обслуживания и их разделение на внутренние (решаемые силами обслуживающего подразделения) и внешние (решаемые специализированными сервисными организациями), – производится определение круга исполняемых функций и разделение ответственности;

- проведение анализа имеющихся внутренних и внешних ресурсов, необходимых для организации технического обслуживания в рамках описанных задач и разделения компетенции (наличие гарантии на оборудование, состояние ремонтного фонда, квалификация персонала);

- подготовка плана организации технического обслуживания, в котором необходимо определить этапы исполняемых действий, сроки их исполнения, затраты на этапах, ответственность исполнителей.

Управление проектом связано с вопросами планирования и организации работ, создания коллективов разработчиков и контроля сро-

ков и качества выполняемых работ. Техническое и организационное обеспечение проекта включает:

- выбор методов и инструментальных средств реализации проекта;
- определение методов описания промежуточных состояний разработки;
- разработку методов и средств испытаний созданного программного обеспечения;
- обучение персонала.

Обеспечение качества проекта тесно связано с проблемами верификации, проверки и тестирования компонентов информационной системы.

Верификация - это процесс определения соответствия текущего состояния разработки, достигнутого на данном этапе, требованиям этого этапа.

Проверка - это процесс определения соответствия параметров разработки исходным требованиям. Проверка отчасти совпадает с тестированием, которое проводится для определения различий между действительными и ожидавшимися результатами и оценки соответствия характеристик информационной системы исходным требованиям.

На начальном этапе существования компьютерных информационных систем их разработка велась на традиционных языках программирования. Однако по мере возрастания сложности разрабатываемых систем потребовались новые средства, обеспечивающие значительное сокращение сроков разработки. Так появилось новое направление в области программного обеспечения - **инструментальные средства для быстрой разработки приложений**.

Вопросы для самоконтроля

1. Строение информационных систем и их характеристика.
2. Основные процессы информационных систем.
3. Моделью жизненного цикла информационной системы
4. Каскадная модель жизненного цикла информационной системы
5. Спиральная модель жизненного цикла информационной системы

Тема 2.3. Система электронного документооборота

Основные модули

- Система электронного документооборота
- Современная система межведомственного документооборота
- Электронно-цифровая подпись

В настоящее время в Республике Узбекистан придается большое значение внедрению информационно-коммуникационных технологий в государственное управление, в том числе введению систем электронного документооборота в каждом ведомстве и обеспечению межведомственного взаимодействия данных систем. Внедрение систем электронного документооборота в государственные органы осуществляется с целью автоматизации основных этапов жизненного цикла документов, в частности, по формированию, обработке, маршрутизации, контролю движения и исполнения документов, учету, архивному копированию и хранению.

На сегодняшний день система электронного документооборота широко используется в деятельности государственных учреждений. В том числе, в Министерстве юстиции, судах общей юрисдикции и других органах государственной власти и управления. Электронный документооборот представляет собой совокупность процессов отправки и получения электронных документов через информационную систему.

Отметим, что в Республике Узбекистан создана нормативно-правовая база для развития и распространения информатизации и информационно-коммуникационных технологий в области государственного управления, которую составляют законы Республики Узбекистан «Об информатизации», «Об электронной цифровой подписи», «Об электронном документообороте», «Об электронных платежах», «О гарантиях и свободе доступа к информации», а также Указ Президента Республики Узбекистан от 30 мая 2002 года за № УП–3080 «О дальнейшем развитии компьютеризации и внедрении информационно-коммуникационных технологий», постановление Президента Республики Узбекистан от 8 июля 2005 года за № ПП–17 «О дополнительных мерах по дальнейшему развитию информационно-коммуникационных технологий», постановление Кабинета Министров Республики Узбекистан от 6 июня 2002 года за № 200 «О мерах по дальнейшему развитию компьютеризации и внедрению информационно-коммуникационных технологий» и другие нормативные акты. Эти документы создали условия для широкого использования ИКТ и применения электронного

формата делопроизводства в деятельности органов государственных власти и управления.

Создание инфраструктуры систем информатизации является приоритетным направлением в деятельности Министерства юстиции Республики Узбекистан. В министерстве функционирует корпоративная сеть передачи данных, объединяющая территориальные подразделения министерства и подведомственные организаций.

В центральном аппарате министерства функционирует система электронного документооборота «Justice».

Назначением системы электронного документооборота является автоматизация процесса делопроизводства и документооборота с использованием современных средств передачи и хранения информации.

Целями системы электронного документооборота (СЭД) являются:

- автоматизация бумажного документооборота. Отсутствие ранее электронного документооборота не предоставляла возможности комплексного информационно-аналитического обеспечения процессов работы;

- повышение эффективности исполнения процессов путем сокращения непроизводительных и дублирующих операций, операций, выполняемых «вручную», оптимизации информационного взаимодействия участников процессов.

Узбекистан идет большими шагами на пути внедрения ИКТ во все сферы жизни. В прошлом году разработана комплексная программа формирования системы «Электронное правительство», которая охватывает управленческие процессы и процессы оказания государственных услуг бизнесу и гражданам. Создается национальная система, интегрирующая межведомственные и ведомственные комплексы информационных систем. Весь этот процесс невозможен без наличия эффективных систем электронного документооборота.

На сегодняшний день в Узбекистане пройден достаточно большой путь по внедрению систем электронного документооборота. Построена межведомственная информационная сеть обмена электронными документами на уровне Кабинета Министров, аппарата Президента, министерств и ведомств, органов государственной власти на местах – хокимиятов областного, городского и районного уровня.

В соответствии с постановлением Президента Республики Узбекистан от 21 марта 2012 года № ПП–1730 «О мерах по дальнейшему внедрению и развитию современных информационно-коммуникационных технологий» принимаются меры по совершенство-

ванию судебной системы в стране, с внедрением ИКТ в деятельность судов.

Реформа системы правосудия и обеспечения открытости и доступности гражданских судов являются ключевыми компонентами Концепции дальнейшего углубления демократических реформ и формирования гражданского общества в стране, озвученной в 2010 году Первым Президентом Республики Узбекистан И. Каримовым.

В соответствии с положениями и задачами, определенными в данной Концепции, направленными на совершенствование организационной деятельности системы судопроизводства, улучшения доступа к системе правосудия, а также внедрения и использования ИКТ в 2013 году в Узбекистане была разработана и запущена Информационная система электронного судопроизводства (ИСЭС) «E-SUD», предполагающая внедрение электронного документооборота.

Судебным департаментом широко используется Система автоматизации исполнительного производства «E-Sudijro», которая позволяет создать общереспубликанскую единую базу данных исполнительных производств.

Назначением системы «E-Sudijro» является автоматизация процесса исполнения судебных решений, обеспечение надёжного учёта, хранения и обработки документов, автоматизация обработки статистической информации, обеспечение информационной полноты данных в электронном виде и возможность выдачи информации по запросам в систему в реальном масштабе времени, а также контроль за взыскиваемыми судебными исполнителями средствами.

В настоящее время в Республике Узбекистан различные ведомства применяют системы электронного документооборота, выполненные разными производителями программного обеспечения и реализованные на разных платформах. Разноплатформенность СЭД является основным сдерживающим фактором организации эффективного межведомственного электронного документооборота. Документооборот между двумя разными организациями сегодня осуществляется в основном в бумажном виде, что сильно усложняет оперативный обмен служебной информацией.

Учитывая взаимосвязь деятельности органов юстиции, судопроизводства, а также других государственных учреждений, появляется потребность в разработке единой системы межведомственного документооборота, объединяющая использование различных систем государственных органов в едином стандарте, целью которой является оптимизация взаимодействия данных органов власти.

В соответствии с постановлением Президента Республики Узбекистан от 27 июня 2013 года № ПП–1989 на ряд ведомств, в частности Конституционный суд, Верховный суд, Высший хозяйственный суд и Министерство юстиции, возложена задача по созданию комплекса информационных систем «Адлия-2» по сбору, обработке, систематизации и хранению информации о деятельности судов, их решениях, исполнении решений судов, а также информации о деятельности нотариата.

Целью указанного комплекса информационных систем «Адлия-2» является повышение уровня правового обслуживания граждан, с применением современных информационных систем, создание ведомственных баз данных, обеспечение обмена информацией в электронном виде с другими ведомствами в рамках формируемой в стране системы «Электронное правительство».

Комплекс информационных систем «Адлия-2» предназначен для обеспечения обмена информацией между Конституционным судом, Верховным судом, Высшим хозяйственным судом, Министерством юстиции, Департаментом по исполнению судебных решений, материально-техническому и финансовому обеспечению деятельности судов, который осуществляется с помощью интеграции информационных систем судов, Министерства юстиции, Судебного департамента и т.д., для обмена электронными документами, обработки, систематизации и хранения информации о деятельности судов, их решениях, исполнении решений судов, а также информации о деятельности нотариата.

Современная система межведомственного документооборота – это эффективная инфраструктура, позволяющая обеспечить интеграцию различных СЭД за счет введения единого формата электронного документа, передаваемого по межведомственной системе и создать инструменты для эффективного межведомственного документооборота.

Создание системы межведомственного документооборота позволяет:

- сократить объем бумажного межведомственного документооборота и увеличить объем межведомственного электронного документооборота;
- обеспечить высокую степень интеграции государственных ведомств в процессы электронного взаимодействия, а также повысить уровень взаимодействия ведомств с хозяйствующими субъектами;
- повысить уровень контроля исполнений решений правительства;
- обеспечить централизованное хранение документов в электронных архивах системы межведомственного документооборота;
- достичь необходимого уровня безопасности данных, вовлеченных в межведомственный документооборот и обеспечить их кон-

фиденциальность за счет использования сертифицированных систем электронной цифровой подписи, шифрования и аппаратных средств защиты информации;

– повысить эффективность учета и поиска документов за счет использования единых баз данных электронных документов;

– провести мониторинг процедур прохождения документов и поручений, анализ эффективности организации делопроизводства и их оптимизацию за счет осуществления статистического анализа исполнения поручений и формирования электронных отчетов;

– повысить эффективность и оперативность принятия решений и их исполнение государственными ведомствами.

Таким образом, внедрение МСЭД способствует оптимизации функционирования государственных ведомств и повышению эффективности их работы.

Электронно-цифровая подпись (ЭЦП) - это реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа электронной цифровой подписи и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе.

Электронно-цифровая подпись - это программно-криптографическое средство, которое обеспечивает:

- проверку целостности документов;
- конфиденциальность документов;
- установление лица, отправившего документ.

Преимущества использования электронно-цифровой подписи

Использование электронно-цифровой подписи позволяет:

- значительно сократить время, затрачиваемое на оформление сделки и обмен документацией;
- усовершенствовать и удешевить процедуру подготовки, доставки, учета и хранения документов;
- гарантировать достоверность документации;
- минимизировать риск финансовых потерь за счет повышения конфиденциальности информационного обмена;
- построить корпоративную систему обмена документами.

Виды электронно-цифровой подписи

Существует три вида электронной цифровой подписи:

- простая электронно-цифровая подпись;
- усиленная неквалифицированная электронно-цифровая подпись;
- усиленная квалифицированная электронно-цифровая подпись.

Простая электронно-цифровая подпись

Посредством использования кодов, паролей или иных средств, простая электронно-цифровая подпись подтверждает факт формирования электронной подписи определенным лицом.

Простая электронно-цифровая подпись имеет низкую степень защиты. Она позволяет лишь определить автора документа.

Простая электронно-цифровая подпись не защищает документ от подделки.

Усиленная неквалифицированная электронно-цифровая подпись

1) получена в результате криптографического преобразования информации с использованием ключа электронной подписи;

2) позволяет определить лицо, подписавшее электронный документ;

3) позволяет обнаружить факт внесения изменений в электронный документ после момента его подписания;

4) создается с использованием средств электронной подписи.

Усиленная неквалифицированная электронно-цифровая подпись имеет среднюю степень защиты.

Чтобы использовать неквалифицированную электронную подпись, необходим сертификат ключа ее проверки.

Усиленная квалифицированная электронно-цифровая подпись

Для квалифицированной электронной подписи характерны признаки неквалифицированной электронной подписи.

Усиленная квалифицированная электронно-цифровая подпись соответствует следующим дополнительным признакам подписи:

1) ключ проверки электронной подписи указан в квалифицированном сертификате;

2) для создания и проверки электронной подписи используются средства электронной подписи, получившие подтверждение соответствия требованиям законодательства.

Усиленная квалифицированная электронно-цифровая подпись является наиболее универсальной и стандартизированной подписью с высокой степенью защиты.

Документ, визированный такой подписью, аналогичен бумажному варианту с собственноручной подписью.

Использовать такую подпись можно и без каких-либо дополнительных соглашений и регламентов между участниками электронного документооборота.

Если под документом стоит квалифицированная подпись, можно точно определить, какой именно сотрудник организации ее поставил.

Вопросы для самоконтроля

1. Строение информационных систем и их характеристика.
2. Современная система межведомственного документооборота
3. Электронно-цифровая подпись

ГЛАВА III. КОМПЬЮТЕРНАЯ ГРАФИКА

Тема 3.1. Понятие компьютерной графики и их виды

Основные модули

- Компьютерная графика
- Области применения компьютерной графики
- Виды компьютерной графики

Компьютерная графика – это наука, предметом изучения которой является создание, хранение и обработка моделей и их изображений с помощью ЭВМ, т.е. это раздел информатики, который занимается проблемами получения различных изображений (рисунков, чертежей, мультипликации) на компьютере.

В компьютерной графике рассматриваются следующие задачи:

- представление изображения в компьютерной графике;
- подготовка изображения к визуализации;
- создание изображения;
- осуществление действий с изображением.

Под компьютерной графикой обычно понимают автоматизацию процессов подготовки, преобразования, хранения и воспроизведения графической информации с помощью компьютера. Под графической информацией понимаются модели объектов и их изображения.

Области применения компьютерной графики

Область применения компьютерной графики не ограничивается одними художественными эффектами. Во всех отраслях науки, техники, медицины, в коммерческой и управленческой деятельности используются построенные с помощью компьютера схемы, графики, диаграммы, предназначенные для наглядного отображения разнообразной информации. Можно рассмотреть следующие области применения компьютерной графики.

Научная графика. Первые компьютеры использовались лишь для решения научных и производственных задач. Чтобы лучше понять полученные результаты, производили их графическую обработку, строили графики, диаграммы, чертежи рассчитанных конструкций. Первые графики на машине получали в режиме символьной печати. Затем появились специальные устройства – графопостроители (плоттеры) для вычерчивания чертежей и графиков чернильным пером на бумаге.

Деловая графика. Деловая графика – область компьютерной графики, предназначенная для наглядного представления различных

показателей работы учреждений. Программные средства деловой графики включаются в состав электронных таблиц.

Конструкторская графика. Конструкторская графика используется в работе инженеров–конструкторов, архитекторов, изобретателей новой техники. Этот вид компьютерной графики является обязательным элементом САПР (систем автоматизации проектирования). Средствами конструкторской графики можно получать как плоские изображения (проекции, сечения), так и пространственные трехмерные изображения.

Иллюстративная графика. Иллюстративная графика – это произвольное рисование и черчение на экране компьютера. Пакеты иллюстративной графики относятся к прикладному программному обеспечению общего назначения. Простейшие программные средства иллюстративной графики называются графическими редакторами.

Художественная и рекламная графика. Художественная и рекламная графика – ставшая популярной во многом благодаря телевидению. С помощью компьютера создаются рекламные ролики, мультфильмы, компьютерные игры, видеоуроки, видеопрезентации. Графические пакеты для этих целей требуют больших ресурсов компьютера по быстродействию и памяти. Отличительной особенностью этих графических пакетов является возможность создания реалистических изображений и «движущихся картинок».

Компьютерная анимация. Компьютерная анимация – это получение движущихся изображений на экране дисплея. Художник создает на экране рисунок начального и конечного положения движущихся объектов, все промежуточные состояния рассчитывает и изображает компьютер, выполняя расчеты, опирающиеся на математическое описание данного вида движения. Полученные рисунки, выводимые последовательно на экран с определенной частотой, создают иллюзию движения. Мультимедиа – это объединение высококачественного изображения на экране компьютера со звуковым сопровождением. Наибольшее распространение системы мультимедиа получили в области обучения, рекламы, развлечений.

Графика для Интернета. Появление глобальной сети Интернет привело к тому, что компьютерная графика стала занимать важное место в ней. Все больше совершенствуются способы передачи визуальной информации, разрабатываются более совершенные графические форматы, ощутимо желание использовать трехмерную графику, анимацию, весь спектр мультимедиа.

Во время лекции предстоит самостоятельно заполнить предложенную таблицу. Во время подведения итогов правильность заполнения таблицы будет проверена.

Виды компьютерной графики.

Различают четыре вида компьютерной графики. Это *растровая, векторная, трехмерная и фрактальная*. Они отличаются принципами формирования изображения при отображении на экране монитора или при печати на бумаге. Каждый вид используется в определенной области. Растровую графику применяют при разработке мультимедийных проектов. Иллюстрации, выполненные средствами растровой графики, чаще создаются с помощью сканера, а затем обрабатываются специальными программами – графическими редакторами. Программные средства для работы с векторной графикой наоборот предназначены для создания иллюстраций на основе простейших геометрических элементов. В основном применение векторной графики – это оформительские работы. Создание фрактальной художественной композиции состоит не в рисовании, а скорее в программировании. Программные средства для работы с фрактальной графикой предназначены для автоматической генерации изображений путем математических расчетов. Применение – заставки на ТВ автоматической генерации изображений путем математических расчетов. Применение – заставки на ТВ.

Растровая графика. Компьютерное растровое изображение представляется в виде прямоугольной матрицы, каждая ячейка которой представлена цветной точкой. Основой растрового представления графики является пиксель (точка) с указанием ее цвета. Изображение представляется в виде большого количества точек – чем их больше, тем визуально качественнее изображение и больше размер файла. Растровые изображения напоминают лист клетчатой бумаги, на котором любая клетка закрашена либо черным, либо белым цветом, образуя в совокупности рисунок. Пиксель – основной элемент растровых изображений. Именно из таких элементов состоит растровое изображение, т.е. растровая графика описывает изображения с использованием цветных точек (пиксели), расположенных на сетке. При редактировании растровой графики редактируется пиксели, а не линии. Растровая графика зависит от разрешения, поскольку информация, описывающая изображение, прикреплена к сетке определенного размера.

Достоинства: Растровая графика эффективно представляет реальные образы. Растровые изображения могут быть очень легко распечатаны на таких принтерах, потому что компьютерам легко управлять устройством вывода для представления отдельных пикселей с помощью точек.

Недостатки: Большие объемы данных требуют высоких технических характеристик ПК. Память 128 мб и выше, высокопроизводи-

тельный процессор – для обработки, и большой винчестер для хранения. Невозможность увеличения для рассмотрения деталей (пикселизация).

Близкими аналогами являются живопись, фотография

Программы для работы с растровой графикой:

Paint, Adobe Photo Shop

Применение:

- для обработки изображений, требующей высокой точности передачи оттенков цветов и плавного перетекания полутонов.

Например, для:

- ретуширования, реставрирования фотографий;
- создания и обработки фотомонтажа, коллажей;
- применения к изображениям различных спецэффектов;
- после сканирования изображения получаются в растровом виде.

Векторная графика. Основным элементом изображения – линия.

Линия представлена в памяти ПК несколькими параметрами и в этом виде занимает гораздо меньше места, чем растровая линия, состоящая из точек, для каждой из которых требуется ячейка памяти.

Линия – элементарный объект векторной графики. Любой сложный объект можно разложить на линии, прямые или кривые.

Свойства линии

- Форма
- Толщина
- Цвет
- Стиль (пунктир, сплошная)

Замкнутые линии имеют свойство заполнения – цветом, текстурой, узором и т.п. Каждая незамкнутая линия имеет 2 вершины, называемые узлами. С помощью узлов можно соединять линии между собой.

Векторная графика описывает изображения с использованием прямых и изогнутых линий, называемых векторами, а также параметров, описывающих цвета и расположение с помощью математических описаний объектов, окружностей и линий. Такая особенность векторной графики дает ей ряд преимуществ перед растровой графикой, но в тоже время является причиной ее недостатков. Векторную графику часто называют объектно – ориентированной графикой или чертежной графикой. Простые объекты, такие как окружности, линии, сферы, кубы и тому подобное называется примитивами, и используются при создании более сложных объектов

Достоинства векторной графики: малый объем, возможность масштабирования.

Недостатки. Векторное изображение, не содержащее растровых объектов, занимает относительно не большое место в памяти компьютера. Векторный формат становится невыгодным при передаче изображений с большим количеством оттенков или мелких деталей (например, фотографий из-за большого веса файла).

Программы для работы с векторной графикой:

Corel Draw, Adobe Illustrator, AutoCAD

Применение:

- для создания вывесок, этикеток, логотипов, эмблем и пр. символьных изображений;
- для построения чертежей, диаграмм, графиков, схем;
- для рисованных изображений с четкими контурами, не обладающих большим спектром оттенков цветов;
- для моделирования объектов изображения;
- для создания 3-х мерных изображений

Трехмерная графика. С точки зрения компьютера трехмерные объекты – это лишь пустотелые, не имеющие физической толщины оболочки. как от каждой их точки отражаются в направлении глаза наблюдателя воображаемые световые лучи, испускаемые заданными в составе сцены источниками света

Для создания реалистичной модели объекта используют геометрические примитивы (прямоугольник, куб, шар, конус и прочие) и гладкие поверхности. Вид поверхности при этом определяется расположенной в пространстве сеткой опорных точек

Достоинства: Большие возможности для поддержки технического черчения. С помощью графических редакторов трёхмерной компьютерной графики можно выполнять наглядные изображения деталей и изделий машиностроения, а также выполнять макетирование зданий и архитектурных объектов, изучаемых в соответствующем разделе архитектурно-строительного черчения. Наряду с этим может быть осуществлена графическая поддержка таких разделов начертательной геометрии, как перспектива, аксонометрические и ортогональные проекции, т.к. принципы построения изображений в трёхмерной компьютерной графике частично заимствованы из них. Для декоративно-прикладного искусства трёхмерная компьютерная графика предоставляет возможность макетирования будущих изделий с передачей фактуры и текстуры материалов, из которых эти изделия будут выполнены.

Недостатки:

- повышенные требования к аппаратной части компьютера, в частности к объему оперативной памяти, наличию свободного места на жестком диске и быстродействию процессора
- необходимость большой подготовительной работы, но созданию моделей всех объектов сцены, которые могут попасть в поле зрения камеры
- необходимость контроля за взаимным положением объектов в составе сцены,
- особенно при выполнении анимации. В связи с тем, что объекты трехмерной графики «бестелесны», легко допустить ошибочное проникновение одного объекта в другой или ошибочное отсутствие нужного контакта между объектами

Программы для работы с трехмерной графикой:

3D Studio MAX 5, AutoCAD, Компас

Применение:

- научные расчеты,
- инженерное проектирование,
- компьютерное моделирование физических объектов
- изделия в машиностроении,
- видеороликах,
- архитектуре

Фрактальная графика. Фрактальная графика – одна из быстроразвивающихся и перспективных видов компьютерной графики.

Фрактал – структура, состоящая из частей, подобных целому. Одним из основных свойств является самоподобие. (Фрактус – состоящий из фрагментов).

Фрактальная графика, как и векторная вычисляемая, но отличается тем, что никакие объекты в памяти не хранятся. Изображение строится по уравнению, или системе уравнений, поэтому ничего кроме формулы хранить не надо. Изменив коэффициенты, можно получить совершенно другую картину.

Таким образом, мелкие объекты повторяют свойства всего объекта. Процесс наследования можно продолжать до бесконечности.

Полученный объект носит название – **фрактальной фигуры**.

Абстрактные композиции можно сравнить со снежинкой, с кристаллом.

Фрактальная графика основана на математических вычислениях. Базовым элементом фрактальной графики является сама математиче-

ская формула, то есть никаких объектов в памяти компьютера не хранится и изображение строится исключительно по уравнениям.

Фрактальными свойствами обладают многие объекты живой и неживой природы. (снежинка, ветка папоротника)

Способность фрактальной графики моделировать образы вычислительным путем часто используют для автоматической генерации необычных иллюстраций.

Программа для работы с фрактальной графикой:

Фрактальная вселенная 4.0 fracplanet

Применяют:

- Математики, Художники

Форматы графических файлов.

В настоящее время существует множество различных форматов хранения графической информации. Условно можно разделить все форматы на три группы

- графические метафайлы (метафайл обычно разрабатывается как составная часть какой-либо графической системы, например *.wmf в Windows)

- растровые графические файлы
- векторные графические файлы

Вопросы для самоконтроля

1. В чем преимущества растровой графики?
2. В чем недостатки векторной графики?
3. Основной элемент векторной графики...
4. Как называется графика с представлением изображения в виде совокупности объектов
5. Назовите программы для работы с трехмерной графикой

Тема 3.2. Введение в программу Corel DRAW. Возможности программы и её интерфейс

Основные модули

- Интерфейс окна. Документы Corel DRAW
- Специальные эффекты.
- Заливки: градиентная, узорная, текстура. Прозрачность.
- Создание объектов по отрасли используя программу Corel DRAW.

Corel DRAW Graphics Suite содержит все необходимые графические инструменты для выполнения следующих операций (рис.3.2.1.)

- Создание иллюстраций с помощью мощных инструментов для работы с векторной графикой;
- Верстка проектов;
- Редактирование и улучшение качества цифровых изображений;
- Преобразование растровых изображений в файлы векторной графики.

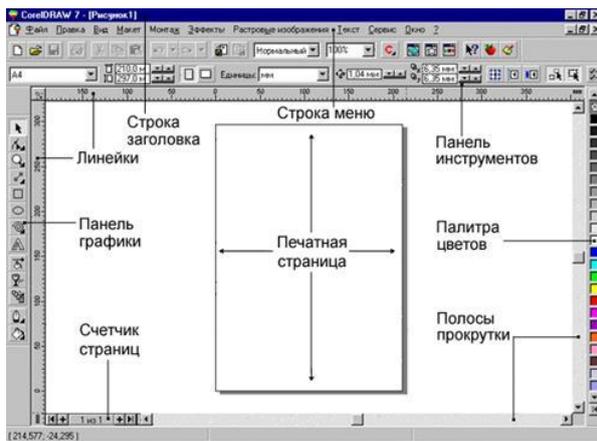


Рис.3.2.1. Рабочее окно CorelDraw

Профессиональное программное обеспечение CorelDRAW для редактирования фотографий помогает ретушировать и улучшать фотографии.

Растровые изображения с CorelDRAW можно легко преобразовать в редактируемые и масштабируемые векторные файлы. Каким бы

ни был ваш проект, CorelDRAW упростит рабочий процесс и вдохновит вас новыми возможностями для творчества. В состав CorelDRAW входят редактор векторной графики CorelDRAW, редактор растровой графики PHOTO-PAINT, программа для преобразования растровых изображений в векторные CorelPowerTRACE, программа для создания скриншотов Corel CAPTURE.

CorelDRAW сочетает в себе полный набор великолепных возможностей дизайна, скорость и простоту в использовании. CorelDRAW предоставляет полный набор инструментов для создания логотипов, иллюстраций, брошюр, бюллетеней, рекламных объявлений, буклетов и рекламных щитов и является оптимальным графическим решением.

Работа в CorelDRAW

Запускаем программу нажатием на ярлык с рабочего стола, либо через **Пуск - CorelDRAW(Пуск – Все программы - CorelDRAW)** (рис.3.2.2.).

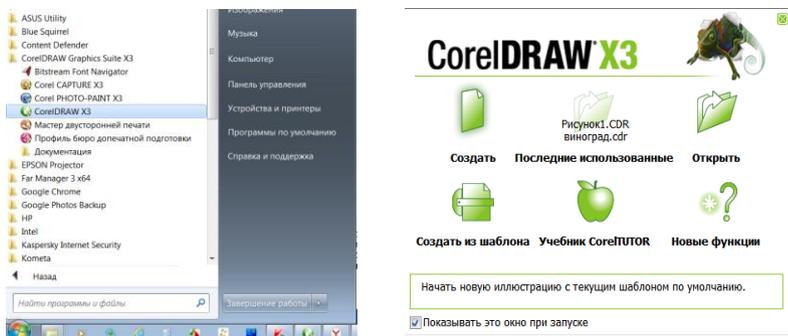


Рис.3.2.2. Работа в CorelDRAW

Сразу при старте программы, выводится окно для выполнения наших действий (по умолчанию). Из этого окна мы можем начать делать новый проект, открыть существующий и выполнить другие действия. Если данное окно не отобразилось, то с помощью Файл – Новый создаем новый проект. По умолчанию проект располагается на листе А4, поэтому в окне создания документа все оставляем без изменений и нажимаем кнопку ОК.

С помощью инструмента Прямоугольник, изображаем квадрат и поворачиваем его, чтобы он принял вид ромба. Таким же образом рисуем второй квадрат, поворачиваем, уменьшаем его в размере и помещаем внутрь первого ромба. На углах, между ромбами рисуем 4 небольших круга (рис.3.2.3.)

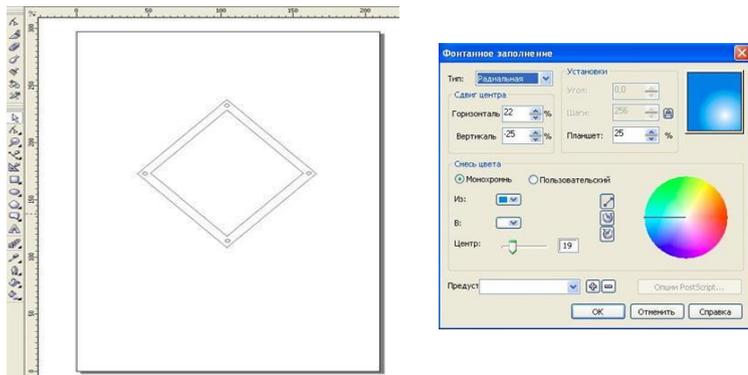


Рис.3.2.3. Работа в CorelDRAW

Заполняем все фигуры цветом, при этом, для большой фигуры ставим обводку контура толщиной примерно 4,0 т. В центре маленького ромба изображаем букву D для обозначения компании. Текст вводится с помощью инструмента . Для заливки фигур используем градиентную заливку. Пример для маленького ромба (рис.3.2.4.).

Получаем:



Рис.3.2.4. Работа в CorelDRAW

По левому и правому краю фигуры пишем название компании и род деятельности, используя шрифт . Слово DOT заполняем градиентной заливкой и делаем обводку контура. Используя инстру-

мент Интерактивный Объем  вытягиваем фигуры, чтобы они приняли объемный вид (рис.3.2.5.).



Рис.3.2.5.-3.2.6. Работа в CorelDRAW

Далее, с помощью инструмента Интерактивная тень, добавляем эффект тени и получаем готовый логотип (рис.3.2.6.)

Для того, чтобы все элементы были как единое целое, а не отдельные фигуры, нужно их сгруппировать. Выделяем все элементы и нажимаем кнопку Группа  расположенную на верхней панели.

Построение прямоугольников

В CorelDRAW все примитивы строятся с помощью соответствующих инструментов. Среди инструментов имеется кнопка Rectangle (Прямоугольник). Попробуем воспользоваться этим инструментом и построить наш первый объект.

1. Создайте новый документ. Для этого упражнения можно выбрать лист бумаги с произвольной ориентацией любого размера.

2. Щелкните на кнопке инструмента Rectangle (Прямоугольник) в наборе инструментов. После этого указатель мыши на экране примет форму перекрестья с прямоугольником – это визуальное подтверждение того, что в настоящий момент активен инструмент построения прямоугольников.

3. Чтобы построить прямоугольник, перетащите указатель инструмента Rectangle (Прямоугольник) по диагонали создаваемого объекта. Обратите внимание, что в процессе перетаскивания указателя мыши в строке состояния выводятся текущие значения высоты и ширины прямоугольника, а на экране отображается его постоянно меняющийся абрис. В момент отпускания кнопки мыши при окончании перетаскивания на экране появляется прямоугольник в окружении маркеров выделения и с маркером центра, а в строке состояния - «общение о том, что выделен объект, относящийся к классу прямоугольников.

Аналогично с помощью соответствующих инструментов строятся другие фигуры.

Кривая Безье

В основе принятой в CorelDRAW модели линий лежат два понятия: узел и сегмент. Узлом называется точка на плоскости изображения, фиксирующая положение одного из концов сегмента. Сегментом называется часть линии, соединяющая два смежных узла. Узлы и сегменты неразрывно связаны друг с другом: в замкнутой линии узлов столько же, сколько сегментов, в незамкнутой – на один узел больше.

Элементы кривой Безье: 1- узлы, 2 - сегменты, 3- направляющие

Любая линия в CorelDRAW состоит из узлов и сегментов, и все операции с линиями на самом деле представляют собой операции именно с ними. Узел полностью определяет характер предшествующего ему сегмента, поэтому для незамкнутой линии важно знать, который из двух ее крайних узлов является начальным, а для замкнутой – направление линии (по часовой стрелке или против нее). По характеру предшествующих сегментов выделяют три типа узлов: начальный узел незамкнутой кривой, а также прямолинейный (line) и криволинейный (curve) узлы.

Для узлов, смежных хотя бы с одним криволинейным сегментом, имеется еще одна классификация типов: они подразделяются на точки излома (cusp) и сглаженные узлы (smooth). Частным случаем сглаженного узла является узел симметричный (symmetrical), но таким может быть только узел, расположенный между двумя криволинейными сегментами.

Со стороны примыкания к выделенному узлу криволинейного сегмента отображается так называемая направляющая точка. На экране она показана в виде зачерненного квадратика, соединенного с узлом штриховой линией. Эта штриховая линия совпадает с касательной к криволинейному сегменту в точке его вхождения в узел. Чем дальше направляющая точка располагается от узла, тем медленнее криволинейный сегмент отклоняется от касательной по мере удаления от узла. При выделении узла, разделяющего два криволинейных сегмента, на экране отображаются четыре направляющих точки - с обоих концов каждого сегмента.

Тип узла (он отображается в строке состояния) определяет возможное взаимное расположение его направляющих точек.

Пристыковываемые окна

Пристыковываемые окна (dockers) могут быть использованы для выполнения широкого круга функций: от перемещения объектов и расположения их по слоям, до показа коллекций веб-ссылок.

Пристыковываемое окно Transformation (Преобразование)

Как правило, каждое пристыковываемое окно содержит несколько вкладок. Например, у пристыковываемого окна Transformation (Преобразование), изображенного на рисунке, их пять. Для перехода на нужную вкладку, нужно щелкнуть на соответствующем значке в верхней части пристыковываемого окна.

Трассировка

Трассировка - это преобразование пиксельного изображения в векторное. Преобразование векторного изображения в пиксельное (растрирование или рендеринг) представляет собой достаточно простой, формальный процесс. Почти всегда векторное изображение перед выводом или в процессе вывода на печать преобразуется в точечное, в компьютерной графике этот процесс называется – рендерингом.

Преобразование пиксельного изображения в векторное (векторизация или трассировка), в подавляющем большинстве случаев требует не просто вмешательства, а творческого участия.

Графический редактор CorelDRAW позволяет выполнить операцию векторизации автоматически и вручную. Для достижения наилучшего результата векторизации (особенно автоматической), старайтесь использовать исходные пиксельные файлы большого разрешения (250-300 dpi) с четкими ровными краями изображения.

Автоматическая векторизация

Для перехода к автоматической векторизации импортированного пиксельного изображения следует выделить его инструментом Pick (Выбор) и выбрать команду Bitmaps > Trace Bitmap (Пиксельное изображение > Трассировать пиксельное изображение). После нажатия в открывшемся окне программы мы видим поле поделенное на две части, в левой части исходный растровый объект, правая часть предназначена для получения векторного результата.

В верхней панели выбираем позицию трассировка, здесь предложены различные варианты выполнения трассировки, лучше всего добиться желаемого результата опытным путем.

В графе точность необходимо выставить цифровое значение от 1 до 100 – шаг векторизации. Чем выше цифровое значение, тем лучше полученный результат. Но однако нужно помнить, что если исходное пиксельное изображение многоцветное и имеет сложную форму, то в результате при выставленном максимуме шага (100), можно получить слишком большое количество векторных «узлов», из за которых полученный файл будет иметь большой вес. Нужно опытным путем пытаться найти такое оптимальное значение, при котором в результате векторизации сохраняются все элементы и силуэтная форма объекта, но при этом файл не имеет лишнее количество «узлов».

Визуально сравниваем исходное и полученное изображения, и если результат устраивает нажимаем на "ОК". Нужно заметить, что полученный результат для дальнейшей работы необходимо будет разгруппировать (Ctrl-U).

Далее можно удалить ненужные мелкие детали, которые могли остаться после трассировки. При помощи инструмента «указатель» (стрелка), выделяем объект, который хотим удалить и после его выделения нажимаем на клавиатуре клавишу Delete. Для того чтобы видеть неровности линий и количество мелких деталей нужно перевести изображение в режим «Каркас» в этом режиме показано бесцветное, контурное изображение объектов и становятся видны все наложения и пересечения.

У метода автоматической векторизации есть существенный недостаток. Как правило объекты, полученные методом трассировки при помощи Corel Trace имеют неровные, рваные края. Сгладить их можно при помощи инструмента «Форма».

Выбрав инструмент, «Форма» (треугольная стрелка) нажать на объект и удерживая левую кнопку мыши выделить (обвести) редактируемые узлы после выделения удалить клавишей Delete. Этим же инструментом можно выправить линию формы, для этого нужно подвести курсор к краю объекта и удерживая левую клавишу сдвигать указатель в нужную сторону задавая форму. Таким образом можно отредактировать всё изображение.

Векторизация вручную. Обрисовка

В некоторых случаях наиболее оптимальным не только по качеству получаемого результата, но и по затратам времени может оказаться метод векторизации вручную. При этом методе человек по мере работы не только решает, какие детали изображения важны, а какие – не очень, но и структурирует изображение, строя объекты так, чтобы с ними было удобнее работать. Метод заключается в том, что поверх пиксельного изображения строятся замкнутые кривые.

Далее ранее построенным объектам назначаются цвета заливки, а также параметры контурных линий. По завершении этого этапа в вашем распоряжении оказывается полнофункциональное векторное изображение.

Вопросы для самоконтроля

1. Создание объектов по отрасли используя программу Corel DRAW.
2. Специальные эффекты.
3. Заливки: градиентная, узорная, текстура. Прозрачность.

Тема 3.3. Автоматизированные системы проектирования в строительстве.

Общие сведения программы AutoCAD.

Основные модули

- Основные понятия AutoCAD
- Описание окна AutoCAD

1. Примитив

Примитивы - это стандартные геометрические элементы, при помощи которых строятся изображения в AutoCAD. Дело в том, что Автокад работает не с изображением как таковым, а с геометрическим описанием объектов, составляющих изображение, что обусловлено задачами САПР. У такого представления есть как преимущества, так и недостатки. С одной стороны, графическое представление изделия с его геометрическим описанием более компактно и позволяет производить различные геометрические преобразования, а также напрямую использовать такое описание в автоматизированных системах технологической подготовки производства. С другой стороны, мы не можем рисовать кривые произвольной формы. Однако геометрического представления достаточно для создания любого технического изображения. Такое представление данных называется векторным представлением в отличие от пиксельного описания картинка как поля цветных или черно-белых точек экрана.

Все примитивы AutoCAD обладают рядом свойств (принадлежность слою, цвет, тип линии, ширина).

2. Система координат

В AutoCAD используется стандартная система декартовых координат. Кроме того пользователю предоставляется возможность определять свои собственные системы координат. Такие введенные пользователем системы координат называют пользовательскими системами координат – ПСК (UCS).

Таким образом, работая с рисунком можно пользоваться разными системами координат. Однако в каждый момент времени пользователь работает только с одной предварительно выбранной системой координат, которая называется текущей. Вся работа с изображением осуществляется в текущей системе координат. В левом нижнем углу графической зоны AutoCAD постоянно показывает пиктограмму текущей системы координат.

3. Единицы измерения

Расстояния между точками на рисунке измеряются в условных единицах. Конкретный формат представления размеров (дюймы, сантиметры, миллиметры) не имеет значения для AutoCAD. Иначе говоря, при создании объектов в чертеже AutoCAD измеряет все расстояния в относительных единицах. Соответствие между условными единицами AutoCAD и конкретной системой (метрической, дюймовой) устанавливается выбором формата представления.

4. Вид

Когда создается чертеж на AutoCAD, работа ведется с изображением части (или всего) чертежа выводимого на дисплей. Эту часть изображения называют видом. Эту видимую часть чертежа (окно зрения) можно увеличивать (при этом изображение чертежа будет уменьшаться), уменьшать (изображение будет увеличиваться) или перемещать по полю чертежа без изменения масштаба отображения (панорамирование).

5. Слой

AutoCAD даёт возможность распределять выбранные фрагменты чертежа по различным слоям. Количество слоёв не ограничено. Слои можно делать видимыми и невидимыми. С каждым слоем чертежа связывается цвет и тип линий. По мере создания чертежа можно вводить новые слои и менять свойства уже существующих. При создании слоя ему присваивается имя, отличающее его от других слоёв. Подобно системе координат в любом чертеже AutoCAD всегда существует по крайней мере один слой с именем “0”.

6. Чертёж

Чертёж – это файл с информацией, описывающей графический объект. AutoCAD позволяет создавать и редактировать чертеж множеством различных способов. Любое изображение создается при помощи базового набора примитивов. AutoCAD предоставляет в распоряжение пользователя все вычислительные ресурсы ЭВМ, новые средства создания и редактирования чертежа. Помимо высокой точности построений (задаваемой пользователем), AutoCAD позволяет упростить геометрические построения, определяя геометрические характерные точки объектов, предоставляет информацию о чертеже (координаты, расстояния, площади).

7. Блоки

Примитивы могут быть простыми, сложными и составными. Составные примитивы в AutoCAD называют блоками. Перечислим преимущества использования блоков:

1. Блоки хранятся отдельно от остального чертежа, и один и тот же блок может использоваться в чертеже многократно, что позволяет сократить время создания чертежа, упростить редактирование и т.д.

2. Блок можно вставлять в разрабатываемый чертёж под любым углом и в любом масштабе. Это позволяет создавать изображения из любых символов.

3. Есть возможность создания библиотеки постоянно используемых блоков, что значительно ускоряет процесс проектирования.

4. Блоки отличаются от других примитивов тем, что имеют имя. Поэтому, определив новый блок с тем же именем, что и старый мы заменим блок во всём чертеже (при этом старый блок будет удалён).

8.Атрибут

Атрибут – специальный примитив AutoCAD, предназначенный для работы с блоками. Будучи записан в блок, он служит в качестве текстовой переменной, в которую при вставке блока можно записать некоторую строку. Это позволяет, однажды создав блок, в который входит атрибут, с каждой вставкой блока в чертёж связывать новую текстовую строку. Атрибуты особенно удобны при создании различных схем, в которых один и тот же графический символ отрисовывается несколько раз с разными обозначениями

Описание окна AutoCAD

Заголовок окна располагается в самом верху окна. Он содержит имя файла, с которым идет работа, и используется для изменения положения окна на экране. В правом углу заголовка расположены кнопки свертывания, развертывания и закрытия окна. Свернутое окно отображается как кнопка на панели задач.

Большую часть экрана занимает окно документа (рис.3.3.1.).

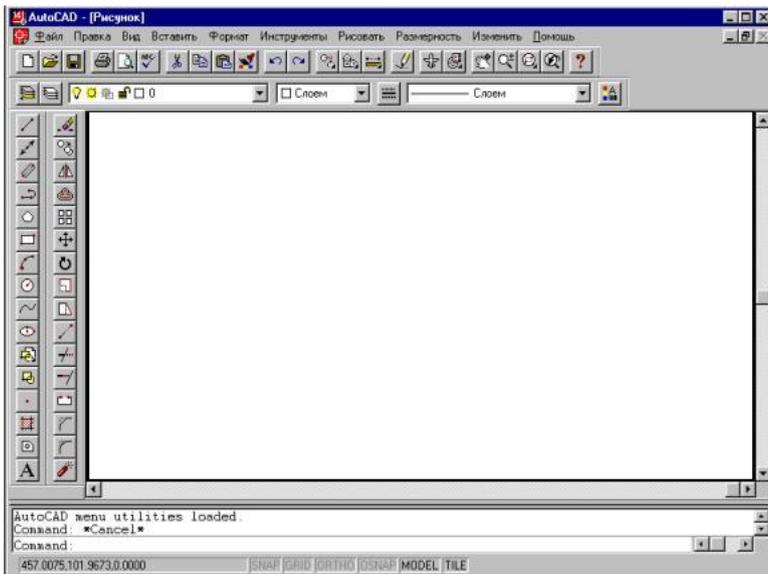


Рис.3.3.1. 1. Окно документа

Главное меню расположено вверху экрана сразу под заголовком окна. Оно содержит названия выпадающих меню. Если щелкнуть на название меню, на экране появится список команд, предназначенных для доступа к различным функциям AutoCAD.

Строка состояния и командная строка располагаются внизу экрана. Командная строка – это самая главная зона графического экрана. В ней отображается весь диалог с AutoCAD, вне зависимости от того, как выбирается команда. В строке состояния отображаются сведения о выделенном объекте или команде.

Панель графики обеспечивает доступ инструментам, служащим для создания объектов разных форм, изменения заданных объектов.

При работе в системе AutoCAD также часто используются диалоговые окна (рис.3.3.2.).

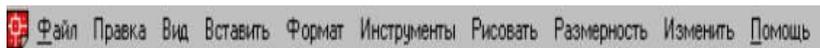


Рис. 3.3.2. Главное меню

Как уже говорилось выше главное меню содержит названия ниспадающих меню. Если щелкнуть на название меню, на экране появится список команд, предназначенных для доступа к различным функциям AutoCAD. Многие пункты подменю также являются корнями подменю более низкого уровня.

Пункт главного **меню Файл** открывает ниспадающее меню, опции которого предназначены для непосредственной работы с файлами. Все опции данного меню имеют свои подменю или вызывают диалоговые окна (рис.3.3.3.).

- команда **Новый** меню **Файл** служит для создания нового чертежа.
- команда **Открыть** меню **Файл** служит для открытия уже созданного чертежа.
- команды **Сохранить** и **Сохранить как** меню **Файл** служат для сохранения созданного чертежа.
- команды **Экспорт** и **Отправить** меню **Файл** служат для передачи данных, находящихся в разных файлах.
- команды **Настройки принтера** и **Печать** меню **Файл** служат для настройки печатающего устройства и передачи чертежей, созданных в системе AutoCAD на печать.
- команда **Предварительный просмотр** меню **Файл** служит для просмотра расположения чертежа на бумаге заданного формата.
- команда **Выход** меню **Файл** служит для завершения работы в системе AutoCAD.

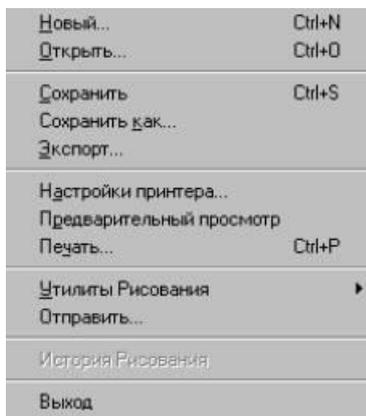
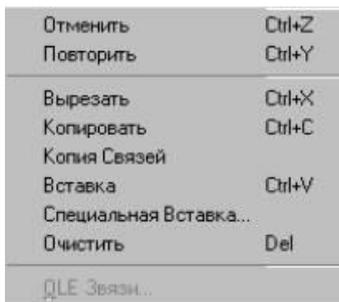


Рис. 3.3.3. Меню Файл

Пункт главного **меню Правка** открывает ниспадающее меню, опции которого предназначены для редактирования создаваемых изображений (рис.3.3.4).

- команды Копировать, вырезать, вставить, Специальная вставка в меню Правка служат для копирования, удаления и помещения выделенных объектов в буфер обмена Windows.
- команды Отменить и повторить меню Правка служат для отмены или повтора каких-либо команд.



Отменить	Ctrl+Z
Повторить	Ctrl+Y
Вырезать	Ctrl+X
Копировать	Ctrl+C
Копия Связей	
Вставка	Ctrl+V
Специальная Вставка...	
Очистить	Del
OLE Связи...	

Рис. 3.3.4. Меню Правка

Пункт главного **меню Вид** открывает ниспадающее меню, опции которого предназначены для изменения режимов просмотра создаваемых изображений и настройки системы меню и панели инструментов. Команды этого меню помогают изменять размеры изображения, его ориентацию в пространстве. Создавать и просматривать именованные виды чертежа. Многие опции содержат подменю или вызывают диалоговые окна (рис.3.3.5).

- команда Перерисовать выполняет перерисовку экрана – очищает экран от маркеров и от графического “мусора”, остающегося после выполнения команд редактирования.
- команды Регенерация и регенерировать всё полностью перестраивают изображение после редактирования геометрического описания чертежа. После регенерации всегда автоматически выполняется перерисовка изображения на экране.
- команда Панорама позволяет, не изменяя масштаба отображения чертежа на экране, перемещать его относительно экрана на заданный вектор.
- команды Форточка и Вид с воздуха позволяют управлять размером вида на экране.

- команда Названные Виды открывает диалоговое окно для работы с именованными видами изображения.
- при помощи команд 3D Окно и 3D Динамический Вид AutoCAD предоставляет богатые возможности для работы в трёхмерном пространстве – выбор точки зрения с обработкой проекций, выполнение сечений в трёхмерном пространстве и т.д.
- команда Панель Инструментов предназначена настройки системы меню и панели инструментов.

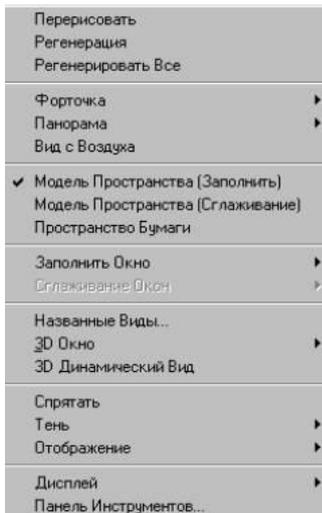


Рис. 3.3.5. Меню Вид

Пункт главного меню **Вставить** открывает ниспадающее меню, опции которого предназначены для вставки в чертеж внешних по отношению к системе AutoCAD объектов (3D Studio, ACIS Solid, мета-файлов Windows) и блоков из другого чертежа. Все опции вызывают диалоговые окна для выбора соответствующих объектов (рис.3.3.6.).

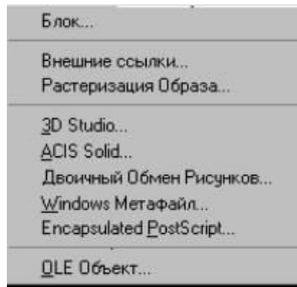


Рис. 3.3.6. Меню Вставить

Пункт главного **меню Формат** открывает ниспадающее меню, опции которого предназначены для задания свойств различным объектам системы AutoCAD (рис.3.3.7.).

- команда **Слой** меню **Формат** открывает диалоговое окно для создания и работы с уже созданными слоями чертежа.
- команда **Цвет** меню **Формат** открывает диалоговое окно для задания цвета объекта чертежа.
- команда **Тип линии** меню **Формат** открывает диалоговое окно для выбора шаблона, по которому обрисовываются линии объекта чертежа.
- команда **Стиль текста** меню **Формат** открывает диалоговое окно для задания стиля выводимого на чертеже текста.
- команды **Стиль разграничений**, **Стиль отметок**, **Стиль мультлиний** меню **Формат** открывают диалоговые окна для задания стиля выводимых на чертеже разграничений, отметок, мультлиний.

Пункт главного **меню Рисовать** открывает ниспадающее меню, опции которого предназначены для создания различных примитивов системы AutoCAD. Команды данного меню дублируются в виде пиктограмм, расположенных на панели графики (таблица 3.3.1.).

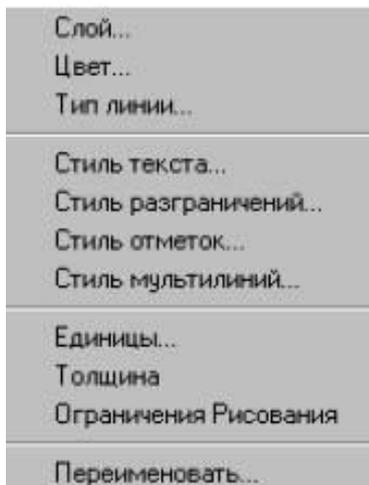


Рис. 3.3.7. Меню Формат

Таблица 3.3.1.

Инструмент	Характеристика изображения
Точка (Point)	Простейший примитив AutoCAD. Точка определяется координатами (X,Y,Z). Точки могут изображаться на экране дисплея различными графическими знаками.
Отрезок (Line)	Часть прямой линии, определяемая двумя крайними точками. Отрезок в AutoCAD имеет нулевую толщину и изображается на экране размером раstra.
Фигура (Solid)	Часть плоскости, ограниченная 4(3) отрезками, определяемыми по 4 точкам, 2 из которых могут совпадать. На отрисовку фигуры влияет установка режима закрашивания.
Прямоугольник	Рисование прямоугольников и квадратов, т.е. фигур, ограниченных 4 отрезками, расположенными под углом 90°.
Полоса (Trace)	Примитив того же типа, что и фигура. Он определен в системе по 4 точкам, однако отрисовывается как отрезок заданной ширины.
Эллипс (Ellips)	Рисование эллипсов и окружностей. Отрисовывается в AutoCAD одной полилинией, сегменты которой являются дугами двенадцатицентрового овала.
Дуга (Arc)	Часть окружности, определяемая центром, радиусом

Инструмент	Характеристика изображения
	и двумя центральными углами. Дугу в AutoCAD можно построить 10 способами.
Кольцо (Donut)	Объект, создаваемый одной замкнутой полилинией, состоящей из одного имеющего ширину дугового сегмента.
Круг (Circle)	Часть плоскости, ограниченная окружностью. Задается центром и радиусом.
Полилиния (Polyline)	Составная линия, включающая в себя прямолинейные и дуговые сегменты. Её особенность в том, что её сегменты могут иметь не только постоянную, но и переменную ширину.
Многоугольник (Polygon)	Объект, создаваемый одной замкнутой полилинией, состоящей из прямолинейных сегментов равной длины.
3М Грань (3D Face)	Часть плоскости, ограниченная четырьмя отрезками. Основным свойством является непрозрачность грани.
3М Сеть (3D Mesh)	Трёхмерный объект, определяемый двухпараметрическим массивом вершин. Трёхмерная сеть является развитием трёхмерной грани и может быть расчленена на самостоятельные трёхмерные грани.
Текст (Text)	Примитив, являющийся совокупностью строки текста и специфических для текста свойств: гарнитуры, угла наклона, точки вставки и т.д. Работая с текстом, можно выполнять следующее: выбирать шрифт и задавать его размер и другие атрибуты; задавать интервалы между символами, словами, строками и абзацами.
Граница	Используется для изменения контуров выделенных объектов. Открывает для этого соответствующее диалоговое окно.
Тела	Используется для выбора из соответствующего диалогового окна необходимых геометрических тел.
Блок (Block)	Составной примитив, сформированный из других примитивов и имеющий имя.

Пункт главного меню **Размерность** открывает ниспадающее меню, опции которого предназначены для нанесения размеров на чертежи, задания атрибутов текста размеров, выносных линий, стрелок и т.д (рис.3.3.8.).

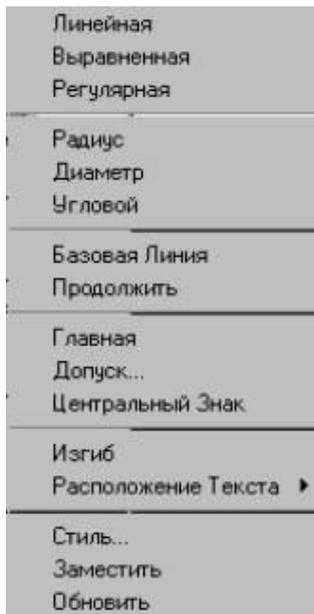


Рис. 3.3.8. Меню Размерность

AutoCAD позволяет наносить различные линейные, угловые размеры, радиусы, диаметры и т.д. в соответствии с нормами международной системой ЕСКД. Допускается привязка к базовой линии. Значения размеров вычисляется самой системой и выводится в заранее заданных единицах измерениях.

Пункт главного меню **Изменить** открывает ниспадающее меню, опции которого предназначены для внесения изменений в создаваемый чертеж, преобразования объектов, изменения стилей, свойств и атрибутов объектов AutoCAD (рис.3.3.9.).

Преобразование положения объекта в программе AutoCAD означает изменение его ориентации или внешнего вида без изменения его исходной формы. Для изменения положения объекта используются команды **Двигать** и **Повернуть**.

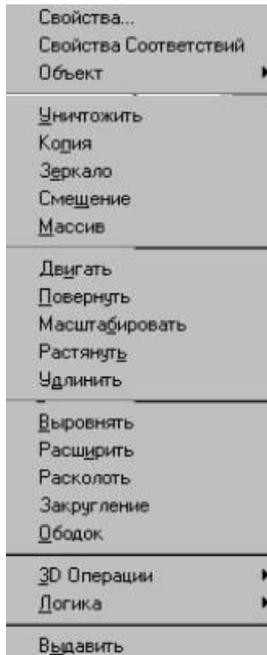


Рис. 3.3.9. Меню Изменить

Преобразование объекта в программе AutoCAD означает изменение его исходной формы. Для этого служат команды Выровнять, расширить, растянуть, удлинить, расколоть, Выдавить.

Программа AutoCAD также позволяет создавать копии объектов и преобразовывать их, оставляя исходные объекты неизменными. Для этого служат команды Копия, Зеркало, Массив.

К объекту или группе объекта можно применять любое количество операций по преобразованию.

- команда Свойства изменяет свойства созданных примитивов – цвет, принадлежность слою, тип линии, высоту и т.д.
- команда Уничтожить удаляет выделенный объект.
- команда Копия копирует выделенные объекты, оставляя оригиналы не тронутыми, и размещает копии в заданном месте или на заданном расстоянии от оригинала.
- команда Зеркало позволяет формировать зеркальные отражения существующих объектов, удаляя или сохраняя при этом оригиналы.

- команда **Смещение** обеспечивает плоскопараллельный перенос одного или нескольких объектов в указанное место.
- команда **Массив** позволяет получить несколько копий объектов, группируя их в прямоугольной или круговой матрице.
- команда **Повернуть** поворачивает заданную группу объектов вокруг заданной точки на заданный угол.
- команда **Масштабировать** изменяет размеры одного или нескольких объектов. Изображение при этом изменяется относительно заданной базовой точки.
- команда **Растянуть** обеспечивает перемещение выбранной части изображения, сохраняя при этом связь с остальной частью.
- команда **Удлинить** удлиняет отрезки, дуги и двумерные полилинии до пересечения с аналогичными примитивами.
- команда **Закругление** плавно сопрягает отрезки, дуги и окружности дугами разного, в том числе и нулевого, радиуса, а также сопрягает полилинии. При этом “лишние” части примитивов автоматически удаляются.
- команда **Ободок** проводит линию фаски, удаляя ненужные части примитивов.

Пункт главного **меню Помощь** открывает ниспадающее меню, опции которого предназначены для предоставления пользователю в любой момент времени дополнительной справочной информации:

- о наборе команд AutoCAD.
- о каждой команде в отдельности – контекстная справка.
- о геометрических характеристиках объектов – координатах точек, углах.
- о площади, периметре и другой вспомогательной вычисляемой информации.

Примитивы

В отличие от "художественных" графических редакторов, AutoCAD работает не с изображением как таковым, а с геометрическим описанием объектов, составляющих изображение. Так, например, отрезок во внутреннем представлении графического редактора AutoCAD описывается двумя точками, круг описывается центром и радиусом.

Все примитивы AutoCAD обладают рядом свойств (принадлежность слою, цвет, тип линии, ширина). Некоторые из этих свойств (например, цвет) присущи всем примитивам.

Системы координат

Используется традиционная декартова система координат. Можно ввести пользовательские системы координат с помощью команды USC. В определенный момент времени пользователь работает только с одной предварительно выбранной системой координат, которая называется текущей. Вся работа с изображением проводится в текущей системе координат.

Единицы измерения и масштаб

Расстояния между точками на рисунке измеряются в условных единицах. Конкретный формат представления размеров (дюймы, футы, миллиметры и др.) не имеет значения для AutoCAD. Иначе говоря, при создании объектов в чертеже AutoCAD "измеряет" все расстояния в относительных единицах. В AutoCAD нет масштаба в обычном понимании конструктора, конструктор задает все расстояния и координаты в реальных единицах - мы как бы работаем в масштабе 1:1. Масштабирование различных частей изображения в соответствии с желаемым форматом документа может осуществляться в момент компоновки чертежа (команда SCALE (МАСШТАБ)) или при выводе чертежа или его части на плоттер (принтер).

Вид

Когда вы создаете чертеж, то работаете с изображением части чертежа, выводимой на дисплей. Будем называть эту часть изображения видом. Эту видимую часть чертежа (окно зрения) можно увеличивать (при этом изображение чертежа будет уменьшаться), уменьшать (изображение будет увеличиваться) или перемещать по полю чертежа без изменения масштаба отображения (панорамирование). Изменение вида осуществляется командой ZOOM.

Слой

AutoCAD дает возможность распределять выбранные фрагменты чертежа по различным слоям. Работая за кульманом, конструктор имеет дело с одним листом бумаги и располагает изображения объектов на нем и только на нем. В среде AutoCAD располагать изображения можно как бы на нескольких совмещенных в пространстве носителях (это можно сравнить с наложенными друг на друга прозрачными кальками). Например, чертеж может содержать на одном слое построение призмы со всеми вспомогательными линиями, на другом - сам чертеж в окончательном исполнении, а на третьем - ход лучей в призме. Количество слоёв не ограничивается. Слои можно делать видимыми и невидимыми. С каждым слоем чертежа связывается цвет и тип линий. По мере создания чертежа вы можете вводить новые слои, менять свойства су-

существующих. Подобно системе координат, в любом чертеже AutoCAD всегда существует по крайней мере один слой с именем "0".

Чертёж

Чертёж - это файл, содержащий некую графическую и вспомогательную информацию, полностью описывающую графический объект. AutoCAD предоставляет в распоряжение конструктора все вычислительные ресурсы микроЭВМ и новые средства создания и редактирования чертежа.

Вместе с каждым рисунком AutoCAD содержит так называемые *системные переменные*, в которые заносится определенная информация: о текущих установках рисования (т. е. установках слоя, цвета, типа линий и т. п.), о последнем выполненном действии (т. е. имя последней команды, последняя точка, последний радиус и т. п.), о настройках некоторых команд (т. е. длины фаски, радиус сопряжения и т. п.) и многое другое. Пользователь может вывести на экран перечень и значения системных переменных, и большую часть из них изменить. Остальные изменяются самой системой в процессе работы.

Система AutoCAD позволяет настраивать многие элементы пользовательского интерфейса. Параметры настройки формируются уже на стадии установки AutoCAD на ваш компьютер – большая часть по умолчанию, а что-то (например, размещение папок для программного обеспечения) задает пользователь. Для изменения установок нужно воспользоваться либо командой НАСТРОЙКА (OPTIONS), либо пунктом **Настройка...** (Options...) падающего меню **Сервис** (Tools), либо пунктом **Настройка...** (Options...) контекстного меню, вызываемого по щелчку правой кнопкой мыши в зоне командных строк.

Типы примитивов

Примитивы могут быть простыми и сложными. К простым примитивам относятся следующие объекты: точка, отрезок, круг (окружность), дуга, прямая, луч, эллипс, сплайн, текст.

К сложным примитивам относятся: полилиния, мультилиния, мультитекст, размер, выноска, допуск, штриховка, вхождение блока или внешней ссылки, атрибут и растровое изображение.

Способы ввода координат точек

Можно задавать конечные точки отрезка с помощью мыши. Но этот способ ввода (указания) точек не является единственным. Больше распространен **второй способ** – ввод координат точки с клавиатуры, например: **65,113.24**

В данном примере введена точка с двумя координатами: $X=65$ мм, $Y=113.24$ мм. При вводе координат с клавиатуры запятая является разделителем между абсциссой и ординатой, а точка используется как разделитель между целой и дробной частью числа. Вводимые координаты могут быть целыми или вещественными. При вводе координат следует учитывать, где вы выбрали точку с координатами 0,0. Чаще всего это точка левого угла графического экрана (хотя в процессе работы вы перемещаетесь по рисунку, и точка 0,0 может оказаться в любом месте, даже уйти в невидимую часть чертежа).

Третий способ ввода точек – это относительный ввод в декартовых координатах с клавиатуры, например: @50,25

Данная запись означает, что новая точка задается относительно предыдущей (что определяет символ "@"), со сдвигом по оси X на +50 мм (т. е. вправо на 50 мм) и сдвигом по оси Y на +25 мм (т. е. вверх на 25 мм). Здесь запятая также является разделителем координат. Вводимые числа могут быть целыми и вещественными, положительными, нулевыми и отрицательными.

Четвертый способ ввода точек – это относительный ввод в полярных координатах с клавиатуры, например: @33.5<45

В этой форме записи уже нет запятых, зато появился символ "<", который интерпретируется как знак угла. В данном примере новая точка задается относительно предыдущей, причем расстояние между ними в плоскости равно 33,5 мм (т. е. числу влево от символа угла), а вектор из предыдущей точки в новую образует угол 45 градусов с положительным направлением оси абсцисс (угол измеряется в тех угловых единицах, которые вы задали в настройке). Расстояние должно обязательно быть положительным, а угол может быть любым числом.

Пятый способ ввода точек – это указание с помощью функций объектной привязки. Доступ к функциям объектной привязки осуществляется либо через групповую кнопку панели **Стандартная** (Standard), либо через панель **Объектная привязка** (Object Snap). *Групповой* называется кнопка, у которой в правом нижнем углу имеется черный треугольник. Если выбрать указателем мыши такую кнопку и нажать (не отпуская!) левую кнопку мыши, то раскроется набор кнопок инструментов, которые входят в данную группу. Нужно опуститься по появившимся кнопкам до той, которая вам нужна, и только тогда отпустить нажатую левую кнопку мыши, однако объектной привязкой лучше пользоваться, имея на экране одноименную панель.

Получение справок

В процессе работы очень полезными оказываются команды получения справочной информации о создаваемых объектах. Групповая кнопка справочных операций позволяет получить информацию об:

- **Расстоянии** (Distance)
- **Площади** (Area)
- **Массе** (Mass Properties)
- **Списке** (List)
- **Координатах** (Locate Point).

Команды общего редактирования

Кнопки команд общего редактирования объектов (копирование, перенос, удлинение и т. п.) расположены в панели **Редактирование** (Modify). Каждую из этих команд можно ввести по имени с клавиатуры, а также вызвать с помощью падающего меню **Редакт** (Modify). Многие команды данной группы работают либо с набором предварительно выбранных объектов, либо при отсутствии такого набора выдают запрос **Выберите объекты:** (Select objects:). Остальные команды запрашивают редактируемые объекты в соответствующий момент времени.

Свойства.

У каждого примитива могут быть свои цвет, слой, тип линии, масштаб типа линии, стиль печати, вес линии, гиперссылка и высота – все это в AutoCAD отнесено к *свойствам*. Напомню, что определить текущие значения свойств объекта можно, например, с помощью команды СПИСОК (LIST).

При создании сложных рисунков возникает необходимость присвоения имен отдельным объектам или группам объектов, чтобы ими можно было удобнее оперировать в дальнейшей работе. Особенно это важно при разработке своих собственных приложений, функционирующих в среде AutoCAD. Данной цели служит еще одно свойство примитивов – слой. Более того, слой обладает неоценимой возможностью замораживания (выключения), когда ряд второстепенных в данный момент объектов можно, не удаляя, сделать невидимыми, что позволит успешнее работать с главными объектами.

Вес линии – совершенно новое свойство примитивов, которое отсутствовало в предыдущих версиях системы AutoCAD – это толщина, с которой объект будет выводиться на устройство печати (или графопостроитель). Вы можете нарисовать объекты тонкой линией, но задать ненулевой вес и получить при этом жирные линии на листе бумаги.

Высота – это свойство примитива, применяемое в трехмерных построениях. Оно задает величину выдавливания вдоль оси Z, расположенной перпендикулярно осям X и Y. Например, чтобы круг преобразовать в цилиндр, его нужно выдавить на ненулевую высоту.

В рисунках системы AutoCAD могут присутствовать описания стилей некоторых объектов, что, конечно, облегчает оформление чертежа. К таким стилям относятся: текстовые, размерные и стили мультилиний.

Вывод на плоттер.

Под понятием "плоттер" будем иметь в виду не только графопроектор, но и любое другое устройство вывода, в том числе и принтер. Стоит заметить, что для современных устройств, использующих струйную и лазерную технологию, практически отсутствует грань между принтерами и плоттерами. Поэтому принтерами часто называют плоттеры небольшого формата (не превышающего A2). Поскольку в меню и в документации системы AutoCAD чаще используется термин плоттер, то мы будем применять именно его для обозначения любого устройства вывода.

Любое устройство (локальное или сетевое), к которому вы обращаетесь для вывода чертежа из AutoCAD, должно быть специальным образом конфигурировано (описано) в системе AutoCAD. Операция по установке плоттеров или редактированию их настроек требует специальных знаний. Ее лучше выполнять опытным пользователям или системным программистам, обслуживающим вычислительные комплексы, на которых функционирует AutoCAD. Дополнительную информацию можно найти либо в справочной системе AutoCAD, либо в документации, поставляемой вместе с системой. Автономная настройка (т. е. не зависящая от связи с AutoCAD) самого устройства выполняется с помощью документации, поставляемой вместе с плоттером.

Для того чтобы определить, настроена ли ваша версия AutoCAD, а если настроена, то на плоттеры каких марок, следует воспользоваться командой НАСТРОЙКА (OPTIONS). Эту команду можно вызвать либо с помощью пункта **Настройка** (Options) падающего меню **Сервис** (Tools), либо с помощью контекстного меню, появляющегося при нажатии правой кнопки мыши, если ее указатель расположен в этот момент в зоне командных строк. Команда НАСТРОЙКА (OPTIONS) вызывает диалоговое окно **Настройка** (Options).

Для работы с наиболее распространенными плоттерами и форматами графических файлов в системе AutoCAD присутствуют специальные программы (драйверы), обеспечивающие передачу данных на соот-

ветствующие устройства или в соответствующие форматы. AutoCAD 2000 в стандартной поставке поддерживает большое количество перьевых и струйных типов плоттеров таких фирм, как Hewlett-Packard, Xerox, Ose, CalComp и Houston Instruments, а также наиболее распространенные форматы растровых файлов (JPEG, BMP, PNG, TGA и др.) и форматы PostScript, применяемые в лазерных устройствах печати.

Стили печати

Стили печати – это новое свойство, отсутствовавшее в предыдущих версиях AutoCAD, которое отображает графические объекты при выводе на плоттер специальным образом. Таким образом, примитив в рисунке может на экране выглядеть совсем не так, как он будет нарисован плоттером на бумаге. Изменяться может цвет, тип, а также вес линии. Можно также задать специальное оформление концов и заливки линии. Все такие установки заносятся в таблицы стилей. Система AutoCAD при установке создает ряд стандартных таблиц стилей печати, которые доступны пользователю.

Стили печати могут быть двух видов: именованные и цвето-зависимые. Именованный стиль печати может быть назначен любому объекту, а цвето-зависимый стиль используется в зависимости от цвета примитива.

Цвето-зависимые стили, которых в каждой таблице 255 (по количеству цветов системы AutoCAD), описывают, каким образом нужно выводить на плоттер объекты, имеющие данный цвет. Такие стили удобны для вывода на перьевой плоттер, который имеет ограниченное количество цветов и размеров перьев. По умолчанию, когда имя таблицы действующего цвето-зависимого стиля не задано, действует стиль, который выводит объекты в том виде, в каком они созданы в рисунке.

Блоки и внешние ссылки.

Важным инструментом автоматизации процесса разработки чертежей является использование блоков и внешних ссылок. **Блок**. – это сложный именованный объект, для которого создается описание, состоящее из любого количества примитивов системы AutoCAD текущего рисунка. Блок имеет базовую точку и может применяться для вставки в любое место чертежа, причем в процессе вставки возможен его поворот и масштабирование с различными коэффициентами по разным осям. Примитив, который образуется от операции вставки блока, называется **ВХОЖДЕНИЕ БЛОКА (BLOCK REFERENCE)**. В рисунке может быть любое количество вхождений одного и того же блока.

Внешняя ссылка – это изображение внешнего файла вместе с элементами текущего рисунка, причем файл, на который Вы таким об-

разом ссылается, не переносится в основной рисунок. В результате текущий рисунок может быть насыщен большим количеством внешних изображений новых объектов, но размер текущего файла от этого практически не увеличится. Прimitives, образующийся от операции вставки внешней ссылки, будем называть *вхождением внешней ссылки* или просто *внешней ссылкой*.

Оба упомянутых инструмента являются средством автоматизации труда конструктора и чертежника. С помощью блоков можно строить однотипные объекты, описывая полностью только один их них. Внешние ссылки дают возможность пользоваться ранее созданными файлами стандартных графических элементов.

Первый шаг к использованию блока – создать описание этого блока. Для этого нужно определиться, из каких примитивов будет состоять блок и где у него будет базовая точка. Объекты, которые были включены в блок при его описании, сохраняют свои основные свойства (слои, цвет, тип линии, вес) и во вставленном блоке. Исключением является специальное значение ПОБЛОКУ, которое может быть дано цвету, типу линии и весу. Таким образом, любые части рисунков могут сохраняться в виде отдельных файлов, а любые созданные файлы могут вставляться в текущий рисунок с образованием (или без образования) блоков.

Часто возникает необходимость вместе с блоком держать и надписи, которые могли бы менять свои значения после вставки блока. Например, если Вы рисуете схему с использованием заранее подготовленных блоков условных элементов, тогда номера или наименования вставленных графических элементов Вам нужно будет оформить в виде текстовых надписей. Однако в системе AutoCAD есть специальный примитив, называемый **ОПИСАНИЕ АТТРИБУТА (ATTRIBUTE DEFINITION)**, который может быть включен в описание блока, а при операции вставки этого блока будет запрошено его значение и создан атрибут (текстовая строка), входящий в состав блока.

Атрибуты могут содержать текстовую информацию, которая дополняет графические примитивы рисунка. Извлечение значений атрибутов может быть сделано с помощью специальной команды **АТЭКСП (ATTTEXT)**, которая выводит извлекаемые данные в текстовый файл. Эта операция полезна при создании систем автоматизированного проектирования на базе AutoCAD.

Внешние ссылки

Вставка с помощью команды **ВСТАВИТЬ (INSERT)** одного файла рисунка в другой рисунок, который является текущим, увеличи-

вает его объем, т. к. в него переносятся примитивы вставляемого файла. Но есть еще один способ добавить к текущему рисунку изображение другого рисунка – вставить файл с помощью внешней ссылки. При этом вставляемый файл в текущий рисунок не переносится, а только запоминается его полное имя (обычно вместе с путем). В дальнейшем, когда AutoCAD открывает рисунок, имеющий внешнюю ссылку, то загружается сначала открываемый файл, а затем – содержимое дополнительного файла-ссылки. Таким образом, файл-ссылка не хранится вместе с основным рисунком. Разумеется, при таком варианте основной файл имеет меньший размер по сравнению с вариантом вставки файла с помощью команды **ВСТАВИТЬ (INSERT)**, но попадает в зависимость от дополнительного файла, т. к. тот должен всегда обнаруживаться на своем привычном месте и не менять своего имени. Возможны вложенные ссылки, когда ссылка выполняется на вставляемый файл, который сам содержит внешнюю ссылку на другой файл. Команда **ССЫЛКА (XREF)** управляет в текущем рисунке внешними ссылками на другие файлы.

Вставка объектов, созданных другими системами.

AutoCAD может читать ряд других графических форматов и вставлять объекты, созданные другими известными приложениями (например, Microsoft Office).

Вставка и редактирование растровых изображений

Растровое изображение – это изображение, состоящее из точек (растров), которые благодаря цветам формируют рисунок. AutoCAD может прочитать файл с растровой картинкой и вставить его в текущий рисунок в виде цветного прямоугольника (аналогично внешней ссылке). Редактировать вставленное изображение на точечном уровне AutoCAD не может, но может выполнять подрезку, масштабирование, перенос и другие простые операции редактирования. При наложении одного растрового изображения на другое можно управлять порядком их следования (переносить на передний план или убирать на задний).

Для операций с растровыми изображениями используется команда **ИЗОБ (IMAGE)**, которой соответствует кнопка **Изображение (Image)** панели **Ссылки (Reference)**, а также кнопка **Изображение (Image)** панели **Вставка (Insert)** и пункт **Диспетчер изображений...** (Image Manager...) падающего меню **Вставка (Insert)**.

Трехмерные построения.

AutoCAD может строить примитивы не только в плоскости XY, но в любой плоскости трехмерного пространства. Кроме того, в системе AutoCAD существует большой набор пространственных примити-

вов (поверхностей, тел и др.), которые позволяют выполнять построения трехмерных моделей зданий, сооружений и различных машиностроительных изделий.

Можно не только строить трехмерные объекты, но и рассматривать их в разных видах и проекциях, используя новые системы координат. Имеются такие возможности AutoCAD, как скрытие невидимых линий, тонирование и назначение объектам тех или иных материалов. Все построенные модели можно с помощью пространства листа, оформлять красиво и удобно в виде чертежей.

Работа по модернизации чертежа

Общеизвестно, что в процессе проектирования чертежа конструктор много времени тратит на редактирование. Несмотря на то, что мы с вами, возможно, традиционным способом сможем начертить новый чертеж быстрее, чем с помощью системы AutoCAD, наверняка для исправления его нужно будет потратить немалые усилия, т.е. извести массу стирательной резинки, пожертвовать товарным видом чертежа в пользу его правильности или долго и нудно перечерчивать чертеж. С использованием редакторских возможностей AutoCAD мы можем значительно облегчить свою жизнь.

Функции редактирования позволяют:

- удалять фрагменты изображения;
- восстанавливать случайно удаленные фрагменты;
- перемещать или поворачивать фрагменты или отдельные изображения относительно других;
- копировать созданные фрагменты и располагать их в указанном месте;
- увеличивать или уменьшать объекты;
- создавать зеркально-симметричное изображение;
- изменять свойства (принадлежность к слою, цвет и тип линии) созданных объектов;
- сопрягать линии и строить фаски;
- делить объекты на равные части или размечать на сегменты с заданным интервалом;
- расчленять блоки или полилинии на составные части;
- редактировать полилинии (сглаживать, изменять свойства и т. д.);
- растягивать части рисунка;
- проводить линии, расположенные на заданном (постоянном) расстоянии относительно других.

Средства реализации систем автоматизации конструкторской документации предоставляет компьютерная графика, обеспечивающая

создание, хранение и обработку моделей геометрических объектов и их графических изображений с помощью компьютера. Автоматизация особенно эффективна при разработке устройств на базе параметрически-управляемых унифицированных и типовых элементов конструкций, обеспечивающих многовариантность конструирования. Модель геометрических объектов, содержащая информацию о геометрии объекта, используется как для получения двумерной геометрической модели, так и для расчета различных характеристик объекта и технологических параметров его изготовления. Отсюда следует, что геометрическое моделирование является ядром автоматизированного конструирования и технологической подготовки производства.

Система автоматизации разработки и выполнения конструкторской документации выполняет ввод, хранение, обработку и вывод графической информации в виде конструкторских документов. Для реализации системы необходимы: документы, регламентирующие работу системы, исходная информация для формирования информационной базы, информационная база, содержащая модели геометрических объектов и графических изображений, элементы оформления чертежа по ГОСТу, технические и программные средства создания моделей геометрических объектов и графических изображений и их вывода, интерфейс пользователя в виде графического диалога с компьютером.

Построение таких систем значительно упрощается, если они создаются на базе универсальной, открытой среды проектирования для реализации графических возможностей САПР. Примером данной среды является AutoCad – универсальная графическая система, в основу структуры которой положен принцип открытой архитектуры, позволяющей адаптировать и развивать многие функции AutoCad применительно конкретным задачам и требованиям.

Можно выделить два подхода к конструированию на основе компьютерных технологий. Первый подход базируется на двумерной геометрической модели графического изображения и использовании компьютера как электронного кульмана, позволяющего значительно ускорить процесс конструирования и улучшить качество оформления конструкторской документации. Центральное место при таком подходе занимает чертеж, который служит средством представления изделия и содержит информацию для решения графических задач, а также для изготовления изделия. В таком случае получение графического изображения за компьютером будет рациональным и достаточно эффективным, если созданное графическое изображение используется многократно.

В основе второго подхода лежит пространственная геометрическая модель изделия, которая является более наглядным способом представления оригинала и более мощным и удобным инструментом для решения геометрических задач. Чертеж в этом случае играет вспомогательную роль, а способы его создания основаны на методах компьютерной графики и отображения пространственной модели.

При первом подходе обмен информацией осуществляется на основе конструкторской, нормативно-справочной и технологической документации, при втором подходе – на основе компьютерного представления геометрических объектов, общей базы данных, что способствует эффективному функционированию программного обеспечения САПР конкретного изделия.

Можно выделить два основных вида геометрических объектов:

- постоянный, с постоянными размерами и геометрической формой. Например: графические изображения условных графических обозначений радиоизделий электрических схем или печатных плат.
- параметрически заданный, с переменными размерами и геометрической формой. Например: радиоизделие, зависящее от типоразмеров, типовые и унифицированные несущие конструкции радиоэлектронных устройств, конструктивные элементы типовых деталей.

Постоянные геометрические объекты могут быть сформированы с использованием графического редактора AutoCad. Методы описания параметрически заданных геометрических объектов характеризуются большими затратами на формирование компьютерного представления. Чтобы сократить эти затраты, при описании некоторых групп технических объектов можно пользоваться одним из двух принципиально различных методов: вариантным или генерирующим.

Вариантный метод основан на том, что для определенного класса изделий выявляется модель-образец, с помощью которой можно получить все геометрические формы этого класса изделий. Исполнение изделия определяется заданными параметрами, обнуление которых приводит к исключению составных элементов геометрического объекта. В простейшем случае изменяют только размеры, а конструкция отдельных вариантов класса изделий остается неизменной. Такой вид конструирования называют принципиальным. В этом методе данные технологической документации не подготавливаются каждый раз заново, а закреплены за уже имеющимися соответствующими чертежами. Применение такого метода предполагает, что выбор геометрии для проек-

тируемого изделия уже сделан. Затраты на описание типовой модели превышают затраты на получение вариантов, поэтому во многих системах используется принцип вложенности моделей: один раз описанные типовые модели используются для формирования других типовых моделей в качестве макрокоманд.

В противоположность вариантному методу при генерирующем определяются различные сочетания конструктивных и технологических элементов и выбирается наилучшее решение. Принцип работы системы, использующей генерирующий метод, основан на разделении геометрического объекта на элементы и создании новых геометрических объектов из имеющихся элементов. Различают следующие группы элементов: основные (функциональные), вспомогательные (конструктивные геометрические и элементы формы) и технологические. С помощью основных элементов создается геометрическая форма детали (наружные и внутренние поверхности), проточки (внутренние и наружные). Это дает прежде всего общее описание детали. С помощью вспомогательных элементов, которые непосредственно связаны с основными, осуществляется более подробное описание детали, что позволяет полностью передать ее геометрическую форму. Технологические элементы или характеристики относятся и к основным, и к вспомогательным элементам. Они также влияют на простановку размеров. САПР, работающие по генерирующему принципу, обладают высокой гибкостью и пригодны для решения множества задач. Использование данного метода эффективно, так как большинство конструкторских разработок, называемых новыми конструкциями, создается путем ранее неиспользовавшегося сочетания элементов, давно известных как по принципу формирования, так и по исполнению.

Вопросы для самоконтроля

1. В каких режимах может работать программа **AutoCAD**?
2. Что такое модель и лист?
3. Какие команды называются прозрачными?
4. Какие особые точки имеет окружность?
5. Как можно задать координаты конечных точек отрезка?
6. Сколькими способами можно начертить окружность?
7. Каковы параметры команды вычерчивания дуг?

Тема 3.4. Строение трехмерных объектов, команды для их изменения

Основные модули

- 3D моделирование в программе AutoCAD
- Построение 3D моделей в AutoCAD
- Редактирование объектов
- Команды редактирования

Твердые тела

В программе AutoCAD преимущество отдается именно твердотельному моделированию. Это такие объемные тела, которые обладают свойствами, присущими обычным объектам в нашей жизни: массой, центром тяжести и т.д. (рис.3.4.1.).

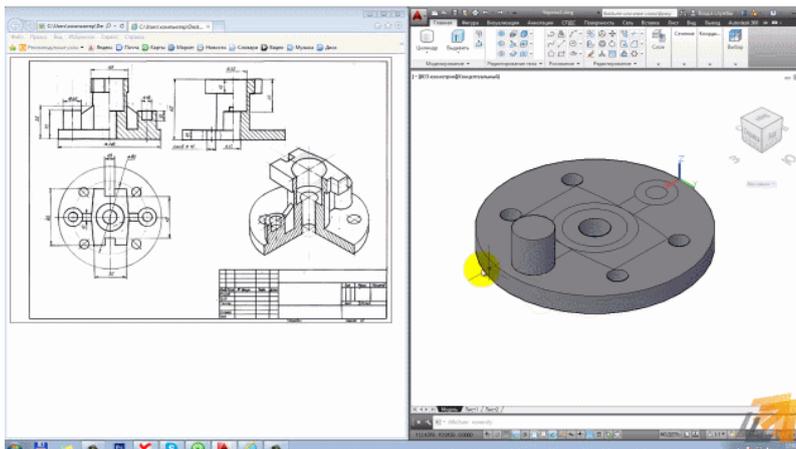


Рис.3.4.1. Твердые тела

Используя логические операции, такие как объединение, вычитание и пересечение, можно создавать твердотельные объекты различной сложности.

Объекты-сети

Данный вид трехмерных моделей использует многоугольное представление. Основными элементами объектов-сетей являются вершины, ребра и грани (рис.3.4.2.).

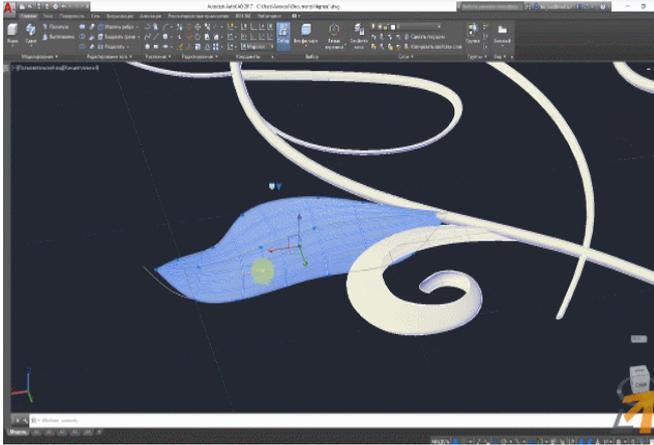


Рис.3.4.2. Объекты-сети

Основные инструменты расположены на соответствующей вкладке "Сети".

Стандартные команды по своему типу идентичны с твердотельными примитивами: параллелепипед, конус, цилиндр и т.д. (рис.3.4.3.).

Эти 3D-модели не имеют свойств массы. Сети позволяют гибко работать с формой объектов в Автокаде. Их можно сглаживать, что является неоспоримым преимуществом (рис.3.4.4.).

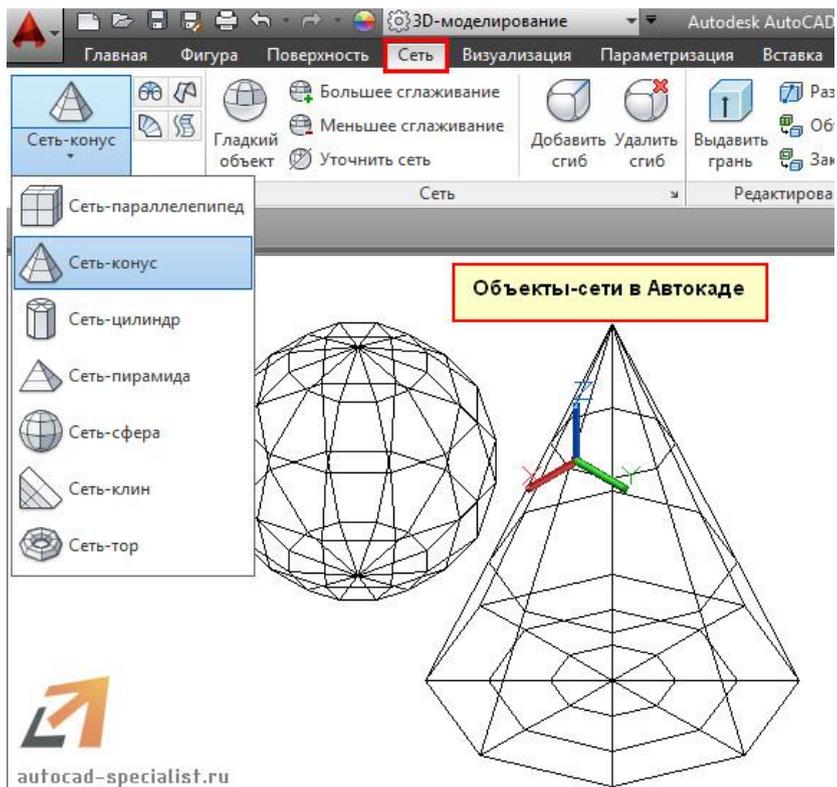


Рис.3.4.3. Твердые тела



Рис.3.4.4. Формы объектов

Поверхности

Поверхность в AutoCAD представляет собой тонкую оболочку, не имеющую объема или массы. В Автокаде существует два вида поверхностей (рис.3.4.5.-3.4.6.):

- NURBS-поверхности;
- процедурные поверхности.

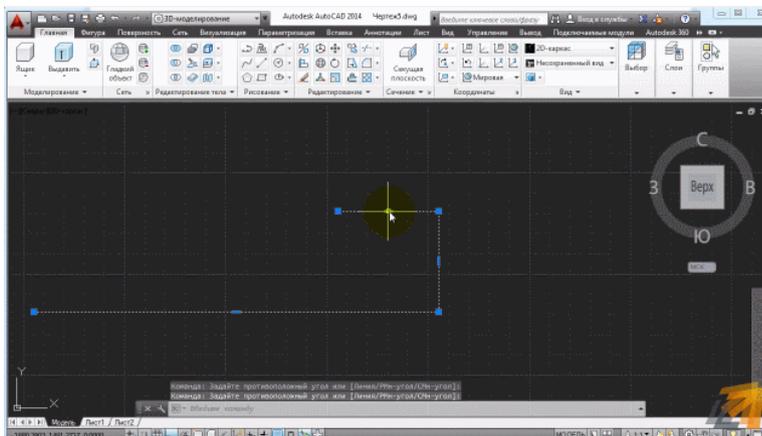


Рис.3.4.5. Поверхность в AutoCAD

Моделирование процедурных поверхностей предоставляет возможность пользоваться преимуществами ассоциативного моделирования, а NURBS-поверхности - преимуществами образования рельефа с помощью управляющих вершин.

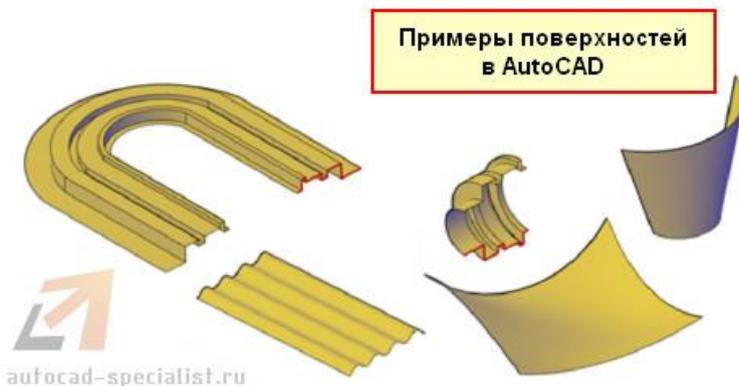


Рис.3.4.6. Поверхность в AutoCAD

Построение 3d моделей в AutoCAD. Общие принципы

Существует два принципиально разных подхода к созданию трехмерных объектов (рис.3.4.7.):

- используя стандартные 3D примитивы (ящик, сфера, конус и т.д.);
- преобразовывая плоский чертеж (2D объекты) в трехмерные, посредством использования соответствующих команд «Выдавить», «Сдвиг» и др.

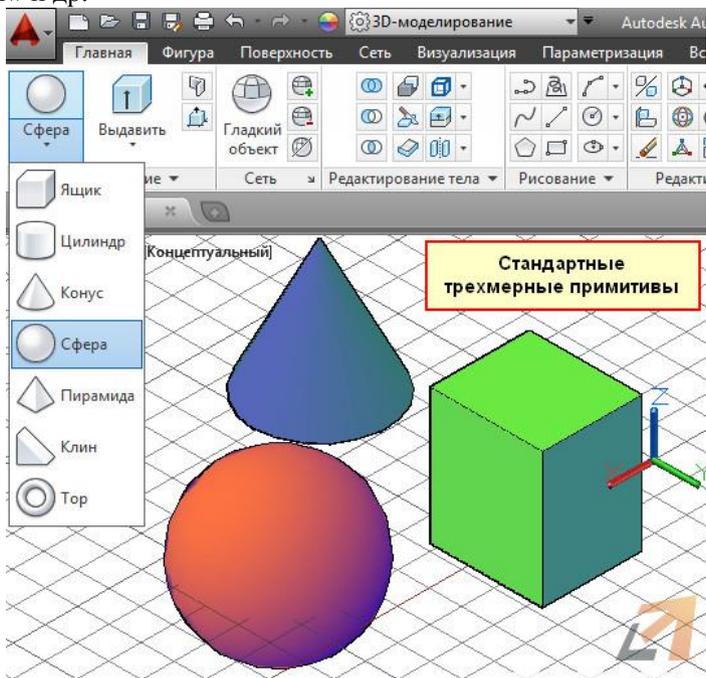


Рис.3.4.7. Построение 3D моделей в AutoCAD

Перевести чертеж из 2D в 3D Автокад можно, если следовать четкому алгоритму (рис.3.4.8.).

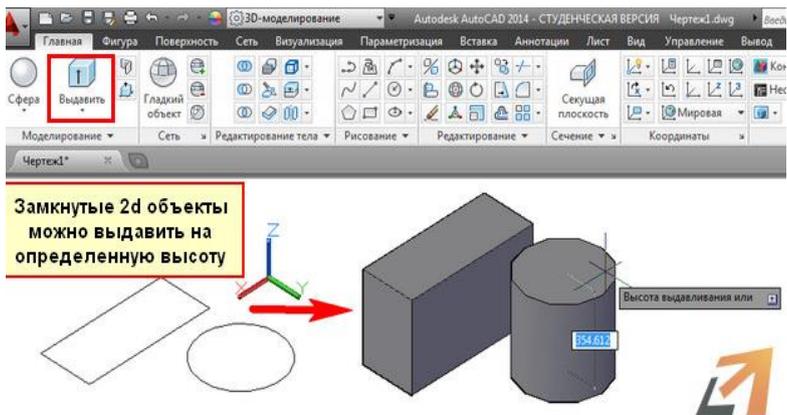


Рис.3.4.8. Использование команды «Выдавить» в AutoCAD

3д чертежи в Автокаде позволяют реализовать проекты различной сложности и в различных сферах, будь то архитектура, ландшафтный дизайн или геодезия.

Программа AutoCAD 3D работает по одним и тем же принципам, в независимости от области применения.

Есть несколько подходов создания 3D модели в Автокад:

1-й подход – создание объектов из примитивов;

2-й подход – создание 3D объектов из плоских примитивов с помощью инструментов «Выдавить», «Лофт», «Вращать» и «Сдвиг» (преобразовывать из 2D в 3D Автокад).

Данные инструменты очень мощные и позволяют создавать универсальные объекты практически любой геометрии. Рассмотрим, как в AutoCAD создавать 3D -модели с помощью перечисленных инструментов.

Создание 3D моделей в AutoCAD. Инструмент «Выдавить»

Принцип работы данного инструмента прост: достаточно начертить плоский примитив и задать ему параметр «Высота».

Рассмотрим на конкретном примере. Используя стандартные 2D примитивы, создадим профиль произвольной формы. В нашем примере возьмем окружность.

Если плоский объект состоит из отдельных отрезков, даже если они представляют собой замкнутый объект, то необходимо выполнить дополнительное их слияние, используя команду «Область», «Контур» или «Соединить». В противном случае, программа AutoCAD 3D создаст объект-поверхность.

Выбираем команду «Выдавить» на вкладке «Главная», панель «Моделирование».

Дальше пошагово выполняем запросы командной строки:

1. Выбрать объект для выдавливания (наш замкнутый 2D профиль).
2. Выбрать один из режимов (по умолчанию необходимо задать высоту).

Результат применения различных режимов команды «Выдавить» продемонстрирован на рис.3.4.9.

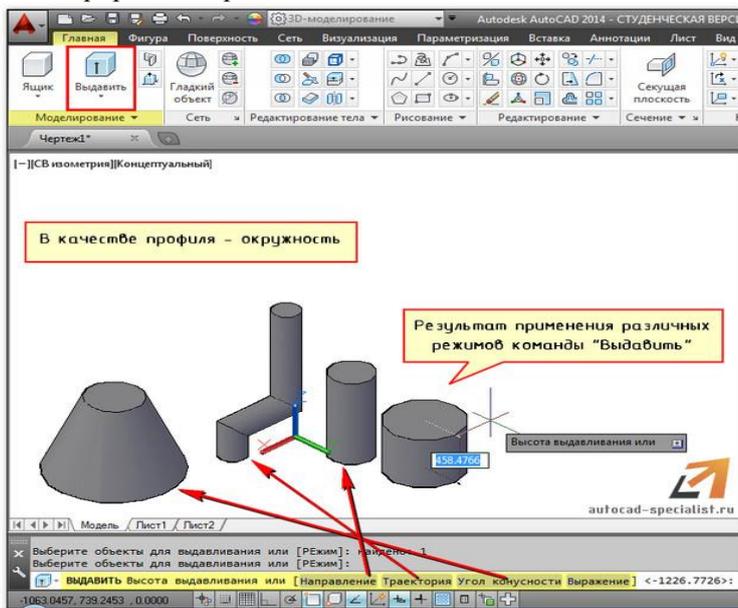


Рис.3.4.9. Использование команды «Выдавить» в AutoCAD

3D моделирование в AutoCAD включает моделирование 3D-тел, поверхностей, сетей и каркасных объектов.

Типы 3D-моделей

В AutoCAD доступно несколько типов 3D-моделирования. В рамках каждой из этих технологий 3D-моделирования доступны различные функциональные возможности (рис.3.4.10.):

- Каркасное моделирование рекомендуется использовать на начальных этапах проекта и в качестве ссылочной геометрии - 3D-каркаса - для последующего моделирования или изменения.

- Моделирование тел-эффективный процесс, с помощью которого легко комбинировать примитивы с выдавленными профилями; он также предоставляет ряд массовых свойств и функций создания сечений.
- Моделирование поверхностей предоставляет функции управления криволинейными поверхностями, обеспечивая точность манипулирования и анализа.
- Моделирование сетей предоставляет функции создания скульптур произвольной формы, создания сгибов и сглаживания.

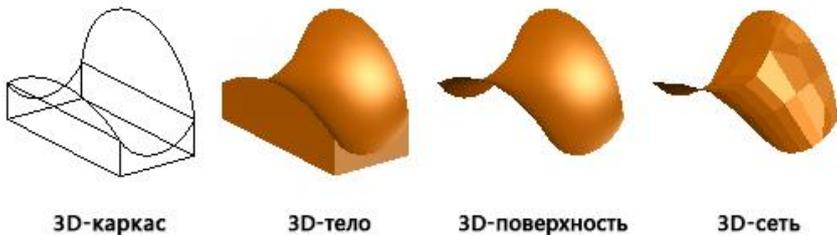


Рис.3.4.10. Типы 3D-моделей в AutoCAD

3D-модели может включать в себя все перечисленные компоненты. Например, можно преобразовать твердотельный 3D-примитив в форме пирамиды в 3D-сеть, чтобы выполнить сглаживание сети. После этого можно преобразовать сеть в 3D-поверхность или обратно в 3D-тело, чтобы использовать возможности соответствующих функций моделирования (рис.3.4.11.).

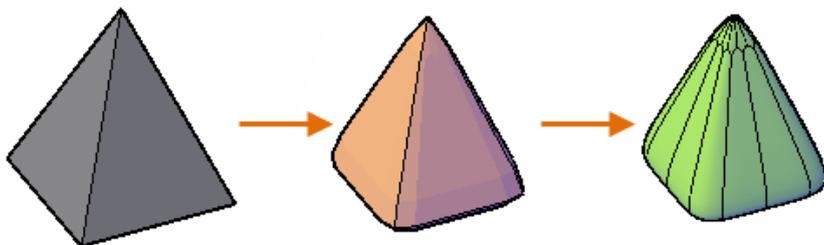


Рис.3.4.11. Преобразование 3D-поверхности в AutoCAD

Просмотр 3D-моделей

Наиболее эффективной командой для динамического просмотра 3D-моделей является команда 3D ОРБИТА (рис.3.4.12.).

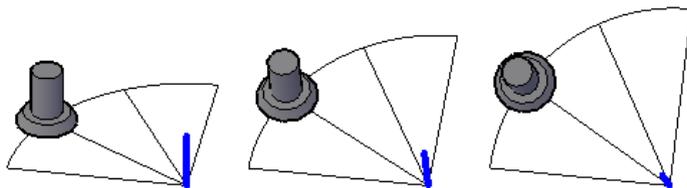


Рис.3.4.12. Просмотр 3D-моделей в AutoCAD

Помимо функций смены видов можно воспользоваться параметрами контекстного меню, появляющегося при щелчке правой кнопкой мыши. К самым популярным параметрам относятся следующие.

- Переход между различными визуальными стилями ("Концептуальный", "Реалистичный" и "Просвечивание")
- Переход между параллельным и перспективным проецированием
- Выбор из стандартных видов ("Вид сверху", "Вид спереди" и др.)

Применение 2D- и 3D-команд AutoCAD

Большинство команд AutoCAD, используемых для 2D-операций, можно также применять и к 3D-моделям. Например, с помощью команды ПОВЕРНУТЬ можно вращать 3D-тело вокруг его оси, которая параллельна оси Z ПСК. Для вращения модели вокруг другой оси может потребоваться изменить направление оси Z ПСК (рис.3.4.13.-3.4.14.).

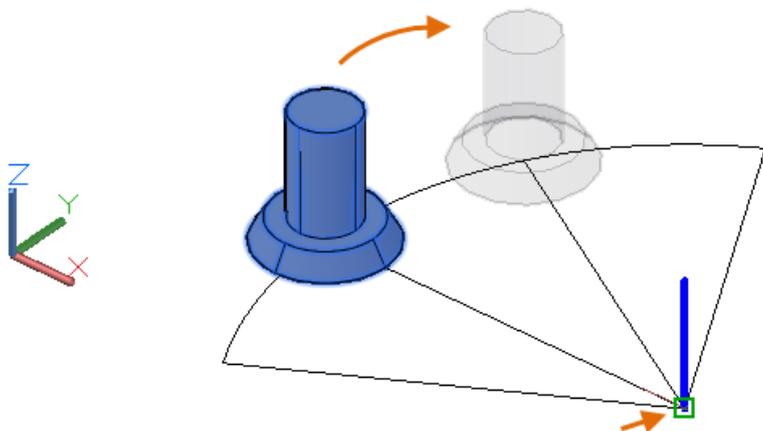


Рис.3.4.13. Просмотр 3D-моделей в AutoCAD

Также доступны команды, специально предназначенные для работы в 3D-средах (например, команда ЗДПОВЕРНУТЬ, которая отображает гизмо для поворота по любой основной оси).

Проще всего начать работу с команды ПОВЕРНУТЬ, поскольку она давно всем знакома. Разумеется, выбор команды зависит от конкретных обстоятельств и предпочтений пользователя.

При создании 3D-моделей можно использовать 3D-поверхности для анализа поверхности и точного управления ей, а 3D-сети – для создания произвольных форм. Эти объекты можно преобразовать в 3D-тела с помощью применения к ним логических операций, формирования разрезов и извлечения физических свойств.

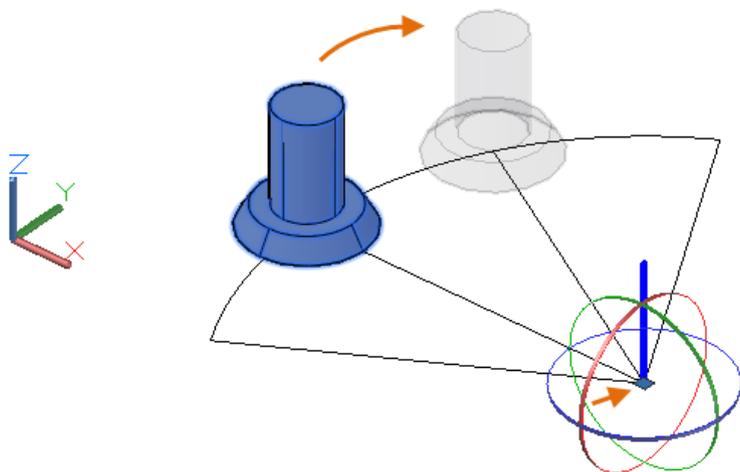


Рис.3.4.14. Просмотр 3D-моделей AutoCAD

Преобразование 3D-поверхностей в 3D-тела

Поверхности можно преобразовать в 3D-тела путем выдавливания поверхности с помощью команды ТОЛЩИНА. В приведенном ниже примере для придания поверхности внутренней толщины вводится отрицательное значение (рис.3.4.15.).

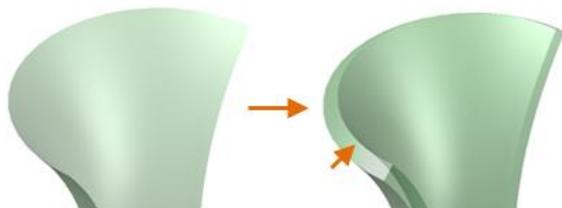


Рис.3.4.15. Преобразование 3D-поверхностей в 3D-тела

Для преобразования группы поверхностей, ограничивающих некоторый объем, в 3D-тело используется команда ПОВЕРХНАПОЛНИТЬ. На рисунке ниже выдавленная дуга и пересекающие ее две плоские поверхности ограничивают непроницаемый объем (рис.3.4.16.).

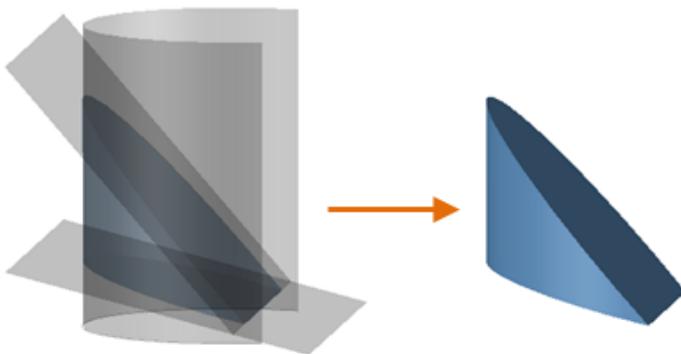


Рис.3.4.16. Преобразование 3D-поверхностей в 3D-тела

Преобразование 3D-сетей в 3D-тела

При преобразовании объектов-сетей в 3D-тела форма нового твердотельного объекта приблизительно соответствует исходному объекту-сети, но не является его точной копией. Точность можно контролировать до определенной степени, указав, будет ли результат сглаженным или многогранным, с помощью системной переменной `SMOOTHMESHCONVERT`. Кроме того, чтобы оптимизировать компланарные грани, их можно объединить.

Например, для 3D-сети на рисунке сначала была увеличена степень сглаживания, а потом эта сеть была преобразована в сглаженное 3D-тело (рис.3.4.17.).

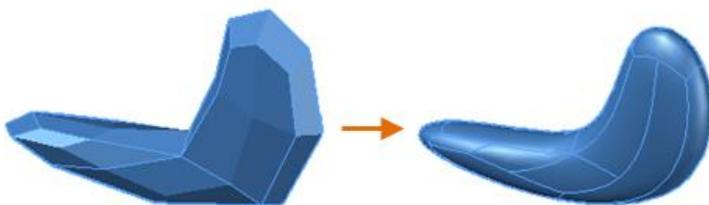


Рис.3.4.17. Преобразование 3D-сетей в 3D-тела

Редактирование объектов в AutoCAD

1. В командной строке введите **-render**.
2. В строке запроса "Задайте стандартные параметры визуализации" введите имя набора параметров визуализации, который требуется использовать.
3. В строке запроса "Задайте направление визуализации" введите **render**, чтобы выполнить визуализацию в окне визуализации.
4. В строке запроса "Ширина вывода" введите значение ширины визуализируемого изображения в пикселах.
5. В строке запроса "Высота вывода" введите значение высоты визуализируемого изображения в пикселах.
6. В строке запроса "Сохранить результаты визуализации в файл?" введите **yes**.
7. В строке запроса "Задайте имя и путь выходного файла" укажите имя файла изображения и путь к нему.

Редактирование объектов предусматривает возможность удаления, копирования перемещения объектов, изменения их свойств таких, как цвет, вес линий, тип линий и ряда других свойств. Над объектами можно выполнять операции удлинения, обрезания, зеркального отражения, поворота на заданный угол и т.д.

Доступ к командам редактирования осуществляется через (рис.3.4.18.):

1. Пункт меню **Modify**.
2. Панели инструментов **Modify**.
3. Введение команды с клавиатуры.

Команды редактирования

Команда Свойства (Properties)

Способы ввода команды:

- Набрать с клавиатуры команду: **Properties**.
- Вызов меню: **Modify - Properties**.
- Дважды щелкнуть левой кнопкой мыши на объекте.
- Из контекстного меню объекта выбрать команду **Properties**.

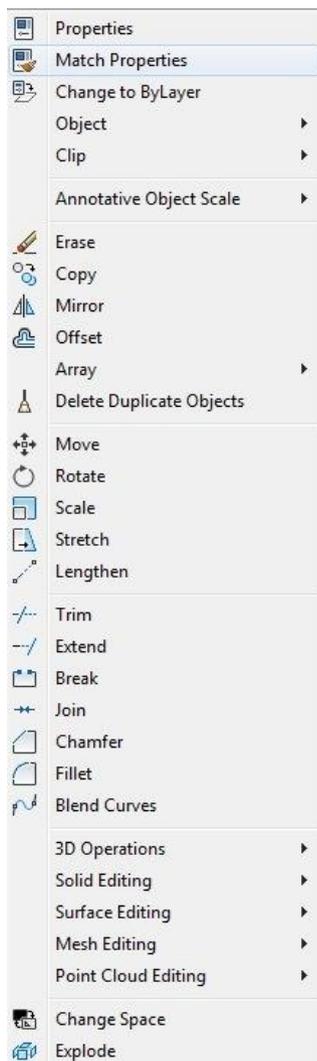


Рис.3.4.18. Панель инструментов для редактирования объектов

Данной командой можно изменить свойства одного объекта или общие свойства нескольких выбранных объектов. Команда открывает диалоговое окно **Properties**, в верхней строке которого содержится список избранных объектов и кнопки (рис.3.4.19.):

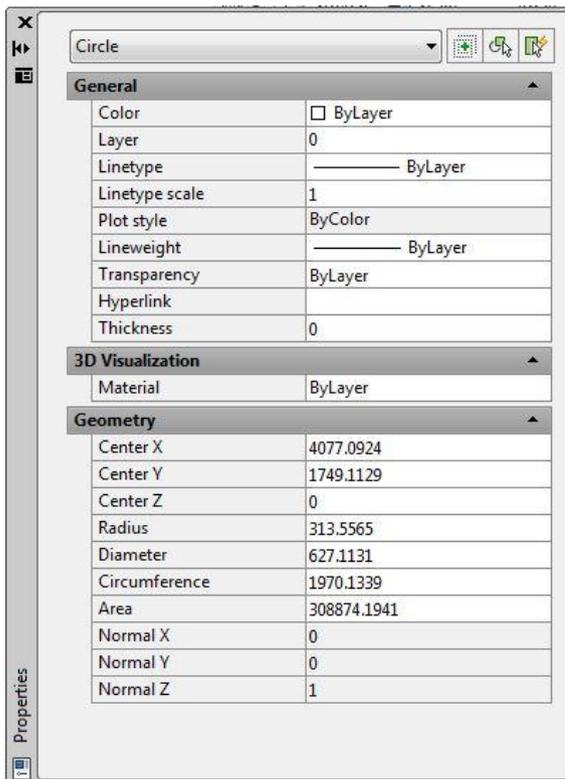


Рис.3.4.19. Редактирование объектов

Окно разделено на две колонки. Первый столбик содержит название свойства, а второй – значение свойства, которое можно изменить.

Команда ERASE (Стереть)

Способы ввода команды:

- Набрать с клавиатуры команду **ERASE**.
- Вызов меню: **Modify-ERASE**.

- Кнопка на панели **Modify**.



- Из контекстного меню объекта выбрать команду **Erase**.

Допускается выбор объектов до ввода команды или после ввода команды. В первом случае после ввода команды выбранные объекты удаляются. Во втором случае система предлагает выбрать объекты.

В командной строке выводится диалог:

Command: _erase

Select objects: 1 found

Select objects: 1 found, 2 total

Select objects: 1 found, 3 total

Select objects:

После выбора объектов нажмите клавишу **Enter**. Объекты будут удалены. Если необходимо удалить объект не полностью, а лишь определенную его часть, то используют команды **TRIM (Обрезать)** и **BREAK (Разорвать)** (рис.3.4.20).

Команда BREAK (Разорвать)

Способы ввода команды:

- Набрать с клавиатуры команду **BREAK**.
- Вызов меню: **Modify -Break**.
- Кнопка на панели **Modify**. 

Командой удаляется часть объекта между двумя указанными точками. Использование команды позволяет разорвать графический примитив, такой как линия, дуга на два объекта. При удалении части отрезка или дуги первой указывается точка на объекте, а второй - точка за пределами конца, который необходимо удалить. Для круга часть, которая удаляется, задается двумя точками в направлении против часовой стрелки.

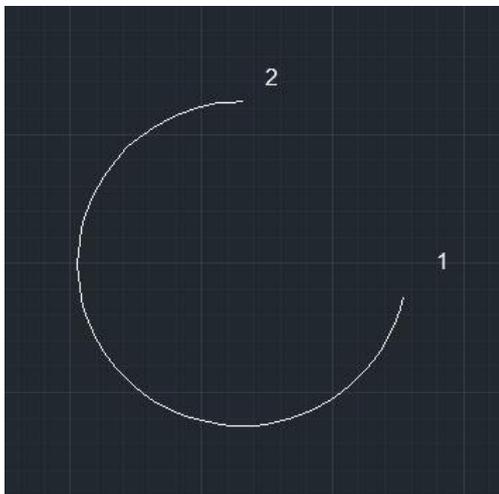


Рис.3.4.20. Команда BREAK (Разорвать)

После ввода команды система выдает запросы (таблица 3.4.1.):

Command: _break	Команда разорвать.
Select object:	Выбрать объект или указать первую точку.
Specify second break point or [First point]	Указать вторую точку.

Выбор объектов осуществляется после ввода команды.

Команда TRIM (Обрезать)

Способы ввода команды:

- Набрать с клавиатуры команду **TRIM**.
- Вызов меню: **Modify - Trim**.
- Кнопка на панели **Modify**. 

Команда обрезает выбранный объект режущей кромкой, которая пересекает его, и может быть дугой, линией, полилинией и т.д.. После ввода команды сначала указывается одна или последовательно выбирается, несколько режущих кромок. Процесс выбора завершается нажатием клавиши **Enter**. Далее система выдаст запрос на выбор объектов, которые обрезаются. Необходимо указывать ту часть объекта, которая обрезается. Для завершения выбора нажать клавишу **Enter** (рис.3.4.21.).

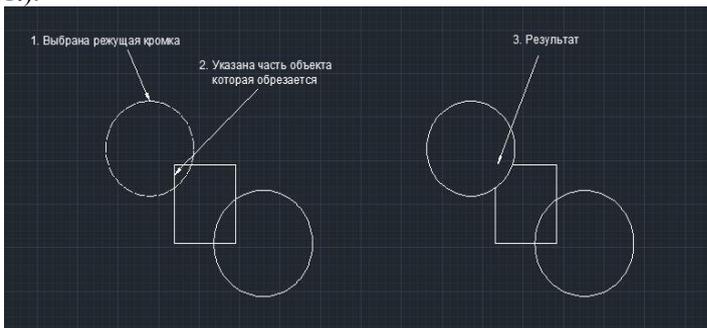


Рис.3.4.21. Команда TRIM (Обрезать)

Работа с командой сопровождается запросам (таблица 3.4.2):

Таблица 3.4.2

Command: Trim	Команда Обрезать
Current settings: Projection = UCS,Edge = NoneSelect cut- ting edges ...	Выбрать режущую кромку
Select objects: 1 found Select ob- jects:	Первая режущая кромка определена. Выбор режущих кромок можно продолжить или нажать клавишу ENTER, чтобы прекратить выбор кромок.
Select object to trim or shift- select toextend or [Project / Edge / Undo]:	Выбрать объект, который обрезается. Последовательно можно выбрать несколько объектов прекратить выбор объектов нажатием клавиши ENTER.

Команда EXTEND (Удлинить)

Способы ввода команды:

- Набрать с клавиатуры команду **EXTEND**.
- Кнопка на панели **Modify**. 

Командой удлиняются графические примитивы (линия, дуга, полилиния, луч, сплайн) до точек пересечения с другими объектами, которые будут предельными кромками. По своему действию она противоположна команде **TRIM**. после введения команды выдаются запросы на выбор предельных кромок. Прекратить выбор можно нажатием клавиши **Enter**. Далее система выдает запрос на выбор объектов, которые необходимо продлить. Завершение выбора осуществляется нажатием

клавиши **Enter**. Указывая объекты, нужно показывать край, который ближе к предельной кромке, иначе команда не будет выполняться (рис.3.4.22.).

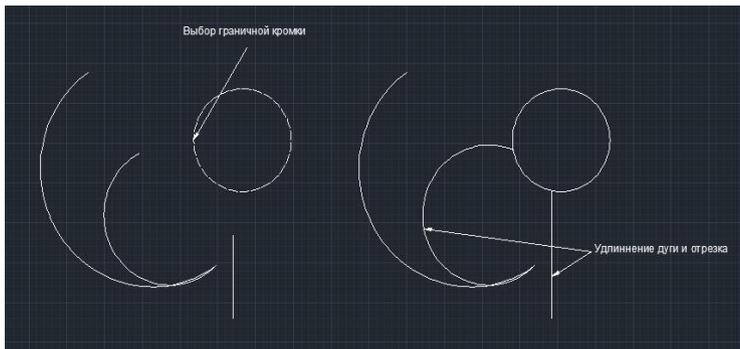


Рис.3.4.22. Команда EXTEND (удлинить)

Подсказки системы будут иметь следующий вид (таблица 3.4.3.):

Таблица 3.4.3.

Command: _extend	Команда Удлинить.
Current settings: Project i on = UCS, Edge = NoneSelect boundary edges ...	
Select objects: 1 found	Показать пре- дельную кромку
Select objects:	Нажать ENTER
Select object to extend or shift-select to trim or[Project / Edge / Undo]:	Выбрать объ- екты, которые удлиняются

Select object to extend or shift-select to trim or[Project / Edge / Undo]:

Нажать
ENTER

Команда MOVE (Перемещение)

Способы ввода команды:

- Набрать с клавиатуры команду **MOVE**.
- Вызов меню: **MODIFY - MOVE**.

- Кнопка на панели **Modify**. 

Для перемещения нужно на выбранном объекте указать базовую точку и задать координаты нового положения базовой точки после перемещения объекта. Координаты задаются любым из способов.

При использовании команды поддерживайте диалог (таблица 3.4.4.):

Таблица 3.4.4.

Command: move	Команда Переместить.
Select objects: 1 found	Выберите объект
Select objects:	Нажмите Enter .
Specify base point or displacement:	Указать базовую точку.
Specify second point of displacement or <use firstpoint as displacement>:	Задать координаты для нового положение базовой точки.

Команда COPY (Копировать)

Способы ввода команды:

- Набрать с клавиатуры команду **COPY**.
- Вызов меню: **MODIFY - COPY**.

- Кнопка на панели **Modify**. 

Копирование объектов осуществляется аналогично их перемещению. Разница заключается в том, что образуется копия объекта в заданной позиции с сохранением оригинала.

Команда ROTATE (Повернуть)

Способы ввода команды:

- Набрать с клавиатуры команду **ROTATE**.
- Вызов меню: **MODIFY - ROTATE**.
- Кнопка на панели **Modify**. 

Команда совершает поворот объекта на заданный угол относительно базовой точки (центра поворота). Положительным считается значение угла при повороте в направлении против часовой стрелки. Базовую точку можно выбрать при включенном режиме объектной привязки.

Диалог с системой (таблица 3.4.5.):

Таблица 3.4.5.

Command: rotate	Команда Повернуть.
Current positive angle in UCS:ANGDIR = counter-clockwiseANGBASE = 0	Система выводит текущие параметры значения угла: ANGDIR = против часовой стрелки.
Select objects: 1 found	Выбрать объект.
Select objects:	Нажмите Enter .
Specify base point:	Указать базовую точку – центр поворота.
Specify rotation angle or [Reference]: 30	Задать значение угла в градусах.

Параметр **Reference** - (опорный угол) позволяет задать вектор и угол поворота данного вектора. Положение вектора после поворота задает новое положение объекта (рис.3.4.23.).

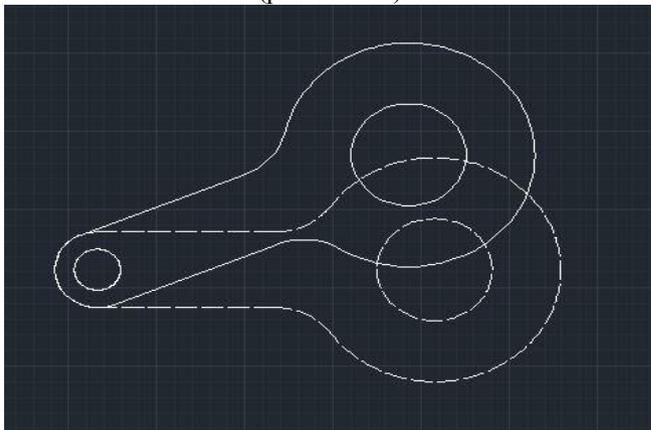


Рис.3.4.23. Команда Reference (опорный угол)

При выборе параметра поддерживайте диалог (таблица 3.4.6.):

Таблица 3.4.6.

Command: rotate	Команда Повернуть.
Current positive angle in UCS:ANGDIR = counter-clockwiseANGBASE = 0	Система выводит текущие параметры значения угла: ANGDIR = против часовой стрелки.
Select objects: 1 found	Выбрать объект.
Select objects:	Нажмите Enter .
Specify base point:	Указать базовую точку – центр поворота.

Specify rotation angle or [Reference] R	Выбор параметра Reference
Specify the reference angle <0>:	Указать первую точку вектора (точка 1).
Specify second point:	Указать вторую точку вектора (точка 2).
Specify the new angle: 50	Ввести значение угла.

Команда MIRROR (Зеркало)

Способы ввода команды:

- Набрать с клавиатуры команду **MIRROR**.
- Вызов меню: **MODIFY - MIRROR**.
- Кнопка на панели **Modify**. 

Командой получают зеркальное отражение объекта с сохранением или удалением исходного объекта. После выбора объектов нужно задать ось, относительно которой образуется зеркальное отражение (рис.3.4.24.).

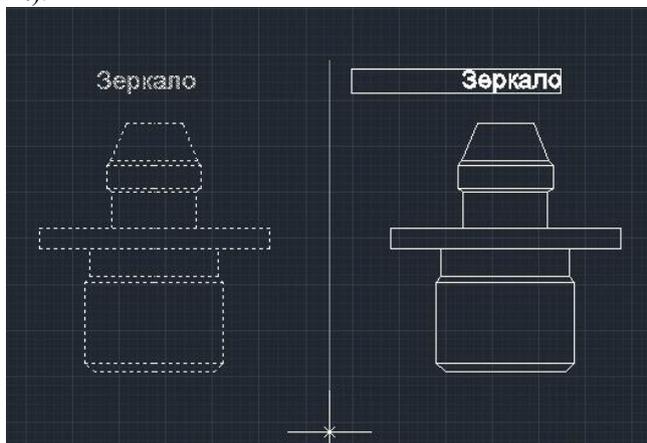


Рис.3.4.24. Команда MIRROR (Зеркало)

Диалог с системой (таблица 3.4.7.):

Таблица 3.4.7.

Command: _mirror	Команда Зеркало.
Select objects: 1 found	Выбрать объект.
Select objects:	Нажмите Enter.
Specify first point of mirror line:	Указать первую точку оси.
Specify second point of mirror line:	Указать вторую точку оси
Delete source objects? [Yes / No] <N>: n	Запрос на удаление исходного объекта

Команда SCALE (Масштаб)

Способы ввода команды:

- Набрать с клавиатуры команду **SCALE** .
- Вызов меню: **MODIFY - SCALE**.
- Кнопка на панели **Modify**. 

Пропорциональное увеличение или уменьшение размеров объекта или группы объектов относительно заданной базовой точки. Для увеличения объектов задают коэффициент больше единицы. Для уменьшения объектов коэффициент имеет значение меньше единицы (рис.3.4.25.).

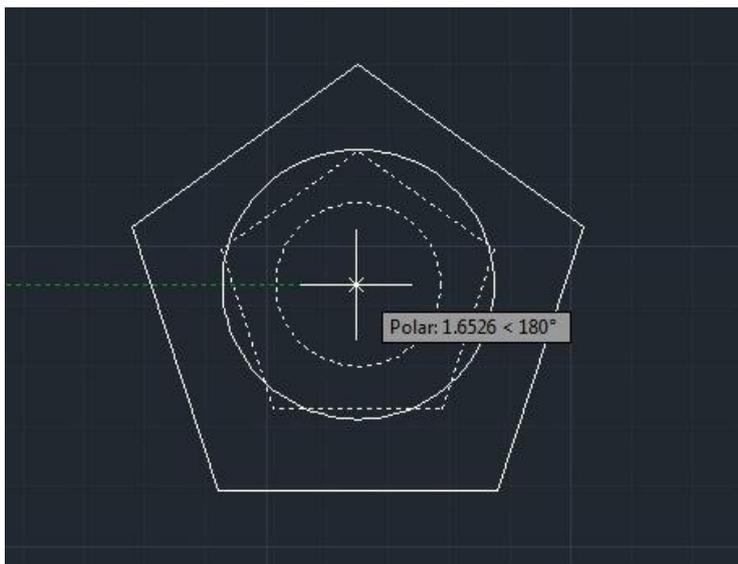


Рис.3.4.25. Команда SCALE (Масштаб)

Подсказки, которые выдаются системой (таблица 3.4.8.):

Таблица 3.4.8.

Command _scale	Команда Масштаб
Select objects: 1 found	Выбрать объект или группу объектов
Select objects :	Нажать Enter
Specify base point:	Указать базовую точку
Specify scale factor or [Reference] : 2	Задать коэффициент масштабирования.

Задать масштабный коэффициент можно интерактивным способом. После выбора базовой точки от нее до перекрестка тянется резиновая нить. Увеличивая или уменьшая ее размер, можно задать значение коэффициента.

Опция **Reference** (Опорный отрезок) позволяет задать коэффициент масштабирования, введя первоначальную длину отрезка и его новую длину. Коэффициент рассчитывается как отношение второго числа к первому. Длина отрезка при этом может быть определена автоматически как расстояние между двумя указанными с помощью курсора точками .

Используем команду для увеличения объекта в 8 / 5 раз.

Диалог с системой будет иметь вид (таблица 3.4.9.):

Таблица 3.4.9.

Command : <code>_scale</code>	Команда Масштаб
Select objects: 1 found	Выбрать объект 1
Select objects: 1 found, total 2	Выбрать объект 2
Select objects :	Нажать Enter
Specify scale factor or [Reference] : r	Выбрали опцию Reference .
Specify reference length < 1 > : 5	Длинна отрезка.

Команда STRETCH (Растянуть)

Способы ввода команды:

- Набрать с клавиатуры команду **STRETCH**.
- Вызов меню: **MODIFY - STRETCH**.
- Кнопка на панели **Modify**. 

Команда используется для растяжения или наоборот сжатия графических примитивов путем изменения положения их предельных то-

чек. При применении команды к дуге изменяется ее радиус. Выбор объектов осуществляется только секущей рамкой или секущими многоугольником. Для работы с кругом, текстом, блоком команда не предназначена. Применение команды к набору объектов дает возможность изменить их без нарушения взаимосвязи между объектами (рис.3.4.26).

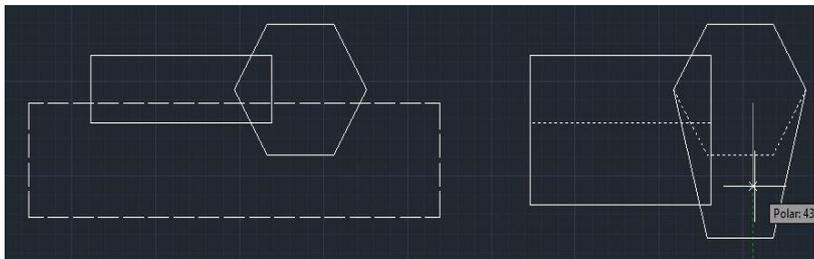


Рис.3.4.26. Команда STRETCH (Растянуть)

Работа с командой сопровождается диалогом (таблица 3.4.10):

Таблица 3.4.10.

Command: _stretch	Команда Растянуть.
Select objects to stretch by crossing-window or crossing-polygon...	Выбрать объекты секущей рамкой или секущим многоугольником.
Select objects: Specify opposite corner: 2found	

Select objects: Нажать **Enter**. **Specify base point or displacement:** Выбрать базовую точку. **Specify second point of displacement or <use first point as displacement>:** Выбрать базовую точку.

Командой изменяется положение только тех точек, попавших в секущую рамку. Положение остальных точек остается неизменным.

Команда LENGTHEN (Продлить)

Способы ввода команды:

- Набрать с клавиатуры команду **LENGTHEN**.

- Вызов меню: **MODIFY- LENGTHEN**.

- Кнопка на панели **Modify**.



Команда служит для изменения длины отрезка или дуги. Чтобы изменить длину отрезка, нужно ввести значение прироста, положительное значение в случае увеличения длины и отрицательное значение при уменьшении длины. Чтобы изменить размер дуги, указывается значение прироста внутреннего угла, положительное, в случае увеличения размера и отрицательное при уменьшении размера.

Опции команды:

DElta – значение величины прироста длины.

Percent – значение увеличения или уменьшения длины отрезка или дуги задается в процентном отношении текущего значения к новому. Если задается величина, превышающая 100%, происходит увеличение длины. Если заданная величина менее 100%, происходит уменьшение длины.

Total – позволяет указать новую полную длину объекта. Для дуги можно указать новое значение внутреннего угла.

DYnamic – длина изменяется динамически с помощью курсора.

Вопросы для самоконтроля

1. Как называется трехмерное рабочее пространство?
2. Как в основном нужно задавать координаты в 3D-пространстве?
3. Как создать поверхность?
4. Что означает "экструзия" и "выдавливание"?
5. Как вырезать внутреннюю полость в теле?
6. Как разрезать тело на части?
7. Каким путем получают отверстия в объемных фигурах?

Тема 3.5. Автоматизация строительных конструкций (Lira)

Основные модули

- Программный комплекс ЛИРА-САПР
- Предназначение программы и его применение
- Автоматизация процессов проектирования
- Вкладки и панели ленты

Программный комплекс ЛИРА-САПР (ПК ЛИРА-САПР) – это многофункциональный программный комплекс для расчета, исследования и проектирования конструкций различного назначения. ПК ЛИРА-САПР с успехом применяется в расчетах объектов строительства, машиностроения, мостостроения, атомной энергетики, нефтедобывающей промышленности и во многих других сферах, где актуальны методы, обеспечивающие прочность и безопасность строительных объектов. Программные комплексы семейства ЛИРА имеют более чем 50-летнюю историю создания, развития и применения в научных исследованиях и практике проектирования конструкций. Программные комплексы семейства ЛИРА непрерывно совершенствуются и адаптируются к новым операционным системам и графическим средам.

Кроме общего расчета модели объекта на все возможные виды статических нагрузок (силовых, температурных, деформационных) и динамических воздействий (ветер с учетом пульсации, сейсмические воздействия по различным нормам, гармонические колебания и т.п.) ПК ЛИРА-САПР автоматизирует ряд процессов проектирования: определение расчетных сочетаний нагрузок и усилий, назначение конструктивных элементов, подбор и проверка сечений стальных и железобетонных конструкций с формированием эскизов рабочих чертежей колонн и балок. ПК ЛИРА-САПР позволяет исследовать общую устойчивость рассчитываемой модели, проверить прочность сечений элементов по различным теориям разрушения. ПК ЛИРА-САПР предоставляет возможность производить расчеты объектов с учетом физической, геометрической, физикогеометрической и конструктивной нелинейностей, моделировать процесс возведения сооружения с учетом монтажа-демонтажа элементов с отслеживанием изменений физических свойств материалов. ПК ЛИРА-САПР состоит из нескольких взаимосвязанных информационных систем. Организация взаимосвязей между этими системами обеспечивает технологичность работы с комплексом так, что комплекс как бы сам ведет пользователя - от создания расчетной моде-

ли к конструированию элементов. Ниже представлена общая схема функционирования ПК ЛИРА–САПР.

ПК ЛИРА-САПР является компонентом BIM-технологии и поддерживает открытый подход к информационному моделированию зданий. Инструментарий ПК ЛИРА–САПР позволяет организовать эффективную связь на базе информационных моделей, согласованных процессов и методов, а также общей терминологии между программными комплексами различного назначения, позволяет импортировать архитектурные модели из различных форматов и экспортировать результаты расчета в различные графические программные комплексы. Используемые форматы данных: *.msh, *.stl, *.dxf, *.obj, *.IFC, *.3ds, *.mdb, *.stp, *.sli, *.docx, *.xlsx, *.bmp, *.gif, *.txt, *.liraKM и др.

Описание ленточного интерфейса ПК ЛИРА–САПР®2018* Ленточный вид интерфейса устанавливается по умолчанию при первоначальной загрузке ВИЗОРСАПР. Ленточный вид интерфейса представляет собой рабочее пространство, основанное на панелях инструментов, разделенных вкладками, на которых отображаются инструменты и элементы управления, предназначенные для решения определенной задачи. Вкладки ленты соответствуют этапам работы со схемой: создание схемы, анализ напряженно-деформированного состояния, конструирование. Варианты лент: Лента Плюс; Лента ЖБК; Лента Сталь; Лента Пользователя. Лента Пользователя - индивидуально настроенный ленточный интерфейс с пользовательскими вкладками, панелями и кнопками. Выключить ленту можно в дополнительном меню ленты Стиль, сняв флажок с пункта Лента . Включить ленту можно в меню ОКНО, выбрав соответствующий пункт во вложенном меню Стиль интерфейса.

Ниже приводится описание стиля ленточного интерфейса Лента Плюс. Лента разделена такими вкладками:

- Создание и редактирование – создание геометрии схемы, базовое редактирование, назначение жесткостей, задание параметров для конструирования, задание нагрузок.

- Расширенное редактирование – расширенное редактирование схемы, сборка схем, работа с блоками и суперэлементами, вызов и управление параметрами работы системы ГРУНТ, работа с линиями влияния, отображение мозаик для анализа геометрии и заданных свойств элементам расчетной схемы.

- Расчет – задание данных для статического, динамического и дополнительных расчетов, формирование таблиц для расчета, выполнение расчета.

- Анализ – наиболее популярные функции анализа: анализ перемещений, усилий, напряжений в стержневых, пластинчатых и объемных КЭ, формирование таблиц результатов.

- Расширенный анализ – анализ усилий в специальных элементах, расчет и анализ результатов дополнительных расчетов.

- Железобетон – задание информации для конструирования, анализ подбора и проверки армирования элементов.

- Сталь – задание информации для конструирования, анализ результатов проверки и подбора стальных сечений и узлов.

- Кладка – задание информации данных для конструирования, анализ результатов проверки прочности кладки и подбора армирования.

Первые три вкладки Создание и редактирование, Расширенное редактирование, Расчет соответствуют режиму создания Расчетной схемы. Вкладки Анализ, Расширенный анализ – режиму просмотра Результаты расчета, вкладки Железобетон, Сталь, Кладка – режиму Конструирование (при отсутствии результатов расчета для открытой в ВИЗОР-САПР задачи режим создания расчетной схемы предполагается независимо от выбранной в ленте вкладки).

При выборе на расчетной схеме объектов определенного типа дополнительно выводятся контекстные вкладки ленты. Каждая из контекстных вкладок содержит операции, которые относятся к выделенным объектам. Контекстная вкладка закрывается при снятии выделения с объектов. Контекстные вкладки, предназначенные для работы с узлами и элементами схемы, содержат команды по созданию и редактированию схемы и не могут быть вызваны в режимах просмотра Результаты расчета и Конструирование.

При выделении объектов в режиме создания Расчетной схемы активируются следующие контекстные вкладки:

- Работа с узлами;
- Работа со стержнями;
- Работа с пластинами;
- Работа с объемными КЭ;
- Работа с одноузловыми КЭ;
- Работа с СЭ.

Кроме вкладок и контекстных вкладок существуют также 3 вкладки, которые вызывают отдельные режимы работы программы:

- 3D вид – пространственное отображение созданного объекта, в том числе отображение сечения элементов и предоставление аппарата для детального исследования с различных точек зрения в отдельном окне созданного объекта.

- Триангуляция – создание и триангуляция плоского фрагмента, содержащего отверстия.

- Редактор контуров – создание и триангуляция плоского фрагмента, содержащего отверстия и внутренние стержни.

В правом верхнем углу содержится дополнительное меню ленты. Оно включает в себя меню Стиль для смены стилей ленточного интерфейса и видов их отображения, меню Окно для работы с окнами задач и меню Справка для работы со справочной информацией.

Кроме вышеупомянутых вкладок при выбранном стиле Лента активированы также 3 панели инструментов: Выбор, Проекция и Проекция 2.

При первоначальной загрузке ВИЗОР-САПР, если ранее не было создано ни одного файла с исходными данными в рабочих каталогах системы, или при закрытии всех окон с ранее загруженными задачами, все команды на вкладках ленты, кроме четырех команд на вкладке Расчет, становятся неактивными.

Команды ленты, которые доступны в режиме начальной загрузки ВИЗОР-САПР (без открытых документов):

- Расчетный процессор – вызов диалогового окна для управления параметрами расчета выбранного файла задачи и собственно расчет.

- Протокол решения – выбор и просмотр текстового файла формата имя Задачи_01.шифрЗадачи, содержащего протокол расчета задачи.

- МЕТЕОР (Вариации моделей) – данный процессор предоставляет возможность производить объединение результатов, полученных после расчета топологически одинаковых расчетных схем (одинаковая геометрия, одинаковое количество и типы элементов), отличающихся граничными условиями, жесткостными характеристиками элементов, параметрами грунтового основания и т.п.

- Расчет пакета задач – вызов диалогового окна для расчета пакета (списка) задач. При расчете пакета задач предполагается, что будут выполнены не только расчет статики и динамики каждой из задач, но и все этапы расчета в соответствии с заданными исходными данными (расчет РСУ, РСН, унификации РСУ, расчет нагрузки на фрагмент, расчет устойчивости, расчет главных и эквивалентных напряжений), а также расчеты конструирующих систем.

Элементы ленточного интерфейса Вкладка (рис.3.5.1) - элемент графического интерфейса пользователя, который позволяет в одном окне приложения переключаться между предопределёнными наборами элементов интерфейса, когда их доступно несколько, а на выделенном для них пространстве окна можно показывать только один из них.

Вкладка представляет собой «выступ» с надписью, расположенный на границе выделенной под сменное содержимое области экрана. Клик мышью по вкладке делает её активной, и на управляемой вкладками области экрана отображается соответствующее ей содержимое. Вкладки располагаются друг за другом горизонтально.

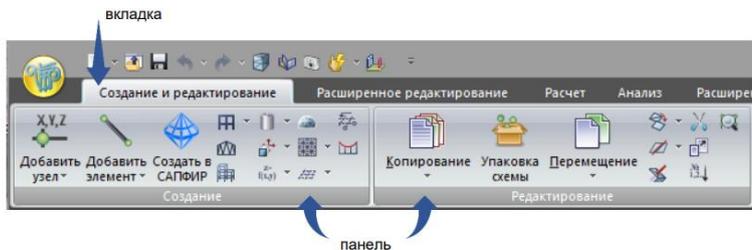


Рис.3.5.1. Вкладки и панели ленты

Контекстные вкладки ленты (рис.3.5.2) – при выборе на расчетной схеме объектов определенного типа вместо инструментальной панели выводится особая контекстная вкладка ленты. По умолчанию контекстная вкладка закрывается при снятии с объектов расчетной схемы выделения. Включает постоянное отображение в ленте контекстных вкладок для тех элементов, которые есть в схеме, можно снять флажок Контекстные закладки по выделенным (кнопка Параметры в меню приложения).

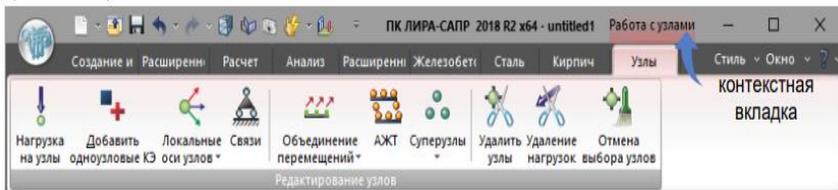


Рис. 3.5.2. Контекстная вкладка ленты Работа с узлами

Каждая вкладка состоит из панелей. Панель (рис. 3.5.1)- организация используемых команд в группы, подобные панелям инструментов. Панели ленты позволяют осуществлять быстрый вызов команд и при этом сокращают количество присутствующих на экране элементов интерфейса. Организация кнопок в панелях ленты Кнопки в панелях ленты организованы в нескольких видах:

- Большие кнопки с текстом;
- Большие кнопки с текстом в виде раскрывающихся списков (рис.3.5.3);

- Малые кнопки;
- Малые кнопки в виде раскрывающихся списков;
- Малые кнопки с текстом.

Раскрывающиеся списки кнопок панели ленты могут содержать целый ряд команд, близких по назначению, занимая при этом на панели ленты место всего одной команды. Команды, содержащиеся в раскрывающемся списке, можно запускать щелчком по главной кнопке или выбрав команду из самого списка, когда нажата кнопка разделения.

Существует несколько возможных способов работы раскрывающегося списка.

Раскрывающийся список с заменой (рис.3.5.3) - раскрывающийся список настроен как кнопка разделения, главная кнопка отображает последнюю использовавшуюся команду, которую впоследствии можно запустить уже не выбирая из списка, а просто щелчком по главной кнопке. Раскрывающийся список открывается щелчком по кнопке разделения.

Раскрывающийся список – после использования одной из команд раскрывающегося списка, главная кнопка не отображает последнюю использованную команду, а всегда работает одинаково. Раскрывающийся список открывается щелчком по кнопке разделения. В раскрывающихся списках Изополя C1z, Мозаика C1z, Мозаика жесткостных характеристик свай (вкладка Создание и редактирование, панель Жесткости и связи и панель Инструменты) щелчок по кнопке всегда приводит к раскрытию списка.

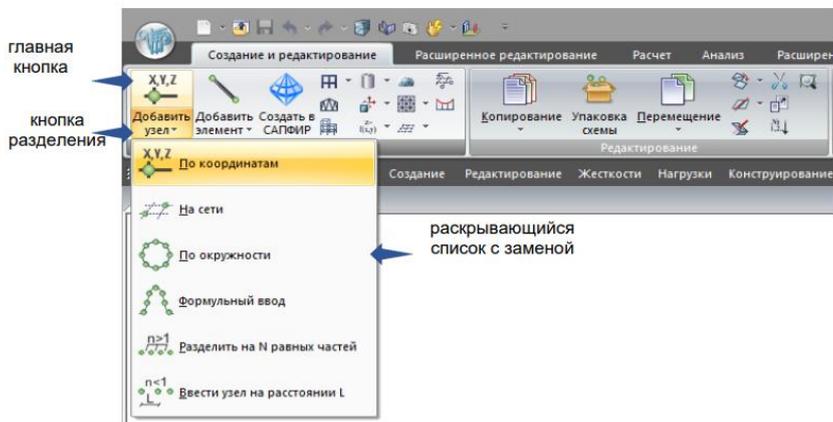


Рис. 3.5.3. Раскрывающийся список с заменой на панели ленты

Панель инструментов быстрого доступа (рис.3.5.4) – панель быстрого доступа располагается в верхней части окна приложения (вдоль ленты, над или под ней) и обеспечивает непосредственный доступ к определенному набору команд. Панель быстрого доступа является настраиваемой и содержит набор команд, не зависящих от вкладки, отображаемой в данный момент на ленте.

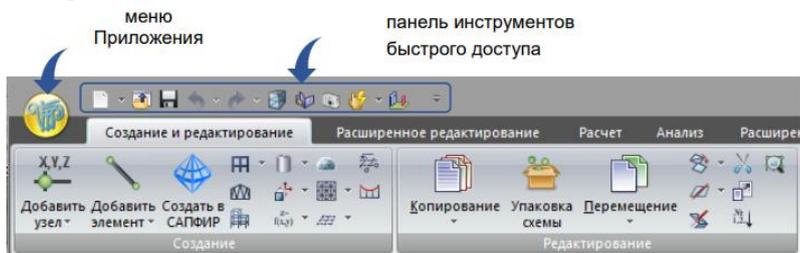


Рис. 3.5.4. Панель инструментов быстрого доступа

Меню приложения (рис.3.5.5) – меню для работы с файлами документа и настройкой параметров для графического отображения схемы и расчета.

На главной странице меню приложения находится список документов, которые открывались последними. Меню приложения содержит основные команды работы с файлами ВИЗОР-САПР, такие как: Создать, Открыть, Сохранить, Закрыть, Импорт/Экспорт и команды настройки параметров.

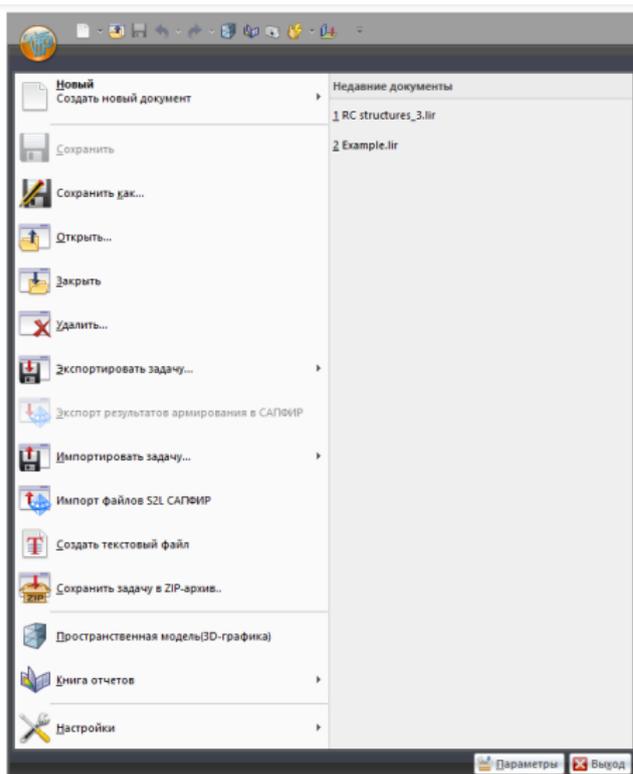


Рис. 3.5.5. Развернутое Меню приложения

Вкладка Создание и редактирование

Операции по созданию и базовому редактированию геометрии схемы, назначению жесткостей и формированию загрузок (рис. 3.5.6.).

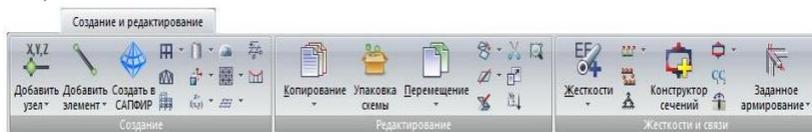


Рис. 3.5.6. Вкладка Создание и редактирование

Вкладка Расширенное редактирование

Операции по расширенному редактированию схемы, сборка схем, работа с блоками и суперэлементами, анализ, вызов и управление параметрами работы системы ГРУНТ, работа с линиями влияния, отображение мозаик для анализа геометрии и заданных свойств элементам расчетной схемы (рис.3.5.7).



Рис. 3.5.7. Вкладка Расширенное редактирование

Вкладка Расчет

Операции по заданию данных для статического, динамического и дополнительных расчетов, формирование таблиц, контроль параметров для расчета и запуск задачи на расчет (рис.3.5.8).

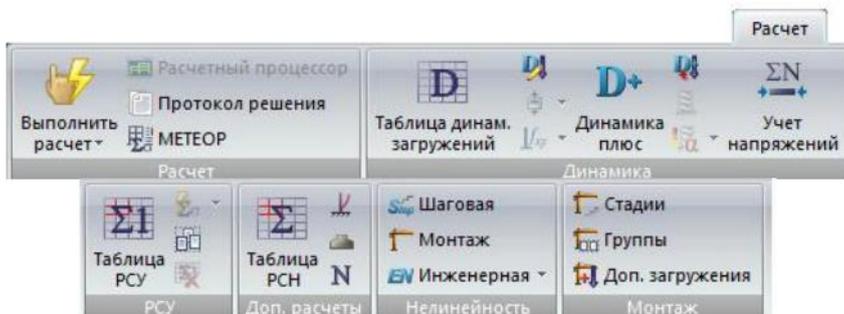


Рис. 3.5.8. Вкладка Расчет

Вкладка Анализ

Наиболее употребляемые функции анализа результатов: вывод на экран численной и графической информации о перемещении в узлах, усилиях и напряжениях возникающих в элементах расчетной схемы (рис.3.5.9).

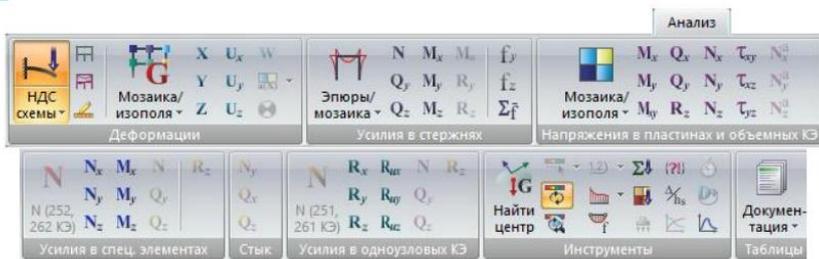


Рис. 3.5.9. Вкладка Анализ

Вкладка Расширенный анализ

Дополнительные функции анализа результатов расчета. Анализ усилий в специальных элементах, задание исходных данных, расчет и анализ результатов расчета дополнительных систем (рис.3.5.10).

Вкладка Железобетон

Операции по заданию исходных данных для конструирования, подбор арматуры и проверка заданного армирования в пластинчатых элементах и в сечениях стержневых элементов, анализ результатов, а также запуск локальных режимов (рис.3.5.11).

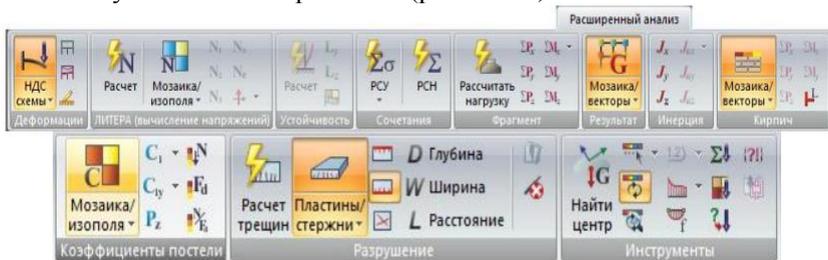


Рис. 3.5.10. Вкладка Расширенный анализ

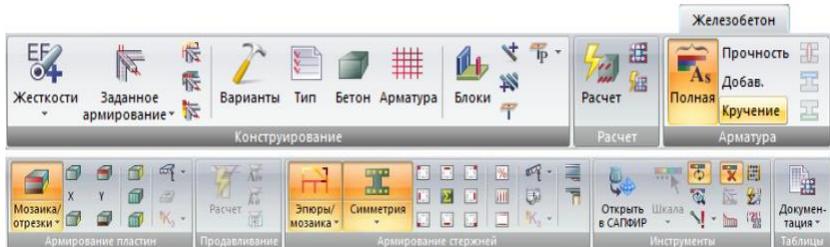


Рис. 3.5.11. Вкладка Железобетон

Вкладка Сталь

Операции по заданию исходных данных для конструирования, проверка и подбор стальных сечений и узлов, анализ результатов (рис.3.5.12).



Рис. 3.5.12. Вкладка Сталь

Вкладка Кирпич Операции по заданию исходных данных для конструирования, проверка прочности кладки и подбор армирования, анализ результатов (рис.3.5.13).



Рис. 3.5.13. Вкладка Кирпич

Контекстная вкладка Работа с узлами

Операции, применимые к узлам схемы, активируется при отметке узлов. Контекстная вкладка содержит команды только по созданию и редактированию схемы и не может быть вызвана из вкладок Анализ, Расширенный анализ, Железобетон, Сталь, Кирпич (рис.3.5.14).



Рис. 3.5.14. Работа с узлами

Контекстная вкладка *Работа со стержнями*

Операции, применимые только к стержням схемы, активируется при отметке стержней. Контекстная вкладка содержит команды только по созданию и редактированию схемы и не может быть вызвана из вкладок Анализ, Расширенный анализ, Конструирование (рис.3.5.15).



Рис. 3.5.15. Работа со стержнями

Контекстная вкладка *Работа с пластинами*

Операции, применимые только к пластинам схемы, активируется при отметке пластин. Контекстная вкладка содержит команды только по созданию и редактированию схемы и не может быть вызвана из вкладок Анализ, Расширенный анализ, Конструирование (рис.3.5.16).

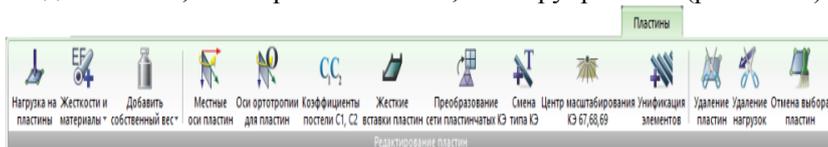


Рис. 3.5.16. Работа с пластинами

Панель инструментов *Выбор*

Операции, позволяющие отмечать различные узлы и элементы схемы для последующего выполнения над ними каких-либо действий (удаление, назначение жесткости, приложение нагрузок и т.д.). А также операции по отображению фрагмента расчетной схемы на экране путем фрагментации схемы или ее масштабирования (рис.3.5.17).



Рис. 3.5.17. Панель инструментов Выбор

Панель инструментов Проекция

Операции, позволяющие представлять расчетную схему на экране различными способами (рис.3.5.18).



Рис. 3.5.18. Панель инструментов Проекция

Вопросы для самоконтроля

1. Программный комплекс ЛИРА-САПР
2. Предназначение программы и его применение
3. Автоматизация процессов проектирования
4. Вкладки и панели ленты

ГЛАВА IV. ТРЕХМЕРНОЕ МОДЕЛИРОВАНИЕ

Тема 4.1. Способы трехмерного моделирования на компьютере. Введение программы 3D Studio MAX

Основные модули

- История программы 3ds max
- Предназначение программы 3ds max и его применение
- Первый запуск программы 3ds Max

Autodesk 3ds Max (ранее 3D Studio MAX) - компания Autodesk. Создатель, аниматор и создатель трехмерной графики, и полнофункциональный профессиональный редактор программного обеспечения. Включает мультимедиа получает самые современные инструменты для профессионалов и художников в этой области. Он работает в семействе операционных систем Windows (как 32-битных, так и 64-бит).

История программы

Первая версия сборника была разработана в 1990 году под названием 3D Studio DOS удален. Йост основанный программистом Гэри Йостом над производством коллекции Группа занималась независимой студией. Autodesk изначально разработал только комплект занимается добычей. Согласно источникам, Гэри Йост был из того периода. Из бесед с Эриком Лайонсом, директором по новым проектам Autodesk затем покинул свое рабочее место.

Первые четыре релиза назывались 3D Studio DOS (1990–1994). Затем сборник переписали для Windows NT и назвали 3D Studio MAX. переименован (1996-1999). Нумерация версий возобновлена.

3ds Max - это трехмерный компьютер разного вида и сложности. Модели, все для создания реальных или фантастических объектов в окружающей среде есть необходимые инструменты. В программе используются различные методы и приемы выполнения работы.

Применяется, в том числе:

- Полигональное моделирование, Редактируемый тест (редактирование поверхности) и Включены редактируемые полигоны (редактирование полигонов). Это самый распространенный как метод моделирования, известный сложными моделями и играми используется для создания моделей.

- Моделирование на основе рациональных В-сплайнов (NURBS) разнополюх (то есть следует отметить, что моделирование NURBS находится на том же уровне в 3ds Max просто сейчас им никто не пользуется);

- Моделирование на основе лицевых панелей (редактируемый патч) – вращение удобен для моделирования объектов;
- С использованием встроенной стандартной библиотеки и модификаторов моделирования;
- Surface является основной альтернативой NURBS на основе шлицевых моделирование с использованием модификатора;
- Использование модификаторов Extrude, Lathe, Bevel, Profile на основе шлицев моделирование или создание на основе сплайнов объектов Loft. Этот метод используется при моделировании зданий.
- 3D Studio MAX трехмерное моделирование и демонстрация - это новый этап создания (Визуализация). Это программа профессиональная качественная анимация и трехмерные модели.
- Использование двухмерных и трехмерных объектов. С этой программой качественные мультфильмы, известные науке.
- В 3D Studio MAX создаются объекты в области просмотра. Для этого необходимо выбрать нужный инструмент и переместите курсор в область построения. Курсор меняет форму по мере ввода.
- Специальная камера и для создания кинематографических эффектов на созданных объектах можно использовать осветительные приборы;
- Можно выбрать разные материалы для поверхности объекта, например можно придать прозрачную или шероховатую поверхность.
- Небольшая анимация при перемещении объектов, созданных на строительной площадке, могут быть сформированы. Для этого необходимо нажать на кнопку {Анимация} и выберите кадр. Перемещая объект в последний кадр. Затем нажмите кнопку {Play} на панели анимации.
- В результате происходит обмен кадрами и создается анимация. После чего следует сохранить анимацию в виде файла в памяти компьютера и в любом мы можем прочесть его с помощью программы, которая может читать видео изображения. Файл сохраняется в формате расширения *.avi.

Предназначение программы 3ds max и её применение

Autodesk 3ds Max - это один из самых мощных 3D-редакторов, которому под силу решать самые разнообразные задачи. Он по многим причинам считается программой архитекторов и дизайнеров интерьера, одна из которых - возможность качественного создания фотореалистичных визуализаций.

Современный 3ds Max - 3D-моделирование, качественная физика объектов на основе реальных законов физики, настройка анимации, система частиц и многое другое.

Основное направление программы -3D-моделирование - моделирование на основе примитивов, сплайнов, полигональное моделирование, NURBS-кривые для моделирования сложных органических форм и так далее.

3D-визуализация также играет немаловажную роль - программа интегрирована с мощными системами рендеринга, такими как V-Ray, Mental Ray, Arnold и многими другими.

Отлично выполненная функция анимации может качественно оживить сцену. Помимо настройки анимации объектов, можно создавать всевозможные эффекты на основе движения частиц: дым, дождь, огонь, симуляцию ткани и многое другое.

Применение программы 3ds Max

- 3D-моделирование, визуализация и презентация архитектурных сооружений.
- Дизайн, 3D-моделирование и визуализация интерьеров.
- 3D-моделирование мебели и предметов интерьера.
- Киноиндустрия и спецэффекты.
- В рекламе – создание рекламных роликов.
- Веб-дизайн и сайтостроение.
- Геймдев – локации и персонажи для компьютерных игр.

Это очень многогранная программа с неограниченными возможностями. Теперь давайте посмотрим на 3ds Max и познакомимся с его интерфейсом.

Первый запуск программы 3ds Max

После установки и запуска программы появится приветственный экран, который на нескольких слайдах показывает возможности графического пакета. Если вы не хотите при каждом запуске тратить время на закрытие этого экрана, уберите галочку снизу напротив надписи **Show this Welcome Screen at Startup** (Показать этот экран приветствия при запуске) (рис.4.1.1.).



Рис.4.1.1. Экран приветствия при запуске 3ds Max

Необходимо поставить галочку в нижнем правом углу, чтобы приветствие больше не беспокоило.

Интерфейс программы.

Давайте рассмотрим, из каких блоков состоит интерфейс программы (рис.4.1.2.):

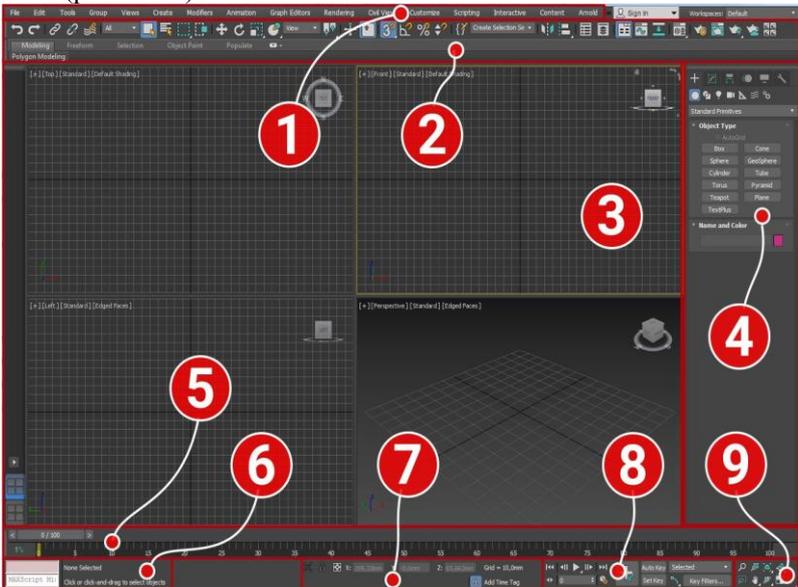


Рис.4.1.2. Интерфейс программы 3ds Max

1. Главное меню.
2. Основная панель инструментов.

3. Окна проекции.
4. Командная панель.
5. Track bar, он же Time Slider (шкала времени).
6. Панель для ввода команд встроенного языка 3ds Max - MaxScript.
7. Панель координат объектов.
8. Панель управления анимацией.
9. Панель навигации – позволяет управлять сценами во всех окнах проекции: приближать, отдалять и вращать объекты.

Теперь разберём эти панели несколько подробнее, рассмотрим, что с их помощью можно сделать и как их настраивать. На главном меню останавливаться не будем, в процессе работы мы ещё не раз к нему обратимся и рассмотрим настройки на конкретных примерах.

Основная панель инструментов

Основную панель вы найдёте под главным меню. Здесь находятся кнопки быстрого доступа к самым востребованным, по мнению разработчиков, функциям программы. При наведении указателя мыши на нужную кнопку появится подсказка с описанием функций этой кнопки (рис.4.1.3.).

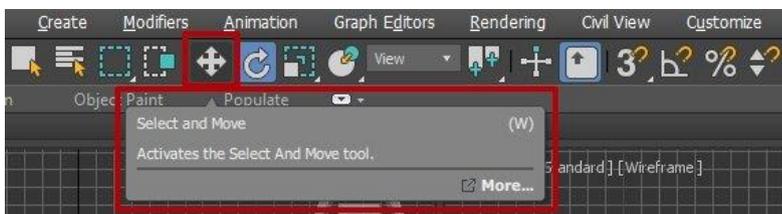


Рис.4.1.3. Основная панель инструментов в программе 3ds Max

Панели в 3ds Max, как и в любом другом современном 3D-пакете или графическом редакторе, переносятся и пристыковываются в любом месте интерфейса, что достаточно удобно и позволяет очень гибко настроить его под себя (рис.4.1.4.).

Панели вы можете расположить так, как вам удобно

Чтобы управлять панелями – показывать или скрывать их, нажмите правой кнопкой мыши на пустом месте основной панели. Появится контекстное меню (рис.4.1.5.):

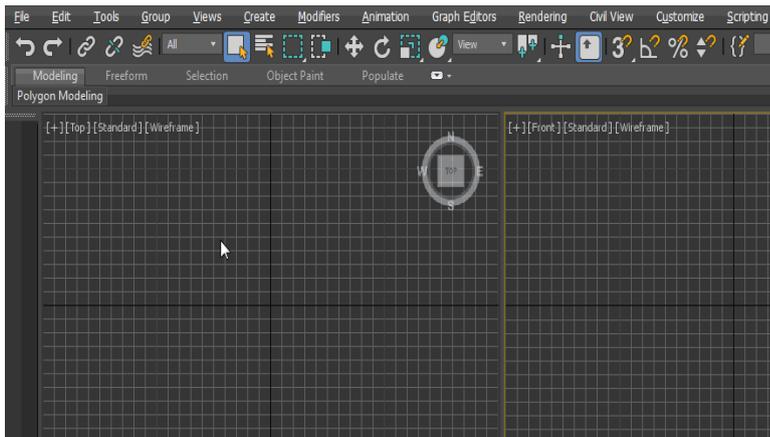


Рис.4.1.4. Основная панель инструментов в программе 3ds Max

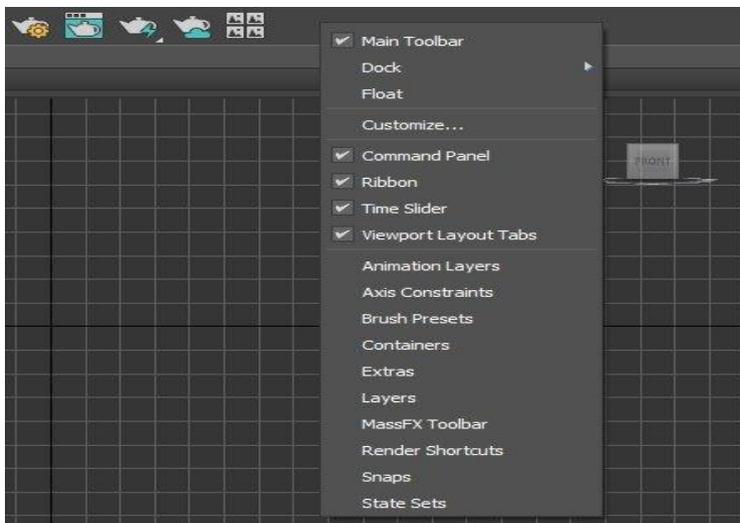


Рис.4.1.5. Контекстное меню с настройками для основной панели в программе 3ds Max

Галочками отмечены активные панели в интерфейсе. Если, например, вы случайно отключили *Main Toolbar* (основную панель инструментов), просто нажмите правой кнопкой мыши на панели главного меню – появится контекстное меню, где вы сможете выбрать

пункт *Main Toolbar* и вернуть отображение основной панели инструментов в интерфейсе (рис.4.1.6.).

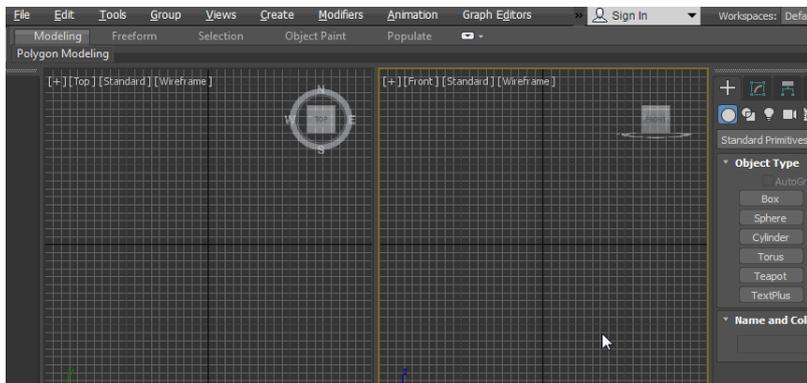


Рис.4.1.6. Возвращение основной панели инструментов в интерфейс программы

Помимо того, что вы можете переносить панели, прикреплять и откреплять их от элементов интерфейса или делать при необходимости плавающими, 3ds Max позволяет настраивать элементы этих панелей или создавать уникальные панели с нужными инструментами.

Создать новую панель можно двумя способами:

1. Необходимо нажать правой кнопкой мыши на основной панели инструментов и в выпадающем меню выберите пункт *Customize* (рис.4.1.7.).

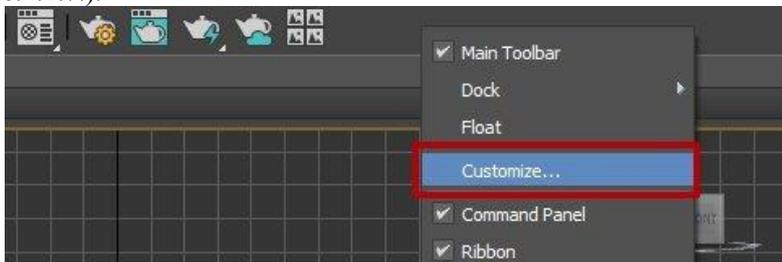


Рис.4.1.7. Создание новой панели

2. В главном меню выберите пункт *Customize - Customize User Interface* (рис.4.1.8.).

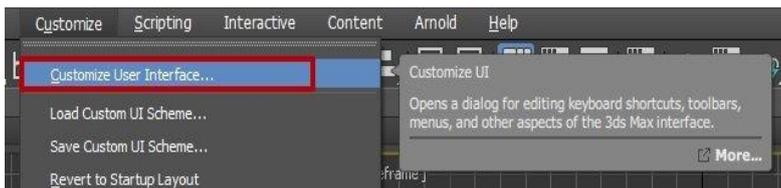


Рис.4.1.8. Создание новой панели

Далее в открывшемся окне выберите вкладку *Toolbars* (рис.4.1.9.).

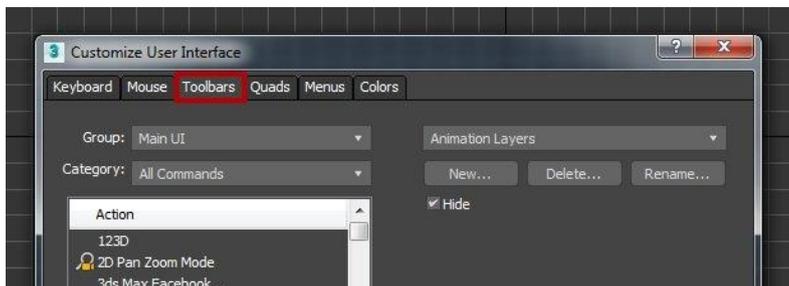


Рис.4.1.9. Вкладка Toolbars

Чтобы создать новую панель инструментов, нажмите кнопку *New*, в открывшемся окошке введите наименование панели, например, *New Tools* (рис.4.1.10).

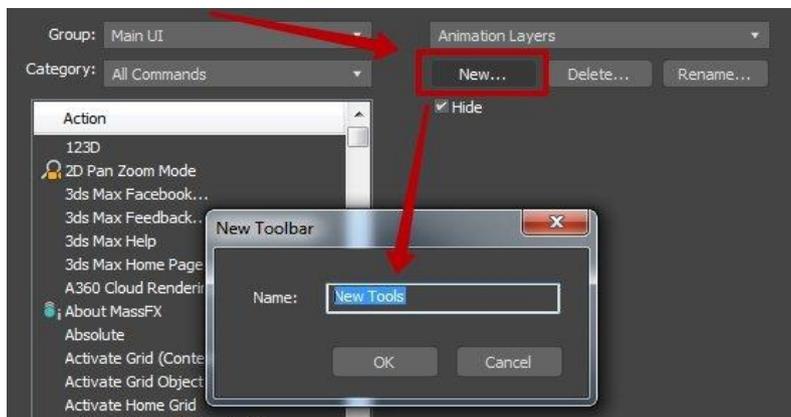


Рис.4.1.10. Создание новой панели инструментов

Задаём название своей панели инструментов. После подтверждения созданная панель будет доступна в рабочем пространстве программы (рис.4.1.11.).

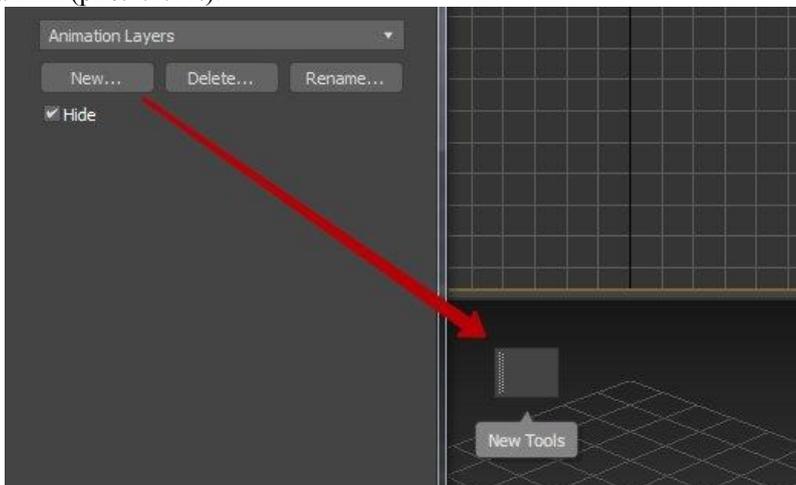


Рис.4.1.11. Создание новой панели инструментов

Ваша панель появилась в рабочем пространстве программы. Теперь нужно наполнить эту панель необходимыми инструментами, которые можно выбрать вот из этого списка (рис.4.1.12.):

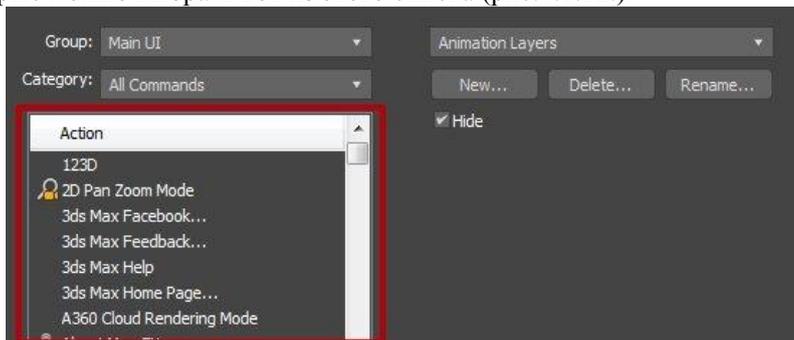


Рис.4.1.12. Заполнение новой панели инструментов

Инструменты, которые можно добавить в свою панель. Также можно выбрать нужную вам группу и категорию. Зацепите мышкой любой пункт из этого списка и перетащите на созданную панель (рис.4.1.13.).

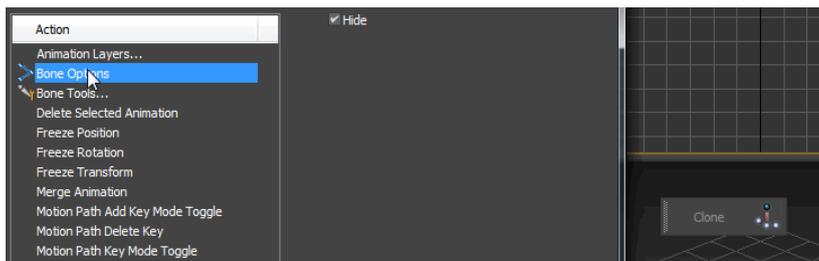


Рис.4.1.13. Группы и категории в панели инструментов

Наполняем панель инструментами. Таким образом можно создавать любое количество нужных вам панелей и формировать их так, как вам удобно, например, по функциональной принадлежности. Поначалу необходимость в кастомных панелях будет минимальной, необходимость в них придёт с наработкой опыта.

Окна проекции

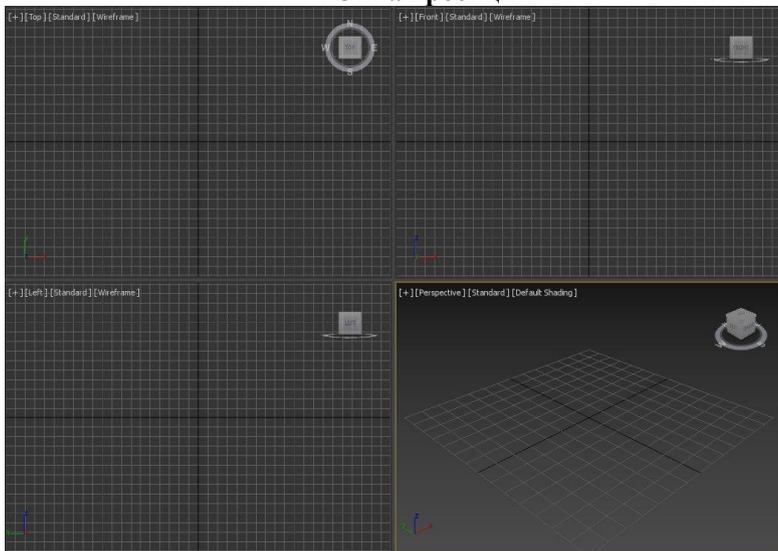


Рис.4.1.14. Окна проекции

Окна проекций занимают большую часть интерфейса программы. Всего доступны четыре вида отображения проекции - Вид сверху (Top), Фронтальный вид (Front), Вид слева (Left), Перспектива (Perspective). В контекстном меню можно выбрать любой нужный вам вид проекции (рис.4.1.14.-4.1.15.).

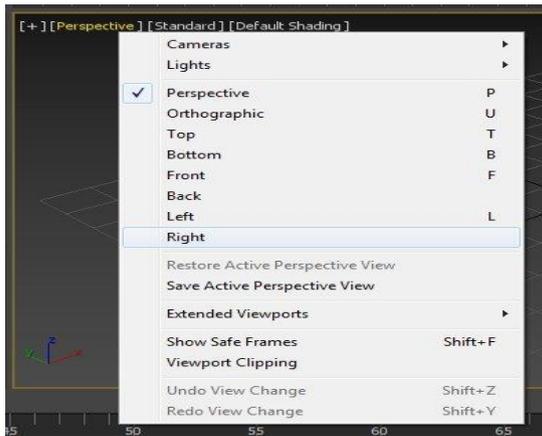


Рис.4.1.15. Вид проекции

Горячие клавиши отображения проекций в 3ds Max. Активное окно проекции подсвечивается жёлтой рамкой, чтобы вы понимали, какая проекция активна в данный момент (рис.4.1.16.).

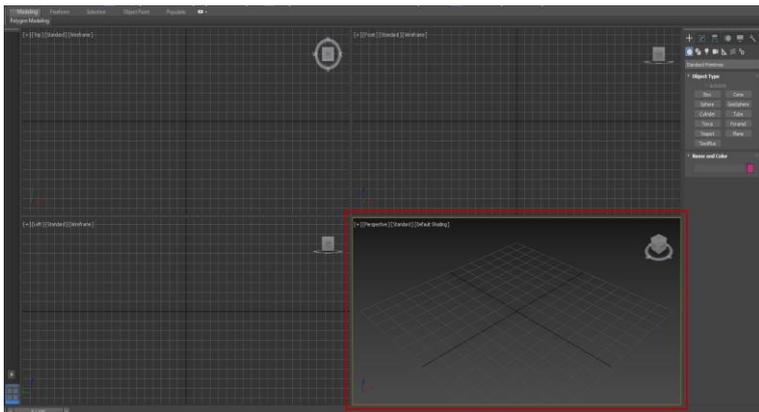


Рис.4.1.16. Активное окно проекции

Активная проекция подсвечивается рамкой. Для удобства работы над объектом или сценой активное окно можно развернуть на весь экран: достаточно нажать сочетание клавиш *Alt+W* на клавиатуре (рис.4.1.17.).

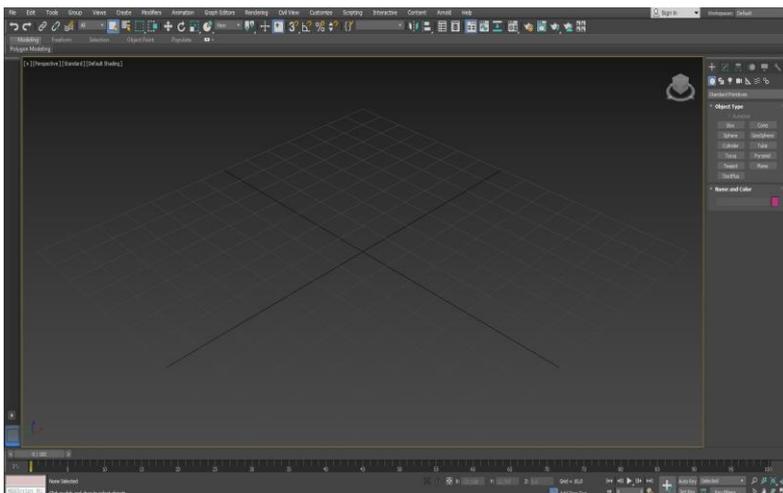


Рис.4.1.17. Окно проекции на весь экран

Разворачиваем активное окно на весь экран по Alt + W. Повторное нажатие этих клавиш вернёт раскладку окон в прежнее состояние.

Обратите внимание на трёхмерный кубик в правом углу экрана - это *ViewCube*, так называемый видовой куб, который позволяет менять проекцию (рис.4.1.18.).

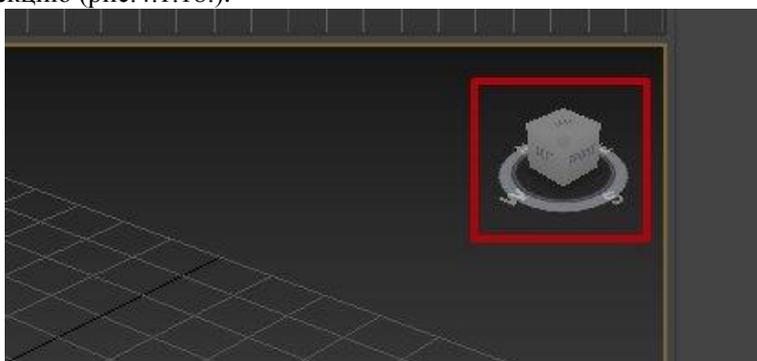


Рис.4.1.18. Изменение направление проекции

ViewCube - инструмент для быстрой смены проекции.

Если вы по какой-то причине решите его убрать (например, вы его случайно задеваете при работе), сделать это можно следующим образом – в окне проекции в левом верхнем углу нажмите на плюсик, появится контекстное меню. Там выберите *Configure Viewports*, далее

в открывшемся окне выберите вкладку **ViewCube** и уберите галочку напротив **Show The ViewCube** (рис.4.1.19.-4.1.20.).

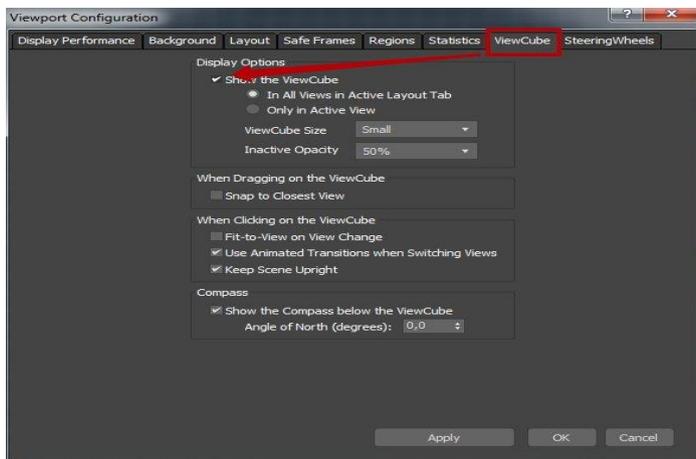


Рис.4.1.19. Изменение направления проекции

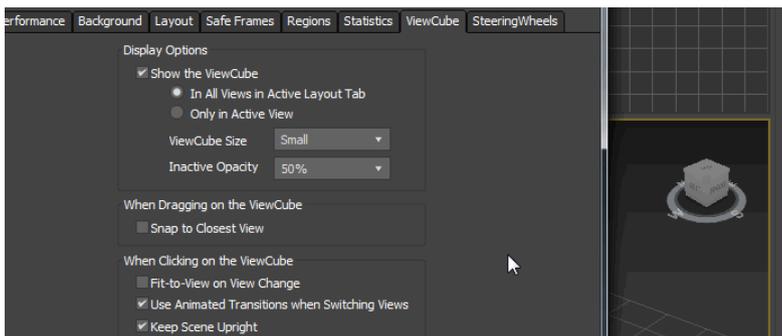


Рис.4.1.20. Изменение направления проекции

Убираем видовой куб и возвращаем на место.

Специальная панель **Viewport layout tabs** (*Макеты окон проекций*) по умолчанию находится в левом нижнем углу интерфейса программы. Она отвечает за разнообразную раскладку окон проекции.

Стандартом считаются четыре одинаковых окна, но разработчик предлагает на выбор предустановленные шаблоны. Если, например, для пользователя какое-то из окон важнее остальных, можно выбрать раскладку – одно большое и три маленьких окна, а также расположение основного окна слева или справа.

Убрать панель *Viewport layout tabs* или отобразить при необходимости можно там же, где и любую панель – в контекстном меню, которое мы рассматривали чуть выше (рис.4.1.21.).

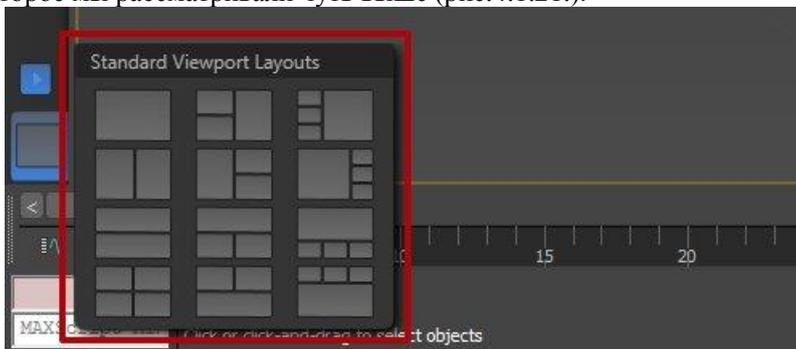


Рис.4.1.21. Viewport layout tabs

Специальная панель Viewport layout tabs

Это основные настройки окон проекции, которые необходимо знать для начала работы в 3ds Max.

Командная панель

Командная панель по умолчанию расположена справа от окон проекции. Она нужна для создания объектов, их модификации и дальнейших действий над объектами. Содержит шесть вкладок – *Create*, *Modify*, *Hierarchy*, *Motion*, *Display*, *Utilities*. Каждая вкладка имеет дополнительные вкладки и свитки с настройками (рис.4.1.22.).

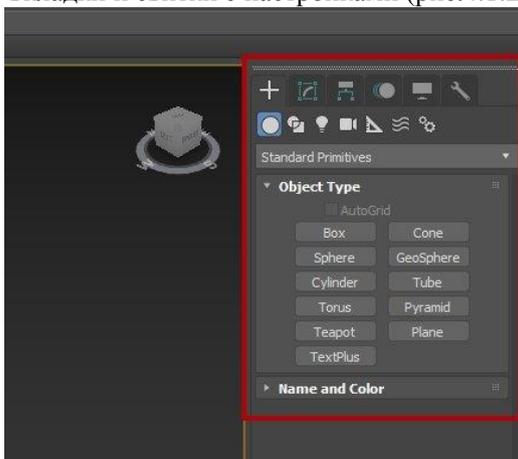


Рис.4.1.22. Командная панель

Более детально мы рассмотрим эту панель на практике в следующих уроках, когда будем работать с примитивами, полигональным моделированием и модификаторами.

Track bar

Это панель для работы с анимацией, проставление ключевых кадров и отслеживание тайминга. Находится в нижней части интерфейса 3ds Max (рис.4.1.23.-4.1.25.).



Рис.4.1.23. Track bar

Как правило, эта панель не нужна при обычной работе с объектами, тем более в начале изучения программы. Необходимость в ней возникает на завершающих этапах проекта, когда нужно продемонстрировать анимацию движения объекта в сцене, для чего сначала настраивают камеры.

Если, к примеру, вы работаете не на большом мониторе, а на ноутбуке, то эта панель может существенно уменьшать полезное рабочее пространство. Чтобы её спрятать, перейдите в главное меню выберите пункт *Customize – Show UI* и снимите флажок с пункта *Show Track Bar*

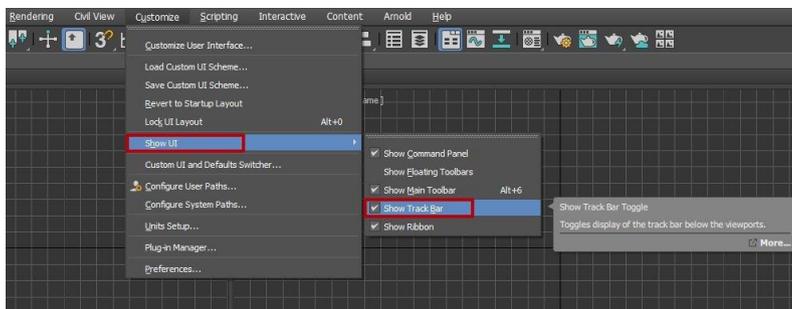


Рис.4.1.24. Track bar

Прячем панель Track bar: она может понадобиться только под конец проекта. Также можно просто открепить эту панель от нижней части интерфейса и нажать крестик в правом верхнем углу. Закрываем панель Track bar кликом по крестичку.

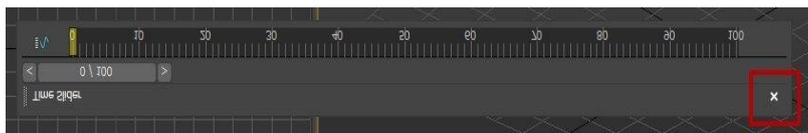


Рис.4.1.25. Track bar

Панель для ввода команд встроенного языка 3ds Max - MAXScript

MAXScript нужен для автоматизации рутинных задач, создания новых инструментов редактирования и решения задач по оптимизации использования существующих функций. Также через скриптовую документацию (API) можно контролировать и модифицировать геометрию объектов, текстуры, анимацию и так далее. С помощью MAXScript можно создавать всевозможные плагины и утилиты для выполнения нестандартных задач.

Очень нужный в работе инструмент, но, конечно, с оговоркой, что нужно разбираться в программировании. Но даже если вы не особо в этом сильны, MAXScript может пригодиться для использования уже кем-то разработанных скриптов, как платных, так и бесплатных, которых в сети достаточно много.

Панель координат объектов и единицы измерения

Очень полезная панель для контроля координат объекта и отслеживания единиц измерения сетки координат. Например, нам нужно переместить куб с того места, где он находится в данный момент, на место с координатами 0 по всем осям (рис.4.1.26.). Делается это так:

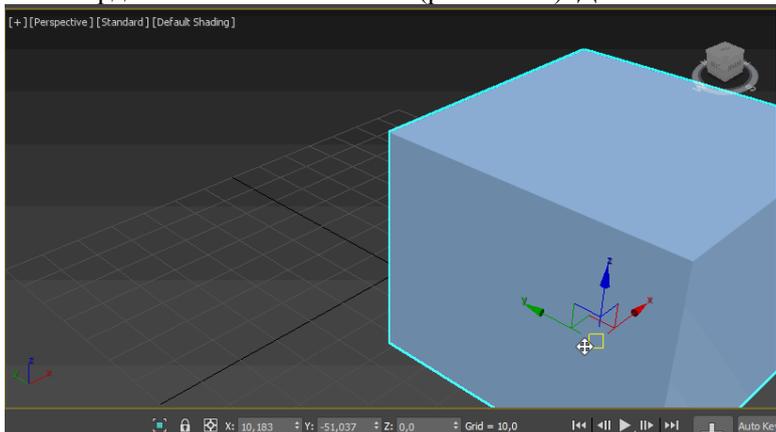


Рис.4.1.26. Панель координат объектов

Переносим куб по заданным осям координат.

Подробнее сетку координат и оси мы также рассмотрим на практике в дальнейших уроках.

Не менее полезный пункт **Grid** (*единицы измерения*), который находится рядом с полями координат и указывает, чему равен шаг сетки координат, то есть один её квадратик (рис.4.1.27.). Сейчас он равен 10 непонятным единицам.

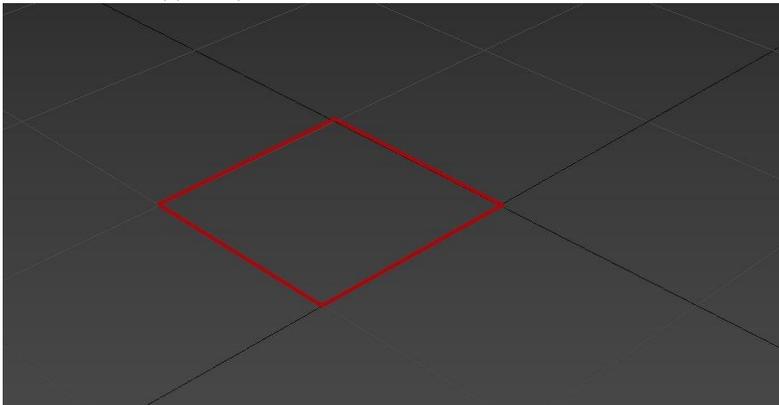


Рис.4.1.27. Сетка координат

Когда работа идёт над проектом, где важны реальные значения в конкретных единицах, например, миллиметрах, то перед началом работы обязательно необходимо настроить эти значения. Иначе впоследствии придётся подгонять и масштабировать сцену.

Для этого необходимо перейти в главное меню, выберите уже знакомый пункт **Customize – Unit Setup** (рис.4.1.28.-4.1.31.).

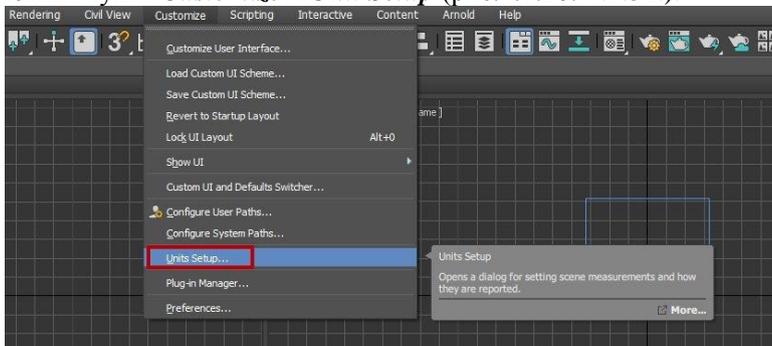


Рис.4.1.28. Настройка элементов в 3ds Max

В открывшемся окне выберите *Метрическую систему (Metric)*, в выпадающем списке выберите миллиметры.

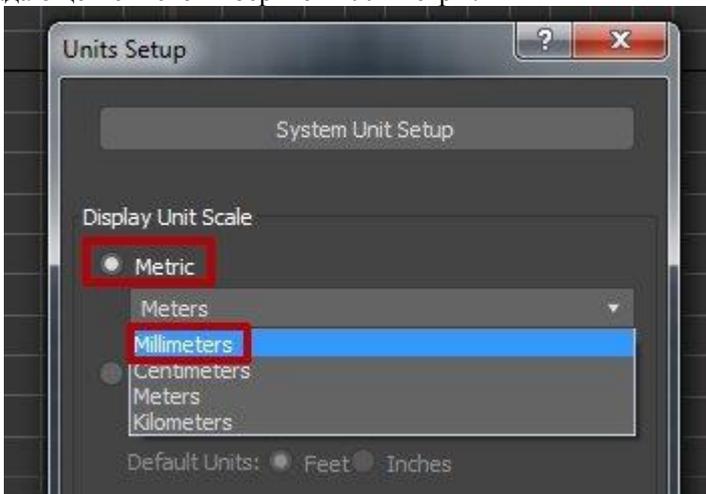


Рис.4.1.29. Настройка элементов в 3ds Max

Настраиваем метрическую систему и выбираем миллиметры

Затем нажмите на кнопку *System Unit Setup*, которая находится в этом же окне чуть выше, и также выберите миллиметры.

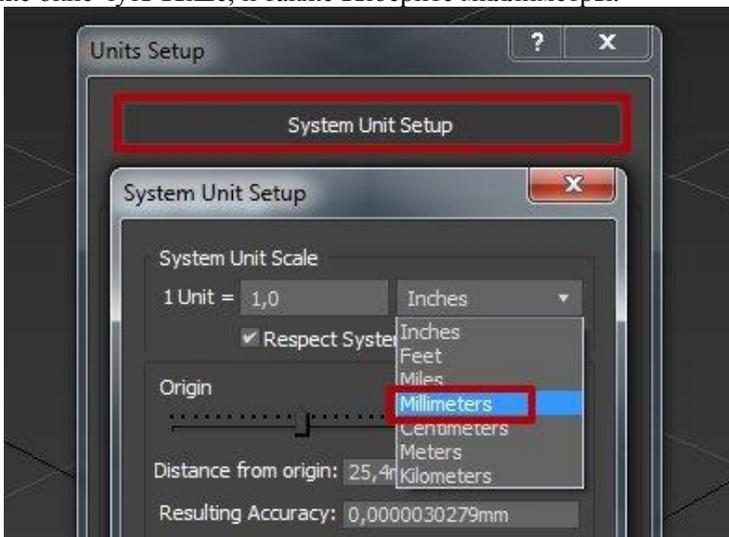


Рис.4.1.30. Настройка элементов в 3ds Max

Настраиваем системные единицы измерения
На нижней панели также произошли изменения.

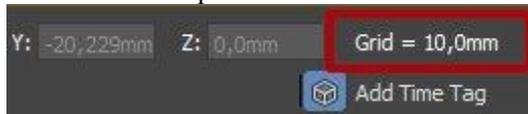


Рис.4.1.31. Настройка элементов в 3ds Max

Теперь сетка имеет шаг 10 мм. Настройка привязки является тоже достаточно важным.

Привязки нужны для максимальной точности стыковки объектов относительно друг друга, для точности построения новых сцен по шагу сетки и так далее. Привязки бывают нескольких видов -2, 2.5, 3.

Найти привязки можно на основной панели инструментов под панелью главного меню (рис.4.1.32.-4.1.33.):

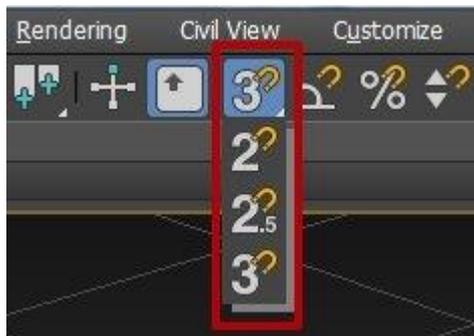


Рис.4.1.32. Настройка привязки в 3ds Max

Чтобы выбрать вид привязки, необходимо нажать на левую кнопку мыши на иконку привязки и, удерживая её, выбрать нужный вид.

Чтобы настроить привязку, необходимо нажать на правую кнопку мыши на иконку, появится окно настроек. В нём необходимо отметить нужные привязки, к примеру три основных Vertex (Вершина), Edge (Грань), Midpoint (Средняя точка):



Рис.4.1.33. Настройка привязки в 3ds Max

Мы рассмотрели основные панели интерфейса программы 3ds Max, выяснили, как можно создавать свои панели и гибко настраивать их под себя. Также теперь вы знаете, как менять окна проекции и разворачивать на весь экран. Понимание того, как устроен интерфейс программы и куда зайти, чтобы что-то изменить, нужно, чтобы в дальнейшем ваше взаимодействие с программой было максимально удобным.

Панель управления анимацией

Панель отвечает за настройку, создание и контроль анимации (рис.4.1.34.).



Рис.4.1.34. Панель управления анимацией в 3ds Max

Перечислим некоторые основные функции этой панели – просмотр анимации, переход на следующий и предыдущие кадры, переход на конкретный кадр, добавление ключевых кадров, настройка плавности анимации, фильтры и так далее.

Способы трёхмерного моделирования.

Общие понятия и определения.

3D моделирование - это процесс создания трёхмерной модели объекта. Задача 3D-моделирования: разработать визуальный объёмный образ желаемого объекта. При этом образ может как копией готового (известного) объекта, так и разработанный с нуля. Для создания трёхмерной графики необходимо знать расположение объекта, которое определяется системой координат. Основной является декартова система координат. В трёхмерной системе координат 3D-3-dimensional оси обозначаются как X, Y, Z, причём Z ось перпендикулярна плоскости XY.

В разных программах ориентация Z оси может быть различной. Местоположение объектов, выраженных по отношению к системе координат XYZ, называется мировой системой координат. Для создания объёмного изображения существует несколько подходов:

1. От плоскости к объёму (когда рисуют плоский объект и для создания трёхмерного образа рассматривают объект с различных сторон также на плоскости, пример - чертежи).

2. От объёма к плоскости (в нём изначально создаётся трёхмерный образ и для получения серии плоских картинок делают снимки этого трёхмерного объекта с различных ракурсов, положений и т. д. Принцип реализуется в 3Ds Max, Cinema).

Достоинства и недостатки трёхмерной графики.

Достоинства: реалистичность, широкая сфера применения, свобода трансформации объектов.

Недостатки: значительный объём файлов, программная зависимость.

Для создания трёхмерных объектов наиболее распространённым способом является построение фигур из сетки полигонов (polygon). Полигон характеризуется вершинами (vertices), рёбрами (edges), гранями (faces). Объект, состоящий из множества полигонов, представляет собой полигональную сетку, отображение которой может быть полным и неполным. Полигональные сетки могут строиться из треугольников или прямоугольников. Поверхность сетки определяется с помощью дополнительных атрибутов. Атрибуты поверхности могут быть простыми (сплошной цвет) и сложными (цвет с эффектом блеска). Поверхность также может быть представлена с помощью одного или более растрового изображения, которые называют текстурными картами (текстурами).

В совокупности свойства поверхности именуется как материалы. Наличие одного или более источников освещения позволяет пред-

ставить объект в более естественной форме. Пространства с объектом и источниками освещения носят названия сцены освещения. Так как полигональные сетки строят по координатам своих вершин, преобразование объектов осуществляют без отдельной прорисовки каждой его вершины с помощью матриц, которые позволяют изменять размеры объектов, их поворот и движение без фактического изменения значений в его вершинах.

Каждая сцена для рисования обладает точкой просмотра, которая визуализируется с помощью камер. Шейдер – это программа, выполняемая на графическом процессоре в процессе обработки кадра. Используется для определения конечных параметров объекта или изображения. Она может включать в себя описание поглощения или рассеяния света, наложение текстуры, смещения поверхности и т. д. Например, Шейдеры могут быть использованы для рисования поверхности кирпичной кладки на абсолютно плоской поверхности.

Выделяют следующие 5 этапов:

1. Моделирование - это создание объектов, которые будут на сцене. Выделяют следующие типы моделирования: о Моделирование на основе примитивов (под примитивами понимают простейшие параметрические формы: углы, сферы, пирамиды).

При визуализации эти объекты преобразуются в полигоны, но получаемая поверхность выглядит более гладкой за счёт специальных алгоритмов закраски. о Моделирование на основе сечений. Объекты на основе сечений названы по аналогии с судостроением, в котором применяется натягивание поверхности на произвольное сечение. Сечение или плоские формы в этом способе располагают вдоль некоторого пути. о Моделирование, основанное на использовании булевых операциях (пересечение, вычитание). Основой служат поверхности. При этом выделяют следующие поверхности: многоугольные каркасы, лоскутки (сплайн-моделирование), в этом в случае объекты изменяются с помощью контрольных точек. Образующие сплайны располагаются по краям создаваемой поверхности. Технология создания плавных форм и моделей, принцип: с помощью управляющих вершин можно воздействовать не только на крайние (контрольные) точки, но и на любую локальную область поверхности. Применяется для создания образов животных, людей.

Моделирование по поверхности сплайновой сетки. При этом создаётся совокупность сплайнов в виде каркаса, на основе которого формируется поверхность.

2. Текстурирование - это придание поверхностям модели вида реальных материалов (дерево, металл, пластика). В процессе создания

простейших примитивов каждому из них назначается цвет, который на самом деле не является цветом поверхности, а обозначает цвет каркасной структуры. Чтобы после визуализации объекта он стал реалистичным, применяют редактор материалов. В редакторе можно установить реальный цвет объекта, при этом он может быть основным (определяет покрытие всего объекта), обтекающим (определяет влияние фонового освещения), зеркальным (определяет наиболее яркие блестящие участки поверхности объекта) и т. д. В процессе создания материалов могут быть использованы карты текстур (растровые изображения реальных объектов) и процедурные карты (изображения, которые генерируются программным путём). В процессе создания объектов могут накладываться несколько карт одновременно. Это определяет эффекты текстурирования. Точное размещение материалов на поверхности объектов достигается картами проецирования. При создании материалов определяются такие свойства объектов, как отражение, преломление, прозрачность. При этом можно изменять силу света, тип поверхности. Это реализуется с помощью спец. алгоритмов.

3. Освещение - это добавление и размещение источников света аналогично студийной съёмке. Благодаря освещению можно сформировать тени объектов сцены, изменить свойства отображения материалов, общее настроение сцены.

4. Анимация - это процесс создания движения путём просмотра быстро сменяющихся кадров (изменение во времени каких-либо свойств объектов, например положения в пространстве, размеры, и материалов, например цвет, прозрачность). Для создания иллюзии движения зачастую прибегают к математическому описанию этого движения.

5. Визуализация необходима для формирования окончательного изображения. Операция носит название рендеринга. При реализации учитывается: о качество изображения, при этом под качеством изображения понимаются эффекты сглаживания, создание скруглённых диагональных линий (рёбер), количества шагов в полигональной сетке; о освещение, например: объёмный свет, прожекторы и их количество и т. д. Чем более сложные эффекты освещения применены, тем более значительные ресурсы требуются для вычислений о размер изображения, при этом под размером может пониматься как габаритное изображение, так и его разрешение в пикселях.

Полигональное моделирование

Полигональное моделирование позволяет показать объект (его трёхмерное изображение) под любым углом, а также раскрыть особенности его конструкции.

Моделирование на основе сплайнов

Объектом и формой трёхмерного объекта могут быть двухмерные и трёхмерные кривые Сплайны

Моделирование на основе неоднородных рациональных B-сплайнов (Non Uniform Rational B-Splines NURBS)

Методы построения NURBS поверхностей Основные задачи, связанные с представлением трёхмерных тел:

- построение сцен с изображениями существующих объектов;
- синтез изображения, не существующего в природе объекта.

Лоскутное моделирование

Моделирование на основе лоскутов основано на использовании поверхностей, форма которых контролируется при помощи решётки деформаций.

Моделирование составных 3D-объектов

Типы составных объектов: Morph (морфинговые), Scatter, Conform, Connect, Blob Mesh (капельный каркас), ShapeMerge (слитые), Boolean (булевы), Terrain (ландшафтные), Mesher (смешивающие), Loft (опорное сечение).

Вопросы для самоконтроля

1. Элементы интерфейса
2. Окна проекций
3. Изменение размещения структуры окон проекций
4. Управляющие элементы состояния
5. Инструменты трансформации

Тема 4.2. Редактор материалов. Источник света. Установка камер. Основы анимации. Рендер

Основные модули

- Основы текстурирования, типы и свойства материалов
- Типы источников освещения.
- Параметры настройки осветителей и камер.
- Модели затенения.
- Типы источников освещения.
- Основы анимации. Рендер.

Тестирование позволяет придать поверхности вид реального материала. Цвет - один из простейших свойств материала. Цвет может быть:

- основным, определяющим покрытием всего объекта обтекающим, т.е. определяющим влияние фонового освещения;
- зеркальным, т.е. определяющим цвет наиболее ярких участков блестящей поверхности.

При создании материалов определяются такие свойства объектов, как:

- отражение (Reflection)
- преломление (Refraction)
- прозрачность (Opacity)

Кроме этого, можно задать реакцию поверхности на свет, свойства отраженного света и его силу. Параметры материала задаются либо с помощью числовых значений (цвета) либо с использованием текстурных карт (растровых изображений).

Материал может присваиваться как ко всей поверхности 3D-объекта, так и к её определённым граням или участкам (составной материал). Свойства материалов (состоит из 3 составляющих):

- отраженный цвет или цвет в тени (Ambient);
- рассеянный цвет или цвет на освещённой стороне (Diffuse);
- зеркальная составляющая или цвет бликов (Specular).

При этом настройки бликов задают интенсивность блика, ширину, величину смягчения блика в отраженном свете.

Для визуализации сцен со сложным освещением в свойствах материала можно задавать самосвечение (Self-Illumination), который даёт иллюзию самостоятельного свечения при отсутствии компонента затенения (Ambient).

Например, светящаяся гирлянда на новогодней ели. По умолчанию все материалы непрозрачны. В стандартных опциях изменяя величину прозрачности можно получить эффект стекла.

Чтобы назначить материал объекту надо выполнить действия:

- Выделить объект (группу объектов);
- Открыть редактор материалов;
- Выбрать ячейку с нужным образцом материала;
- Перетащить материал на выделенный объект либо назначить материал выделению (Assign Material to Selection).

Для материала характерны 3 уровня назначения:

- Hot (горячий). Для объектов в сцене;
- Warm (теплый). Копия одноименная горячего материала;
- Cool (холодный). Материал, который не назначен ни одному объекту текущей сцены;
- Распределение текстур по объектам.

В процессе создания материалов широко используются процедурные карты, т.е. изображения, которые генерируются программным путём. Для точного размещения материала на поверхности используется понятие проекционных координат, которые указывают, как растровые карты будут размещены на поверхности объекта. Такой принцип позволяет сэкономить время, а моделирование объектов. Например, окно либо оконную решетку можно моделировать с помощью полигонов, либо просто присвоить материал на основе соответствующего изображения стандартным примитивам типа Box. Получение рельефности при этом основано на разности яркости цветов.

Координаты проекционные можно задать несколькими способами:

- Через проекции UVW;
- Через уточнённый модификатор Unwrap.

Текстуры нельзя накладывать на объекты. Её можно наложить только на объекты с определённым материалом (вначале надо применить текстуру к материалу, а затем материал к объекту сцены).

Маппинг (наложение карты) состоит в наложении текстурной карты непосредственно на лицевую сторону объекты, либо применение карты при создании объёмного эффекта.

Используются следующие системы проецирования материала на поверхность:

- Плоские (Planar) - используются для проецирования растровых изображений на плоскость.
- Цилиндрические (Cylindrical) - применяются к объектам цилиндрической формы.
- Сферические (Spherical) - применяются к объектам, имеющим округлую форму.

- Обтягивающие (Shrink Wrap) - специальная система координат, применяемая для проецирования текстур на объекты сложной формы. Является сферической, но обеспечивает усечение углов карты текстуры и соединение их в 2 диаметрально противоположных точках полюсах, что дает минимальное искажение рисунка.

- Прямоугольные трёхмерные (Box) - применяются для отображения текстур на объектах, у которых грани расположены под углом 90°. В отличие от плоской системы координат исключается растяжение рисунка на вытянутых гранях.

- Координаты граней (Face) - обеспечивает размещение отдельных копий текстурной карты в центр каждой грани объекта.

Модификатор UVW-mapping позволяет расположить материал на поверхности объекта с помощью специального средства – габаритного контейнера Gizmo. Изменение размеров габаритного контейнера приводит к повторению изображения по всей плоскости объекта. Отключается опцией Tile (повтор).

Для приведения в соответствие размеров графического изображения используется кнопка Bitmap Fit.

Модификатор Unwrap UVW - это развернутый модификатор координат UVW, предназначенный для сложного редактирования координат наложения карты на объект. Подобъектом модификатора является объект Select Face (выделить грань). Выделение подобъектов позволяет отобразить их в диалоговом окне Edit UVW в стеке модификаторов и назначить на каждый из них свой материал. Общее правило создания профессиональных кадров:

- Для наложения текстур на сложные объекты применяются специальные текстурные развертки.

- Чтобы придать объекту реалистичность используйте эффекты износа и грязи. Чередуйте карты зеркальности.

- Отделение, а также дублирование и смещение граней элемента объекта позволяет добавлять дополнительные детали на этот объект и создать иллюзию того, что фрагменты являются частью целого.

- Если нельзя дублировать грани можно воспользоваться вспомогательным многоугольником, расположенным очень близко к каркасной сетке, наложить на него карту непрозрачности, скрывающую элементы, которые не должны отображаться.
- Вместо создания геометрических форм лучше использовать карты-текстуры, что ускоряет визуализацию.
- Для передней поверхности объекта обязательно должны указываться координаты наложения, иначе углубления превратятся в выпуклости.

Источник света. Установка камер.

Точечные источники света – излучают свет по всем направлениям. При их задании необходимо определять положение в пространстве. Прожекторы – излучает свет не во всех направлениях, а в пределах определённого конуса. При задании необходимо определять не только местоположение источников света, но и величину угла в вершине конуса света. Примером прожектора может являться настольная лампа. Направленный свет (directional) – представляет собой свет, идущий из одного определённого направления, задаваемый либо в векторной форме, либо при помощи координат точек, лежащих на одном из его параллельных лучей.

Фоновое освещение (background) окружающий объект освещенное от удалённых источников, чьё положение и характеристики не известны.

Помимо стандартных источников освещения 3DsMAX также использует 2 системы имитации света для освещения открытых сцен (рис.4.2.1.):

- Система солнечного света;
- Система дневного света.

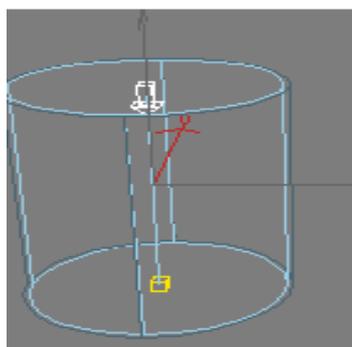
Фотометрические источники света, которые отличаются от стандартных тем, что позволяют точно установить силу света, расстояние до источника света и интенсивность светового потока, моделируя реальные фотометрические величины.



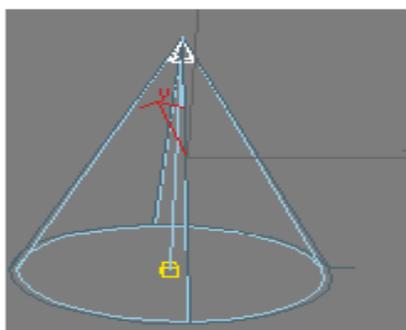
Всенаправленный источник



Система солнечного света



Нацеленный направленный источник



Нацеленный прожектор

Рис.4.2.1. Источников освещения 3DsMAX

Кроме освещения сцены и объектов сцены используется также понятие глобальной освещённости.

Глобальная освещённость (global illumination) – суммарная освещённость объектов трехмерной сцены, которая создается как прямыми лучами света, испускаемыми осветителями, так и лучами, многократно отраженными от различных тел сцены.

Признаки:

- отсутствие потребности в большом количестве осветителей;
- наличие полупрозрачных теней за счет дополнительного подсвечивания затененных областей отраженными лучами света;
- цветовое тонирование объектов светом, отраженным от близлежащих объектов сцены, имеющих выраженную раскраску.

Алгоритмы расчета глобальной освещенности:

1. Трассировщик света.
2. Перенос излучения.

Путь: меню Rendering (Визуализация) – Advanced Lighting (Улучшенное освещение) – Light Tracer (Трассировщик света) и Radiosity (Перенос излучения).

Визуализатор mental ray: окно Render Scene – вкладка Indirect Illumination (Непрямое освещение) – свиток Final Gather (Окончательное накопление) – Enable Final Gather.

Принцип действия алгоритма трассировщика света: плоская проекция сцены разбивается на элементарные участки из точек трёхмерного пространства сцены соответствующим центрам участков разбиения, испускаются воображаемые пучки случайно ориентированных лучей света.

Для каждого такого участка освещённость вычисляется как сумма освещённостей прямыми лучами света от осветителя, а также от тел сцены которых достигли лучи, испускаемые из центрам данного участка.

- Данный алгоритм обладает следующими свойствами: позволяет рассчитывать многократное отражение световых лучей (повышает качество изображения, но замедляет визуализацию).

- Результат расчёта глобальной освещённости зависит от конкретного вида сцены, т. е. если изменяется вид сцены, то производится повторный перерасчёт.

- Наилучший эффект достигается при визуализации сцен вне помещения (так как нет многократного отражения лучей).

Алгоритм переноса излучения: выбор точек сцены, из которых спускаются пучки отражённых лучей, определяются положением тел в сцене, а также положением оболочек, из которых выходят воображаемые лучи света. Данный алгоритм не зависит от текущего вида сцены.

Свойства алгоритма:

- Сложен в использовании, поскольку требует подготовки источников и материалов.

- Наилучший эффект достигается для фотометрических осветителей

(лучше подходит для визуализации сцен внутри помещения).

- Результаты расчета не зависят от конкретного вида сцены.

Параметры настройки осветителей и камер.

3DsMAX позволяет управлять цветом и градациями освещения, а также исключать из сцены объекты, на которые свет не должен падать. Наиболее часто встречаются 2 варианта освещения:

Настройка источников света:

- Type (Тип);
- On (Включить);
- Shadows (Тени);
- Color (Цвет);
- Include/Exclude (Включить/Отключить);
- Multiplier (Множитель);
- Contrast (Контраст);
- Soften Diff. Edge (Смягчить границу рассеяния);
- Diffuse (Диффузная область);
- Specular (Зеркальный блеск);
- Ambient Only (Только подсветка);
- Hot Spot and Falloff (Пятно освещения и спад);
- Attenuation (Затухание);
- Decay (Спад);
- Projector Map (Проецируемая карта);
- Targeted (Цель).

Трёхточечная использует три источника: ключевой, контурный и заполняющий.

Ключевой является основным, самым ярким освещающим большую часть сцены, благодаря ему объекты в сцене отбрасывают тень.

Контурный свет определяет глубину пространства, разделяя основные предметы и фон, обычно располагается позади объектов сцены и слабее ключевого.

Заполняющий – общий рассеянный свет, придаёт мягкость предметам и теням. Местная, применяется для освещения отдельных участков сцены, когда их невозможно осветить трёхточечным освещением.

При освещении важно управлять расстояниями до источника света и углом установки источника света по отношению к сцене, учитывая свойство поверхности.

Модели затенения.

Тени позволяют увеличивать реалистичность изображения. Имеют следующую структуру:

1. Блик – место на освещённой части предмета, где свет отражается непосредственно от источника (на глянцевых поверхностях всегда присутствует непосредственное отражение источников света).

2. Свет – освещённая часть предмета.

3. Полутень (полутон) – переход между светом и собственной тенью, а также переход между тенью и рефлексом.

4. Тень – самое тёмное место, на которое не попадает ни прямой свет, ни отражённый.

5. Рефлекс – место собственной тени на предмете, на которое попадает отражённый свет от другого освещённого предмета (от поверхности) или от освещённой части того же предмета.

6. Падающая тень – самая тёмная градация среди теней, по мере удаления предмета границы размываются, сила тона слабеет.

Линии, ограничивающие тень, называются границей (контуром) тени. Собственной тенью называется неосвещённая, обращённая от источника света часть поверхности предметов.

Грани, затемнённые собственной тенью, являются не лицевыми, если точку наблюдения совместить с точкой источника света.

Падающая тень – это область на обращённой к источнику света части поверхности предмета, закрытая от лучей света другим предметом и частью данного предмета. Чтобы найти такие тени нужно построить проекции всех не лицевых граней относительно источника света на сцену. Центр проекции находится в источнике. Точки пересечения проецируемой грани со всеми другими плоскостями образуют многоугольники, которые помечаются как теневые и заносятся в структуру данных.

Построение теней.

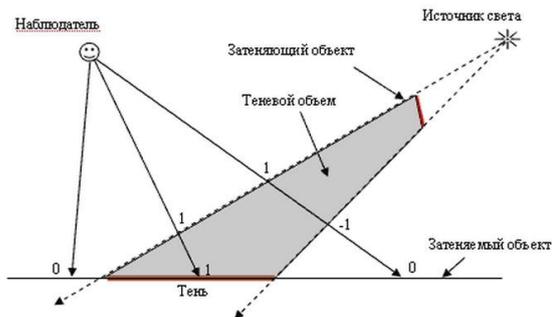


Рис.4.2.2. Источников освещения 3DsMAX

При синтезе изображения на компьютере обычно рассматриваются точечные источники света, которые создают только полную тень, а не полутень. На точность вычислений влияет положение источника света и чем более удалён источник, тем проще проецировать тень (в бесконечности хороший вариант). Самой сложной задачей является определение теней, когда источник находится в поле зрения сцены, тогда необходимо делить пространство на сектора и рассчитывать каждый сектор отдельно (рис.4.2.2.).

Выделяют 3 класса алгоритмов построения теней:

- Вычисление затенения в процессе преобразования в растровый вид.
- Разделение поверхности объекта на теневые и нетеневые площади, предшествующие преобразованию в растровый вид.
- Включение значения теней в данные, описывающие объект.

Этот алгоритм основан на принципе отбрасывания лучей и сопряжения с алгоритмами трассировки лучей.

Основная цель алгоритма трассировки лучей – это удалить невидимые поверхности, учесть эффекты отражения и пропускания света. Третья группа алгоритмов использует двухпроходные алгоритмы построчного сканирования, а для второго – Z-буфера.

Один проход выполняется относительно наблюдателя, второй – относительно источника света. Алгоритм построчного сканирования обрабатывает сцену в порядке прохождения сканирующей прямой, в результате чего трёхмерная задача сводится к двухмерной задаче.

Алгоритм Z-буфера – используется дополнительный двумерный массив памяти (Z-буфер) для хранения Z-координаты (глубины каждого пикселя экрана). В процессе сканирования поверхность преобразуется в значение пикселей в кадровом буфере.

Основы анимации. Рендер.

Визуализация - это средство, предназначенное для создания окончательного вида объектов сцены.

К технологиям программной визуализации относятся:

Технология Radiosity (по методу излучательности). Основные идеи Radiosity заимствованы из физики теплового переноса, оперирующие такими понятиями, как поток энергии и плотность потока энергии. Поток энергии – это количество световой энергии, проходящей через некоторую площадь в единицу времени. Плотность потока энергии – это количество энергии переносимой через единицу площади в единицу времени. Radiosity – это поток энергии, исходящий от единицы площади поверхности в единицу времени.

Для получения уравнения Radiosity делается несколько допущений. Поверхности всех объектов сцены разбиваются на плоские участки – патчи. Размер каждого патча должен быть настолько мал, чтобы плотность распределения интенсивности световой энергии его предела считать постоянной величиной.

Считать, что диффузное рассеяние света не зависит от угла, то есть свет рассеивается равномерно по всем направлениям.

Алгоритм расчёта уравнений Radiosity получил название Stochastic Relaxation Radiosity (SRR).

В алгоритме имеется некоторая случайность природы расчётов, в результате чего мелкие детали в сцене могут выпасть. Для исправления этого существует механизм дополнительного сбора излучений (regathering) за счет испуская дополнительных лучей.

Расчет методом Radiosity позволяет точно находить диффузную освещённость сцен, тени от объектов, переноси смешивание отражённого цвета от одних поверхностей на другие. Расчёт не зависит от положения наблюдателя, но затрудняются расчёты отражений и преломлений вблизи зеркальных углов или поверхности. Имеются трудности в расчёте больших открытых сцен. Метод Radiosity не применим для прозрачных поверхностей.

Метод фотонных карт (Mental Ray). Он состоит в двухпроходной технологии расчёта освещённостей поверхности трёхмерных объектов. Первый проход осуществляет прямую трассировку лучей, в результате чего отслеживается их траектория и сохраняется в базе данных, называемой фотонной картой.

На втором проходе выполняется расчёт освещённости пикселей изображения методом обратного стохастического репрессинга с использованием данных фотонных карт. Таким образом, метод Mental Ray – это расширения метода трассировки лучей. Данный метод фотонных карт на сегодняшний день является самым передовым, позволяет воспроизвести на компьютере почти все явления геометрической оптики.

Особенности визуализаторов.

В настройках рендеринга существует несколько вариантов визуализации трёхмерных объектов. Одни используют ресурсы видеокарты, другие задействуют оперативную память. На сегодняшний день доступны следующие визуализаторы:

- Default Scanline Renderer
- Quick Silver
- Mental Ray
- iRay

Default Scanline Renderer позволяет ускорить процесс визуализации за счет отключения таких элементов как проецирование карт, теней, отражений и процедур сглаживания. Позволяет использовать следующие фильтры:

- Плавное изменение цветов
- Усиление резкости
- Кубическое размывание изображений

Позволяет применить глобальную субдискретизацию как дополнительный процесс сглаживания материалов.

Quick Silver относится к аппаратному типу, основным преимуществом является его скорость. Аппаратный визуализатор QS доступен только если видеокарта поддерживает режим работы ShaderModel. Позволяет отключать прозрачность, отражение, тени, параметры освещения.

Визуализатор Mental Ray на сегодняшний день является самым мощным на сегодняшний день. Позволяет рассчитывать отражение и преломление света для различных материалов, устанавливает дополнительные настройки источников света (работает либо со специальными Mental Ray источниками, либо фотометрическими), использовать в сцене специализированные библиотеки материалов.

Устанавливаемые параметры позволяют включать и отключать режим демонстрации приближённого конечного вида визуализируемой сцены моделируемой обработки, изменять качество изображения, изменять чёткость теней, отражений и преломлений, устанавливать параметры трассировки лучей, переопределять материалы, освещения, включая глобальные освещения, параметры каустики (дополнительное отражение, глянец) и так далее.

Визуализатор iRay представляет собой уменьшенный вариант Mental Ray. В нём сокращены основные параметры освещения, наложения текстур. Результат визуализации напрямую зависит от длительности визуализации сцены. Визуализатор применяется для визуализации сцен с обычными и размытыми изображениями, сложными объектами, с параметрами самоосвещения, то есть для получения конечного результат больших сцен.

Создание и вывод изображения в 3DsMAX

В результате применения рендера можно получить готовые изображения с различным форматом вывода. Для анимированных сцен предпочтительнее использовать покадровое сохранение изображений и объединение их в файл с помощью средства 3DsMAX – RAM Player.

Результат визуализации можно сохранить в стандартных форматах файла. Особенностью является то, что результат визуализации мо-

жет включать все объекты сцены, которые не видны в окне проекций. Для печати визуализированных изображений используется специальный мастер размеров печати Print Size Wizard в котором можно выбрать размер бумаги, ориентацию страниц и разрешение. Окончательная печать осуществляется через специальные настройки печати, а не через это окно.

Факторы, влияющие на реалистичность изображения: Многократное переотражения лучей света от поверхности объектов. Моделируется с помощью средств расчёта глобальной освещённости.

Эффект распространения света в материале. Моделируется эффектом подповерхностного рассеяния Sub-Surface Scattering.

Блики, образованные в результате отражения от зеркальных поверхностей или в результате преломления в прозрачных средах (каустика).

Технические особенности в сцене в видео режиме и фотокамер.

Мелкие объекты снимаются макросъёмки, где присутствует эффект глубины резкости (Depth of Field). Используя возможности стандартного визуализатора, можно смоделировать объекты глобальной освещённости и глубины резкости. Для получения других эффектов можно назначать специальные визуализаторы: Final Render (Cebas), Brazil r/s (Splutter Fish), Maxwell Render (Next Limit Technologies), V-Ray (ChaosGroup).

Вопросы для самоконтроля

1. Редактор материалов.
2. Источник света.
3. Установка камер.
4. Основы анимации.
5. Рендер
6. Основы текстурирования, типы и свойства материалов
7. Распределение текстур по объектам
Текстурные развертки, маппинг
8. Основы текстурирования, типы и свойства материалов

Тема 4.3. Системы проектирования информационных моделей строительных объектов

Основные модули

- Возможности программы Autodesk Revit.
- Проектирование
- Возможности

Autodesk Revit – полнофункциональная САПР, предоставляющая возможности архитектурного проектирования, проектирования инженерных систем и строительных конструкций, а также моделирования строительства. Обеспечивает высокую точность выполняемых проектов. Основана на технологии информационного моделирования зданий – BIM. Данная система обеспечивает высокий уровень совместной работы специалистов различных дисциплин и значительно сокращает количество ошибок. Позволяет создавать строительные конструкции и инженерные системы любой сложности. На основе проектируемых моделей специалисты имеют возможность выработать эффективную технологию строительства и точно определить требуемое количество материалов (рис.4.3.1.).

Возможности и проектирование.

Технология BIM. Пользователь ничего не чертит. Вместо этого он занимается моделированием и оформлением чертежей. При этом процессы моделирования и формирования чертежей разделены. Процедуры же черчения используются крайне редко.

Параметрическое моделирование. Все связи между объектами и элементами задаются с помощью параметров, которые можно динамически менять.

Системы. Поддерживается построение систем различного направления – ОВК, трубопроводные, электрические системы – с соответствующими параметрами и расчётами.

Работа с системами:

- создание пользовательских типоразмеров воздуховодов и трубопроводов;
- доступ к свойствам графических переопределений (цвет, вес линий и образец штриховки);
- назначение неподключенных объектов системам;
- создание пояснений к геометрии с учётом аббревиатуры систем;
- возможность ограничения или отключения расчётов систем.

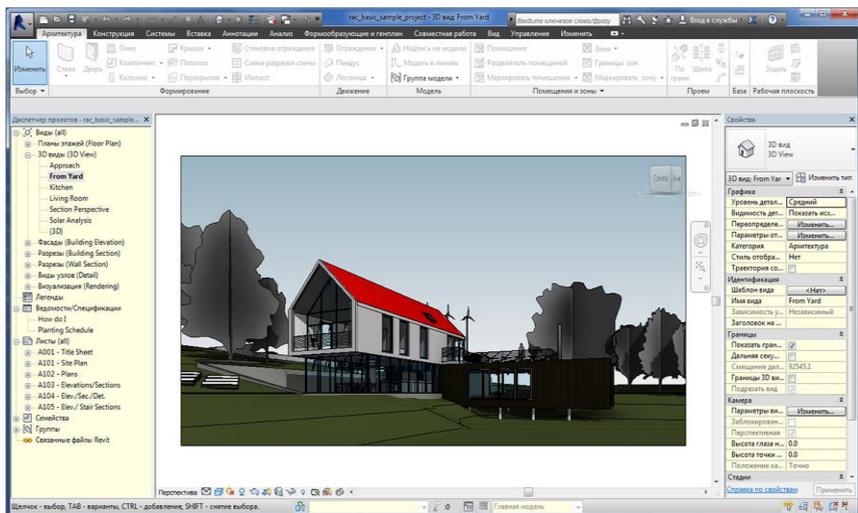


Рис.4.3.1. Autodesk Revit

Варианты. Возможность создавать разные варианты одних и тех же зданий, конструкций и систем.

Стадии. Поддержка различных стадий строительства, в том числе снос старых построек, систем или конструкций.

Лестницы и перила. Отдельные инструменты моделирования, редактирования и выпуска документации по лестницам и перилам. При их реализации использованы многие предложения сообщества пользователей.

Модели систем соответствуют их реальным аналогам.

Управление углами. Углы расположения воздуховодов, труб, кабельных каналов и лотков можно ограничивать в соответствии со стандартами. При этом трубы располагаются строго в соответствии с заданными углами, а для воздуховодов, труб, кабельных каналов и лотков можно задавать либо непосредственно углы, либо их приращение.

Пробки. Быстрое и удобное расположения пробок на открытых концах труб и воздуховодов. В настройках можно определить режим автоматической установки пробок при автоматической трассировке.

Шаблоны санитарно-технических систем. Эти шаблоны могут применяться в проектах ливневой канализации, хозяйственно-бытовой канализации и прочих, что значительно повышает эффективность проектирования.

Зависимости размещения арматуры. Настройка зависимостей при размещении арматуры позволяет обновлять её расположение в соответствии с расчётами, улучшая точность и скоординированность проектов. Благодаря этому совершенствуются текущие правила раскладки арматуры, а также пользователи получают больше возможностей для управления расположением арматуры в железобетонных изделиях.

Моделирование элементов армирования железобетонных конструкций и формирования по ним документации. Имеются в наличии опция округления длин элементов и коды форм по европейским стандартам. Существует инструмент моделирования и отображения проводочных сеток.

Создание каталогов параметрических компонентов с помощью внешних CSV-файлов. Можно внедрять данные из CSV-файлов в семейства. Встроенный в редактор семейств инструмент *Lookup Table Manage* позволяет управлять встроенными данными.

Шаблоны временных видов. С их помощью свойства вида могут быть изменены без воздействия на его сохранённое состояние.

Параметрические компоненты (семейства). Основа всех процессов моделирования и проектирования. Для работы с параметрическими компонентами не требуется знаний языков программирования.

Двунаправленная ассоциативность. Данный механизм обеспечивает централизованное хранение всей информации о проекте. Внесение изменения возможно с любого из видов. Внесённые изменения отображаются на всех видах.

Инструменты концептуального проектирования. Создание и свободное манипулирование эскизами и моделями произвольной формы. Возможность создавать формы и геометрию в виде реальных компонентов здания, что упрощает переход к разработке проекта и выпуску документации. При этом, Revit в ходе работы автоматически создаёт параметрический каркас для форм, обеспечивая высокий уровень точности и гибкости.

Облака точек. Опции визуализации предоставляют качественное графическое представление облаков точек. У пользователей имеются большие возможности для управления ими. Качество изображений и производительность работы с ними регулируются через значения плотности пикселей и размера точки.

Вариация параметров для групп. Имеется возможность варьировать назначенные группам значения параметров. Это делает процесс формирования отчётов гибким и позволяет в значительной мере учитывать требования формата COBie. Доступность проектных данных

при техническом обслуживании и на стадии эксплуатации повышает эффективность жизненного цикла зданий.

Цветовые схемы на планах систем. Цветовые схемы дают возможность наглядно демонстрировать проектные идеи. Любые изменения планов с цветовыми обозначениями автоматически передаются в модель. Количество схем цветовых обозначений неограниченно и может использоваться на протяжении всего процесса проектирования. Трёхмерное моделирование позволяет создавать системы, в которых с помощью цветовых схем проиллюстрированы различные технологические и электрические параметры (расчётный поток, фактический поток, зонирование, силовая нагрузка, освещённость и тому подобное).

Компоненты трубопроводных и электрических систем. В Revit можно использовать компоненты различных систем. Например, систем связи, передачи данных, пожарной безопасности и так далее. Также предусмотрены инструменты моделирования кабельных лотков и каналов. Индивидуальные управляющие устройства и панели управления дают возможность подключать различное оборудование.

Браузер систем. Позволяет работать с системами отдельно. Можно выбрать на виде только ту систему, с которой нужно работать в текущий момент. Можно получить информацию об общей электрической нагрузке, воздушных потоках и текучих средах в любой точке системы. Как следствие – можно контролировать связность элементов любой системы и их соответствие требованиям по нагрузке, что важно для правильного подбора размеров.

Наклонные трубы. Возможность создавать наклонные трубы. Возможность выполнять данную функцию сразу для нескольких труб. Автоматический расчёт соответствующих параметров, в том числе вычисление уровней низа. Автоматическая маркировка концов участков трубопровода.

Материалы при моделировании. Возможность использовать при проектировании строительных конструкций свойства различных строительных материалов: кирпича, монолитного и сборного железобетона, дерева, стали и так далее. Возможность комбинировать материалы.

Мощный API. Для улучшения качества визуализации проекты передаются в 3dsMax, Showcase, сервис Buzzsaw и другие. API функций совместной работы предоставляет мощные средства взаимодействия Revit с системами управления данными. API функций работы с видами позволяет усовершенствовать процессы, основанные на взаимодействии нескольких САПР. Расширенный API функций генплана предоставляет возможность проектировать строительные площадки.

Пользователи могут обращаться к таким функциям, как формирование закреплённых окон или проверка соответствия СНиПам.

Документация

Непрямоугольная область подрезки. Возможность создавать области подрезки произвольной формы. Как следствие, повышается гибкость формирования документации. Экономятся ресурсы проектирования за счёт отказа от трудоёмких обходных процедур.

Частичные фасады. Данный инструмент полезен в случае расположения на строительной площадке нескольких загромождающих друг друга зданий и позволяет оставить на фасаде изображение только необходимого здания.

Альтернативные размеры. Данный инструмент позволяет отображать рядом друг с другом размеры в разных метрических системах. Например, в британских и в метрических.

Маркировка элементов арматуры. Данная функция обеспечивает мощную поддержку выпуска документации по железобетонным конструкциям. Она позволяет назначать единую марку нескольким элементам или субэлементам, что полезно для располагаемой в одну линию арматуры.

Инструмент размещения балок и раскосов. Инструмент детализации стальных конструкций содержит опции, позволяющие описывать геометрию элементов и размещать с высокой точностью балки и раскосы. Проектная модель может применяться в качестве основы для последующей детализации.

Автоматизированные спецификации. Revit обеспечивает высокий уровень контроля при формировании спецификаций, а также доступ к данным из информационной модели здания. Обеспечивается гибкое управление внешним видом спецификаций, что позволяет формировать их в соответствии с ГОСТами и СНиПами. Возможность добавлять графические элементы. Методы навигации сходны с методами навигации в электронных таблицах.

Моделирование строительства. Данный инструмент позволяет точно отразить методы строительства, что достигается благодаря возможности разбиения объектов с последующей манипуляцией их элементами. Можно создавать рабочие чертежи для изготовления строительных изделий.

Взаимодействие. Revit обеспечивает высокую степень совместной работы специалистов различных дисциплин. Имеется возможность импортировать модели из Inventor или экспортировать модели площадок или здания в AutoCAD Civil 3D.

Ведомости материалов. Данный инструмент позволяет определять количество необходимых материалов и их стоимость, что обеспечивается за счёт параметризации изменений.

Поддержка форматов DWG, DWF, DXF и DGN.

Маркировка по месту и по категории. Возможность маркировать компоненты на этапе их размещения. Средства маркировки доступны для большинства объектов Revit.

Взаимодействие с внешними базами данных. Revit располагает инструментарием для вывода информации о моделях в любую ODBC-совместимую базу данных, что позволяет передавать данные в приложения, предназначенные для расчёта стоимости, планирования, составления заказов и управления эксплуатацией.

Спецификация панелей. Возможность составления спецификаций панелей с отображением показателей нагрузки. Многополюсные слоты могут объединяться в одну ячейку. Классификация нагрузок и их показателей может отображаться для каждой цепи.

Чертежи узлов. Инструментарий двумерного черчения позволяет на видах трёхмерной модели изображать детали узлов и добавлять к ним необходимые аннотации. Можно импортировать набор чертежей в формате DWG из других проектов. Специализированные средства черчения оптимизированы под создание таких элементов стальных и железобетонных конструкций, как сварные соединения, анкерные болты, арматурные стержни, зоны армирования.

Совместная работа

С разрабатываемой моделью могут работать несколько проектировщиков, сохраняя результаты своей работы в едином файле.

Проверка на наличие коллизий.

Revit Server. Средство, позволяющее через глобальную сеть наладить совместную работу над моделями Revit для распределённых проектных коллективов. На центральном сервере хранятся объединённые модели Revit. Все специалисты проектной группы имеют доступ к ним посредством локальных серверов. В случае отсутствия соединения по глобальной сети задействуются средства встроенного резервирования, помогающие защитить результаты разработок.

Интеграция с Autodesk Vault. Расширяет возможности управления данными, особенно при межотраслевом проектировании. Позволяет отслеживать и редактировать данные на любом этапе разработки – от планирования до строительства.

Поддержка Citrix (Citrix Ready) и 64-разрядных операционных систем. Revit совместим с Citrix XenApp6. Встроенная поддержка

64-разрядных операционных систем позволяет в полной мере использовать возможности компьютера, что особенно важно в таких задачах, как визуализация, обновлением модели и других.

Интеграция с Autodesk 360. Подписчики Building Design Suite получают доступ к интегрированным ресурсам Autodesk, в том числе к возможностям визуализации и расчётам энергопотребления. Отправка в облако проектов на проведение расчётов и визуализацию ускоряет процесс получения высококачественных результатов. При этом в ресурсы компьютера освобождаются для решения других задач.

Визуализация

Визуализация проектов. Высокое качество визуализации обеспечивается системой рендеринга mental ray, обладающей удобным интерфейсом и поддерживающей высокую скорость работы.

Облачный рендеринг. Данный инструмент позволяет разработчикам экономить время и бюджет. Практически фотореалистичная графика формируется без привлечения специализированного оборудования.

Виды узлов. Пользователи могут настраивать виды в соответствии со своими предпочтениями. Масштабирование видов узлов помогает лучше понимать их структуру и последовательность формирования при строительстве.

Расчёты и анализ

Расчёты энергопотребления для элементов здания. Для расчёта энергопотребления на базе модели здания создаётся аналитическая модель и отправляется в Green Building Studio, где анализируется с помощью DOE2.2. После анализа результатов расчёта можно внести необходимые изменения в модель здания для оптимизации энергопотребления.

Аналитическая модель конструкций. Данный инструмент позволяет наглядно представить связи между элементами конструкций. Имеется возможность заносить свойства аналитических и физических элементов в одну и ту же спецификацию и снабжаться ярлыками аналитические связи и узлы.

Расчёт воздуховодов и трубопроводов через API. С целью учёта отраслевых норм пользователи имеют возможность создавать свои расчётные надстройки вместо используемых по умолчанию.

Расчёт энергопотребления для Revit. Веб-сервис Green Building Studio предоставляет технологии облачных вычислений, с помощью которых можно сравнить значения энергопотребления для разных вариантов проекта. Результаты расчёта могут выводиться в наглядном графическом формате.

Расчёты строительных конструкций в Autodesk 360. Использование облачного расчёта конструкций зданий позволяет уже на ранних этапах проектирования принимать более обоснованные решения. Расчёт доступен для нескольких вариантов модели.

Двусторонняя связь с различными расчётными программами. Обеспечена двухсторонняя связь аналитической модели строительных конструкций с Robot Structural Analysis Professional. Такая связь позволяет модели обновляться с учётом полученных результатов. Параметрическое управление изменениями обеспечивается согласованность всех разделов проекта и документации. Кроме того, имеется возможность передавать аналитическую информацию в программы автоматизированного проектирования сторонних разработчиков. Совместно с внешними программами можно использовать данные опор и граничных условий, данные нагрузок и их комбинаций, данные свойств материалов и профилей.

Применение свойств физических материалов при анализе эксплуатационных характеристик. Данный механизм обеспечивает определение эксплуатационных характеристик зданий по технологии BIM. При этом оболочке зданий назначаются структурные и термические свойства. Рассчитанные в Revit тепловые параметры можно экспортировать в формате gbXML для отправки в специализированные приложения для более детального расчёта.

Интерфейс

Autodesk Exchange. Этот механизм позволяет пользователям Revit приобретать новые приложения через сеть Интернет и устанавливать их, не прерывая текущего сеанса. Так же он позволяет пользоваться ресурсами обширной библиотеки.

Гибкое управление диалоговыми окнами. Возможность объединения их в одно окно. Новое окно представляется в виде ярлыка вкладки в нижней части диалогового окна. Вкладки отдельных окон можно перетаскивать за пределы окна. При этом создается новое окно.

Основные типы файлов Revit

- Файлы проектов *.rvt.
- Файлы семейств *.rfa.
- Файлы шаблонов проектов *.rte.
- Файлы шаблонов семейств *.rft.

Экспорт данных

Revit поддерживает следующие виды экспорта:

- экспорт проектов и семейств в формат DWG;
- экспорт проектов и семейств в формат DXF;

- экспорт проектов и семейств в формат DGN;
- экспорт проектов и семейств в формат ACIS SAT;
- экспорт проектов и семейств в формат DWF/DWFX;
- экспорт проектов в формат ADSK;
- экспорт типоразмеров семейства в текстовый файл *.txt;
- экспорт проектов в формат IFC;
- экспорт проектов и формообразующих элементов в формат gbXML;
- сохранение данных модели в базе данных ODBC;
- запись файлов анимации с помощью свободной камеры или на основе расчёта инсоляции;
- сохранение изображений в форматах *.jpg, *.tif, *.bmp, *.tga и *.png;
- сохранение спецификации в html-файле;
- сохранение расчёта площадей помещений или зон в html-файле;
- экспорт параметров DWG/DXF;
- экспорт параметров DGN;
- экспорт параметров IFC.

Импорт данных

Revit поддерживает следующие виды импорта:

- импорт из формата DWG;
- импорт из формата DXF;
- импорт из формата DGN;
- импорт из формата ACIS SAT;
- импорт из формата SketchUp;
- импорт из формата DWF/DWFX;
- вставка изображений, хранящихся в файлах форматов *.jpg, *.jpeg, *.tif, *.bmp, и *.png;
- импорт из формата ADSK;
- импорт из формата gbXML.

Импорт данных доступен в двух видах – в виде ссылки и в виде импорта. В первом случае в файл Revit'a прописываются ссылки на внешний файл. Во втором случае импортируемые данные полностью записываются в файл проекта или семейства.

Состав и программные комплексы

Revit предоставляется в одном, из следующих видов:

- **Revit Architecture** – САПР для архитектурного проектирования;
- **Revit MEP** – САПР для проектирования инженерных систем;

- **Revit Structure** – САПР для проектирования строительных конструкций;

Кроме того, Revit полностью или частично включён в состав следующих программных комплексов Autodesk:

- **Building Design Suit** версий Premium и Ultimate (включён полностью);

- **Infrastructure Design Suit** версий Premium и Ultimate (**Revit Structure**);

- **Plant Design Suit** версий Premium и Ultimate (**Revit Structure**).

Вопросы для самоконтроля

1. Системы проектирования информационных моделей строительных объектов.
2. О программе Autodesk Revit.
3. Возможности программе Autodesk Revit

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Discovering Computers 2016. Tools, Apps, Devices, and the Impact of Technology. 691 pg.
2. Richard L. Halterman Fundamentals of C++ Programming. Copyright © 2008–2016. All rights reserved. 634 pg.
3. Brian P. Hogan HTML5 and CSS3, Second Edition. Level Up with Today's Web Technologies. Copyright © 2013 The Pragmatic Programmers, LLC. All rights reserved. 290 pg.
4. Raavi O'Connor Autodesk 3ds Max® 2016 Modeling and Shading Essentials. Copyright © 2015 Raavi Design. 466 pg.

Дополнительная литература

5. Mirziyoev SH.M. Tashkiliy tahlil, qa'tiy tartib-intizom va shaxsiy javobgarlik-xar bir raxbar faoliyatining kundalik qoidasi bo'lishi kerak. T., "O'zbekiston". 2017 y. 102 bet.
6. Mirziyoev SH.M. Qonun ustuvorligi va inson manfaatlarini ta'minlash – yurt taraqqiyoti va xalq faravonligining garofi. T., "O'zbekiston". 2016 y. 47 bet.
7. Mirziyoev SH.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. T., "O'zbekiston". 2016 y. 48 bet.
8. Randy H. Shih AutoCAD 2017 Tutorial - First Level: 2D Fundamentals Better Textbooks. Lower Prices.
9. BarBaraZukinHeiman. PH.D. and others Practical Photoshop® CS6, Level 1 Copyright © 2009–2017 by. 53 pg.
10. М.Арипов. Информационные технологии. Учебное пособие Т.: "Noshir" 2009. 366-с.
11. М.М.Арипов. Информатика, Информационные технологии. Учебник Т.: TDYUI 2005. 278-с.
12. С.С. Косисов. Информационные технологии: Учебник для высших учебных заведений. Т.: Alokachi, 2006.-360b.

Internet сайты

13. www.uz –Национальная поисковая система
14. www.gov.uz –Правительственный портал Республики Узбекистан
15. www.ZiyoNET.uz- Информационно-образовательный портал Республики Узбекистан
16. www.e-darslik.uz
17. <http://www.vse.uz/> -Энциклопедия поисковых систем

О.С. Абдуллаева

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СТРОИТЕЛЬСТВЕ
(ЧАСТЬ 2)**

Учебник

Подписано в печать 16.09.2022.
Формат 60×90/16. Усл. печ. л. 12,0.
Тираж 1000 экз.

ООО «Русайнс».
117218, г. Москва, ул. Кедрова, д. 14, корп. 2.
Тел.: +7 (495) 741-46-28.
E-mail: autor@ru-science.com
<http://ru-science.com>