### *Тема 2.1. Программное обеспечение компьютера*

**[2.1.1. Определения и классификация](#_Тема_2.1._Программное)**

**[2.1.2. Операционные системы](#_2.1.2._Операционные_системы)**

**[2.1.3. Системы программирования](H:\\УчебныеМатериалы\\ДокументыСпециальностей\\Спец210700\\Эл-ПОСОБИЕ-ПРАКТИКУМ-ЗАДАЧНИК\\Модуль-1-210700\\Дисциплина Информатика-210700\\Раздел-02-Информатика-210700\\Тема-02-03.docx)**

**[2.1.4. Инструментальные программы и пакеты прикладных программ](#_2.1.4._Инструментальные_программы)**

**[2.1.5. Контрольные вопросы по теме «Программное обеспечение компьютера»](#_2.1.5._Контрольные_вопросы)**

**[2.1.6. Тестовые задания по теме «Программное обеспечение компьютера»](#_2.1.6._Тестовые_задания)**

### 2.1.1. Определения и классификация

**Программное обеспечение (ПО)** является логическим продолжением технических средств. Сфера применения конкретного компьютера определяется созданным для него **ПО**.

Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых на компьютерах программах. К **ПО** можно отнести также всю **область деятельности по проектированию и разработке ПО:**

* технология проектирования программ (например, нисходящее проектирование, структурное и объектно-ориентированное проектирование и др.);
* методы [тестирования программ](http://www.tomsk.ru/Books/informatica/theory/chapter8/1_8_7.html);
* методы доказательства правильности программ;
* анализ качества работы программ;
* документирование программ;
* разработка и использование программных средств, облегчающих процесс проектирования программного обеспечения, и многое другое.

Все программы, работающие на компьютере, можно условно разделить на **три категории**:

* [***прикладные программы***](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_3.html)***,*** непосредственно обеспечивающие выполнение необходимых пользователям работ;
* [***системные программы***](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_4.html)***,*** выполняющие различные вспомогательные функции, например: управление ресурсами компьютера; создание копий используемой информации; проверка работоспособности устройств компьютера; выдача справочной информации о компьютере и др.;
* [**инструментальные программные системы**](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_12.html)**,** облегчающие процесс создания новых программ для компьютера.

Однако в настоящее время развитие ПО пошло как вглубь (появились новые подходы к построению операционных систем, языков программирования и т.д.), так и вширь (прикладные программы перестали быть прикладными и приобрели самостоятельную ценность).

Соотношение между требующимися программными продуктами и имеющимися на рынке меняется очень быстро. Даже классические программные продукты, такие, как операционные системы, непрерывно развиваются и наделяются интеллектуальными функциями, многие из которых ранее относились только к интеллектуальным возможностям человека.

Кроме того, появились нетрадиционные программы, классифицировать которые по устоявшимся критериям очень трудно, а то и просто невозможно.

На сегодняшний день можно сказать, что более или менее определенно сложились следующие группы программного обеспечения:

* [операционные системы](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_5.html) и [оболочки](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_8.html);
* [системы программирования](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_11.html)(трансляторы, библиотеки подпрограмм, отладчики и т.д.);
* инструментальные системы;
* [интегрированные пакеты программ](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_20.html);
* системы машинной графики;
* системы управления базами данных ([**СУБД**](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_17.html#DBMS)**)**;
* прикладное программное обеспечение.

**Прикладная программа** – это любая конкретная программа, способствующая решению какой-либо задачи в пределах данной предметной области. Например, там, где на компьютер возложена задача управления за финансовой деятельностью какой-либо фирмы, прикладной будет программа подготовки платежных ведомостей. Прикладные программы могут носить и общий характер, например, обеспечивать составление и печатание документов и т.п. Причем прикладные программы могут использоваться либо автономно, то есть решать поставленную задачу без помощи других программ, либо в составе программных комплексов или пакетов.

**Системные программы** выполняются вместе с прикладными и служат для управления ресурсами компьютера – центральным процессором, памятью, вводом-выводом. Это программы общего пользования, которые предназначены для всех пользователей компьютера. Системное программное обеспечение разрабатывается так, чтобы компьютер мог эффективно выполнять прикладные программы.

Среди десятков тысяч системных программ особое место занимают **операционные системы (ОС)**, которые обеспечивают управление ресурсами компьютера с целью их эффективного использования.

Важными классами системных программ являются также программы вспомогательного назначения – **утилиты**. Они либо расширяют и дополняют соответствующие возможности **ОС**, либо решают самостоятельные важные задачи. Причем часть утилит входит в состав операционной системы, а другая часть функционирует независимо от нее, т.е. автономно.

**Операционная система** – это комплекс взаимосвязанных системных программ, назначение которого – организовать взаимодействие пользователя с компьютером и выполнение всех других программ. Операционная система – связующее звено между аппаратурой компьютера, с одной стороны, и выполняемыми программами, а также пользователем, с другой стороны.

Операционная система обычно хранится во внешней памяти компьютера –на диске. При включении компьютера она считывается с дисковой памяти и размещается в **ОЗУ**. Этот процесс называется **загрузкой операционной системы.**

В основные функции операционной системы входит:

* осуществление диалога с пользователем;
* ввод-вывод и управление данными;
* планирование и организация процесса обработки программ;
* распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);
* запуск программ на выполнение;
* всевозможные вспомогательные операции обслуживания;
* передача информации между различными внутренними устройствами;
* программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

**Операционную систему** можно назвать программным продолжением устройства управления компьютера. Операционная система скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате этого люди освобождаются от очень трудоёмкой работы по **организации взаимодействия с аппаратурой компьютера.**

В различных моделях компьютеров используют операционные системы с разной архитектурой и возможностями. Для их работы требуются разные ресурсы. Они предоставляют разную степень сервиса для программирования и работы с готовыми программами. Однако в любой **ОС** имеются средства для организации хранения **файлов** на каком-либо носителе – **файловая система**.

**Файл** – это поименованная совокупность любых данных, размещенная на внешнем запоминающем устройстве и хранимая, пересылаемая и обрабатываемая как единое целое. Файл может содержать программу, числовые данные, текст, закодированное изображение и др.Физически файлы реализуются как участки памяти на [внешних носителях](http://www.tomsk.ru/Books/informatica/theory/chapter2/1_2_10.html), например, на магнитных дисках или **CD-ROM**. Каждый файл занимает некоторое количество блоков дисковой памяти. Обслуживает файлы специальный модуль операционной системы, называемый **драйвером файловой системы***.* Каждый файл имеет имя, зарегистрированное в **каталоге** - оглавлении файлов.

**Каталог (директорию или папку)** можно просматривать, переименовывать зарегистрированные в нем файлы, переносить их содержимое на новое место и удалять. Каталог может иметь собственное имя и храниться в другом каталоге наряду с обычными файлами: так образуются иерархические файловые структуры.

**Драйвер файловой системы** обеспечивает доступ к информации, записанной на магнитный носитель, по имени файла и распределяет пространство на магнитном носителе между файлами.

Для выполнения этих функций драйвер файловой системы хранит не только информацию пользователя, но и свою собственную служебную информацию. В служебных областях носителя хранится список всех файлов и каталогов, а также различные дополнительные справочные таблицы, служащие для повышения скорости работы драйвера файловой системы.

К файловой системе имеет доступ также и любая прикладная программа, для чего во всех языках программирования имеются специальные процедуры.

Понятие файла может быть обращено на любой источник или потребитель информации в компьютере, например, в качестве файла для программы могут выступать принтер, дисплей, клавиатура и др.

Структура файловой системы и структура хранения данных на внешних магнитных носителях определяет удобство работы пользователя, скорость доступа к файлам и т.д.

**Оболочки** – это программы, созданные для упрощения работы со сложными программными системами, такими, например, как **ОСDOS**. Они преобразуют неудобный командный пользовательский интерфейс в дружественный графический интерфейс или интерфейс типа «меню». Прежде всего, оболочки предоставляют пользователю удобный доступ к файлам и обширные сервисные услуги.

Одна из самых популярных у пользователей IBM-совместимого **ПК** оболочка – это пакет программ **Norton Commander**. Он обеспечивает:

* создание, копирование, пересылку, переименование, удаление, поиск файлов, а также изменение их атрибутов;
* отображение дерева [каталогов](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_6.html#DIRECTORY) и характеристик входящих в них файлов в форме, удобной для восприятия человека;
* создание, обновление и распаковку архивов (групп сжатых файлов);
* просмотр текстовых файлов;
* редактирование текстовых файлов;
* выполнение из её среды практически всех команд **DOS;**
* запуск программ;
* выдачу информации о ресурсах компьютера;
* создание и удаление каталогов;
* поддержку межкомпьютерной связи;
* поддержку [электронной почты](http://www.tomsk.ru/Books/informatica/theory/chapter2/1_2_27.html#EMAIL) через [модем](http://www.tomsk.ru/Books/informatica/theory/chapter2/1_2_16.html).

В начале 90-х годов во всем мире огромную популярность приобрела графическая оболочка **ОС DOS** - **MS Windows**, преимущество которой состоит в том, что она облегчает использование компьютера, и её графический интерфейс вместо набора сложных команд с клавиатуры позволяет выбирать их мышью из меню практически мгновенно. Операционная среда **Windows**, работающая совместно с операционной системой **DOS**, реализует все свойства, необходимые для производительной работы пользователя, в том числе –многозадачный режим.

### 2.1.2. Операционные системы

В настоящее время большинство компьютеров в мире работают под управлением той или иной версии операционной среды **MS Windows.**

**Windows 95** представляет собой универсальную высокопроизводительную многозадачную 32-разрядную **ОС** нового поколения с графическим интерфейсом и расширенными сетевыми возможностями. **Windows 95** — интегрированная среда, обеспечивающая эффективный обмен информацией между отдельными программами и предоставляющая пользователю широкие возможности работы с мультимедиа, обработки текстовой, графической, звуковой и видеоинформации. **Интегрированность** подразумевает также совместное использование ресурсов компьютера всеми программами.

Эта операционная система обеспечивает работу пользователя в сети, предоставляя встроенные средства поддержки для обмена файлами и меры по их защите, возможность совместного использования принтеров, факсов и других общих ресурсов. **Windows 95** позволяет отправлять сообщения электронной почтой, факсимильной связью, поддерживает удаленный доступ. Применяемый в **Windows 95** защищённый режим не позволяет прикладной программе в случае сбоя нарушить работоспособность системы, надежно предохраняет приложения от случайного вмешательства одного процесса в другой, обеспечивает определённую устойчивость к вирусам.

Пользовательский интерфейс **Windows 95** прост и удобен. В отличие от оболочки **Windows 3** эта операционная система не нуждается в установке на компьютере операционной системы **DOS**. Она предназначена для установки на настольных ПК и компьютерах блокнотного типа с процессором **486** или **Pentium***.* Рекомендуемый размер оперативной памяти 32–128 Мбайт. Пользовательский интерфейс спроектирован так, чтобы максимально облегчить усвоение этой операционной системы новичками и создать комфортные условия для пользователя.

**Windows 98** отличается от **Windows 95** тем, что в ней операционная система объединена с браузером **Internet Explorer** посредством интерфейса, выполненного в виде **Web-браузера** и оснащенного кнопками «Назад» и «Вперед» для перехода на предыдущую и последующую **Web-страницы**. Кроме этого, в ней улучшена совместимость с новыми аппаратными средствами компьютера, она одинаково удобна как для использования на настольных, так и на портативных компьютерах.

**Windows NT** – это операционная система, а не просто графическая оболочка. Она использует все возможности новейших моделей персональных компьютеров и работает без **DOS***.***Windows NT**– 32-разрядная **ОС** со встроенной сетевой поддержкой и развитыми многопользовательскими средствами. Она предоставляет пользователям истинную многозадачность, многопроцессорную поддержку, секретность, защиту данных и многое другое. Эта операционная система очень удобна для пользователей, работающих в рамках локальной сети, для коллективных пользователей, особенно для групп, работающих над большими проектами и обменивающихся данными.

**Windows 2000 Professional**– операционная система нового поколения для делового использования на самых разнообразных компьютерах –от портативных до серверов. Эта **ОС** является наилучшей для ведения коммерческой деятельности в Интернете.

Дальнейшим развитием **MS Windows** стала **Windows XP,** которая позиционируется как универсальная **ОС**– для домашнего и корпоративного использования. Она выпускается в двух модификаций: **Windows XP Professional** и **Windows XP Home Edition**

Наряду с **MS Windows** в настоящее время широко используется сетевая **ОС Unix,** которая была создана в **Bell Telephone Laboratories**. **ОС Unix**–многозадачная операционная система, способная обеспечить одновременную работу очень большого количества пользователей. Ядро **ОС Unix** написано в основном на языке высокого уровня **C**. Это позволяет за считанные месяцы переносить **ОС Unix** на другие аппаратные платформы и достаточно легко вносить в нее серьезные изменения и дополнения. **UNIX** является первой действительно переносимой операционной системой.

### 2.1.3. Системы программирования

**Система программирования**– это система для разработки новых программ на конкретном языке программирования. Современные системы программирования обычно предоставляют пользователям мощные и удобные средства разработки программ**.** В них входят:

* [компилятор](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_10.html#COMPILER) или [интерпретатор](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_10.html#INTERPRETER);
* интегрированная среда разработки;
* средства создания и редактирования текстов программ;
* обширные [библиотеки стандартных программ](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_18.html) и функций;
* [отладочные программы](http://www.tomsk.ru/Books/informatica/theory/chapter8/1_8_6.html), т.е. программы, помогающие находить и устранять ошибки в программе;
* «дружественная» к пользователю диалоговая среда;
* многооконный режим работы;
* мощные графические библиотеки; [утилиты](http://www.tomsk.ru/Books/informatica/theory/chapter6/1_6_4.html#UTILITIES) для работы с библиотеками
* встроенный [ассемблер](http://www.tomsk.ru/Books/informatica/theory/chapter7/1_7_15.html);
* встроенная справочная служба;
* другие специфические особенности.

**Транслятор** – это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трансляторы реализуются в виде компиляторов или интерпретаторов. С точки зрения выполнения работы компилятор и интерпретатор существенно различаются.

**Компилятор** читает всю программу целико*м*, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

**Интерпретатор** переводит и выполняет программу строка за строкой.

После того, как программа откомпилирована, уже ни сама исходная программа, ни компилятор становятся не нужны. В то же время программа, обрабатываемая интерпретатором, должна заново *переводиться* на машинный язык при каждом очередном запуске программы. Откомпилированные программы работают быстрее, но интерпретируемые проще отлаживать и изменять.

Каждый конкретный язык ориентирован либо на компиляцию, либо на интерпретацию - в зависимости от того, для каких целей он создавался. Например, **Pascal** обычно используется для решения довольно сложных задач, в которых важна скорость работы программ. Поэтому данный язык обычно реализуется с помощью компилятора.

С другой стороны, **Basic** создавался как язык для начинающих программистов, для которых построчное выполнение программы имеет неоспоримые преимущества. Иногда для одного языка имеется и компилятор, и интерпретатор. В этом случае для разработки и тестирования программы можно воспользоваться интерпретатором, а затем откомпилировать отлаженную программу, чтобы повысить скорость ее выполнения.

Одним из важнейших признаков классификации языков программирования является принадлежность их к одному из стилей, основными из которых являются следующие: процедурный, функциональный, логический и объектно-ориентированный.

**Процедурное (императивное) программирование** является отражением архитектуры традиционных ЭВМ, которая была предложена фон Нейманом в 40-х годах. Теоретической моделью процедурного программирования служит алгоритмическая система под названием «машина Тьюринга».

Программа на процедурном языке программирования состоит из последовательности операторов (инструкций), задающих процедуру решения задачи. Основным является оператор присваивания, служащий для изменения содержимого областей памяти. Концепция памяти как хранилища значений, содержимое которого может обновляться операторами программы, является фундаментальной в императивном программировании.

Выполнение программы сводится к последовательному выполнению операторов с целью преобразования исходного состояния памяти, то есть значений исходных данных, в заключительное, то есть в результаты. Таким образом, с точки зрения программиста имеются программа и память, причем первая последовательно обновляет содержимое последней.

Процедурные языки характеризуются следующими особенностями:

* необходимостью явного управления памятью, в частности, описанием переменных;
* малой пригодностью для символьных вычислений;
* отсутствием строгой математической основы;
* высокой эффективностью реализации на традиционных ЭВМ.

Одним из важнейших классификационных признаков процедурного языка является его уровень. Уровень языка программирования определяется семантической (смысловой) емкостью его конструкций и степенью его ориентации на программиста. Язык программирования частично ликвидирует разрыв между методами решения различного рода задач человеком и вычислительной машиной. Чем более язык ориентирован на человека, тем выше его уровень. Дадим краткую характеристику реализованным на ПЭВМ языкам программирования в порядке возрастания их уровня.

***Двоичный язык*** является непосредственно машинным языком. В настоящее время такие языки программистами практически не применяются.

***Язык Ассемблера*** — это язык, предназначенный для представления в удобочитаемой символической форме программ, записанных на машинном языке. Он позволяет программисту пользоваться мнемоническими кодами операций, присваивать удобные имена ячейкам и областям памяти, а также задавать наиболее удобные схемы адресации.

***Язык Макроассемблера*** является расширением языка Ассемблера путем включения в него макросредств. С их помощью в программе можно описывать последовательности инструкций с параметрами — макроопределения. После этого программист может использовать снабженные аргументами макрокоманды, которые в процессе ассемблирования программы автоматически замещаются макрорасширениями. Макрорасширение представляет собой макроопределение с подставленными вместо параметров аргументами.

Другими словами, язык Макроассемблера предоставляет средства определения и использования новых, более мощных команд как последовательности базовых инструкций, что несколько повышает его уровень.

Языки Ассемблера и Макроассемблера применяются системными программистами-профессионалами с целью использования всех возможностей оборудования ЭВМ и получения эффективной по времени выполнения и по требуемому объему памяти программы. На этих языках обычно разрабатываются относительно небольшие программы, входящие в состав системного программного обеспечения: драйверы, утилиты и другие.

***Язык программирования С (Си)***первоначально был разработан для реализации операционной системы UNIX в начале 70-х годов. В последующем приобрел высокую популярность среди системных и прикладных программистов. В настоящее время этот язык реализован на большинстве ЭВМ.

В**С** сочетаются достоинства современных высокоуровневых языков в части управляющих конструкций и структур данных с возможностями доступа к аппаратным средствам ЭВМ на уровне, который обычно ассоциируется с языком низкого уровня типа языка Ассемблера. Язык **С** имеет синтаксис, обеспечивающий краткость программы, а компиляторы способны генерировать эффективный объектный код.

Одна из наиболее существенных особенностей **С** состоит в нивелировании различий между выражениями и операторами, что приближает его к функциональным языкам. В частности, выражение может обладать побочным эффектом присваивания, а также может использоваться в качестве оператора. Нет также четкой границы между процедурами и функциями, более того, понятие процедуры не вводится вообще.

Синтаксис языка затрудняет программирование и восприятие составленных программ. Отсутствует и строгая типизация данных, что предоставляет дополнительные возможности программисту, но не способствует написанию надежных программ.

**Ваsic(Бэйсик)** (Beginners All – purpose Symbolic Instruction Code) — многоцелевой язык символических инструкций для начинающих) представляет собой простой язык программирования, разработанный в 1964 году для использования новичками. Он был разработан как простейший язык для непосредственного общения человека с вычислительной машиной. Поэтому первоначально работа велась в интерактивном режиме с использованием интерпретаторов. В настоящее время для этого языка имеются также и компиляторы.

Согласно концепциям, заложенным в **Basic**, этот язык в смысле строгости и стройности является антиподом языка **Pascal.** В частности, в нем широко распространены различные правила умолчания, что считается плохим тоном в большинстве языков программирования подобного типа.

**Basic** широко применяется для ЭВМ различных типов и является очень популярным в среде программистов, особенно начинающих. Существует множество диалектов этого языка, мало совместимых между собой. **Basic** активно поглощает многие концепции и новинки из других языков. Поэтому он достаточно динамичен, и нельзя однозначно определить его уровень.

**Pascal (Паскаль)** является одним из наиболее популярных языков программирования среди прикладных программистов процедурным языком программирования, особенно для **ПЭВМ**. Разработанный в 1970 году швейцарским специалистом в области вычислительной техники профессором Н. Виртом, язык назван в честь французского математика и по замыслу автора предназначался для обучения программированию. Однако язык получился настолько удачным, что стал одним из основных инструментов прикладных и системных программистов при решении задач вычислительного и информационно-логического характера.

В языке **Pascal** реализован ряд концепций, рассматриваемых как основа «дисциплинированного» программирования и заимствованных впоследствии разработчиками многих языков. Одним из существенных признаков языка **Pascal** является последовательная и достаточно полная реализация концепции структурного программирования. Причем это осуществляется не только путем упорядочивания связей между фрагментами программы по управлению, но и за счет структуризации данных. Кроме того, в языке реализована концепция определения новых типов данных на основе уже имеющихся. Этот язык, в отличие от языка **С**, является строго типизированным.

**Pascal** характеризуется:

* высоким уровнем;
* широкими возможностями;
* стройностью, простотой и краткостью;
* строгостью, способствующей написанию эффективных и надежных программ;
* высокой эффективностью реализации на ЭВМ.

**Pascal** реализован на ЭВМ различных типов, но наиболее распространен и развит для ПЭВМ. В настоящее время широко используются такие версии этого языка для ПЭВМ, как **Borland Pascal** и **Turbo Pascal**.

Сущность ***функционального (аппликативного) программирования*** определена А. П. Ершовым как «... способ составления программ, в которых единственным действием является вызов функции, единственным способом расчленения программы на части является введение имени для функции, а единственным правилом композиции — оператор суперпозиции функции. Никаких ячеек памяти, ни операторов присваивания, ни циклов, ни, тем более, блок-схем, ни передачи управления».

Роль основной конструкции в функциональных языках играет выражение. К выражениям относятся скалярные константы, структурированные объекты, функции, тела функций и вызовы функций. Функция трактуется как однозначное отображение из X в *X,* где *X* –множество выражений.

***Аппликативный язык программирования*** включает следующие элементы:

* классы констант, которыми могут манипулировать функции;
* набор базовых функций, которые программист может использовать без предварительного объявления и описания;
* правила построения новых функций из базовых;
* правила формирования выражений на основе вызовов функций.

***Программа*** представляет собой совокупность описаний функций и выражения, которые необходимо вычислить. Данное выражение вычисляется посредством редукции, то есть серии упрощений, до тех пор, пока это возможно по следующим правилам: вызовы базовых функций заменяются соответствующими значениями; вызовы небазовых функций заменяются их телами, в которых параметры замещены аргументами.

***Функциональное*** программирование не использует концепцию памяти как хранилища значений переменных. Операторы присваивания отсутствуют, вследствие чего переменные обозначают не области памяти, а объекты программы, что полностью соответствует понятию переменной в математике. В принципе, можно составлять программы и вообще без переменных. Кроме того, нет существенных различий между константами и функциями, то есть между программами и данными. В результате этого функция может быть значением вызова другой функции и может быть элементом структурированного объекта. Число аргументов при вызове функции не обязательно должно совпадать с числом параметров, указанных при ее описании. Перечисленные свойства характеризуют аппликативные языки как языки программирования очень высокого уровня.

Первым таким языком был ***LISP(Лисп)*** (LIStProcessing– обработка списков), созданный в 1959 году. Цель его создания состояла в организации удобства обработки символьной информации. Существенная черта этого языка – унификация программных структур и структур данных: все выражения записываются в виде списков.

Новую область – ***логическое, или реляционное программирование***, – открыло появление языка ***PROLOG (Пролог)*** (PROgramminginLOGic– программирование в терминах логики). Этот язык был создан французским ученым А. Кольмероэ в 1973 году. В настоящее время известны и другие языки, однако наиболее развитым и распространенным языком логического программирования является именно Пролог. Так, имеется свыше 15 различных его реализаций на ПЭВМ. Языки логического программирования, в особенности Пролог, широко используются в системах искусственного интеллекта, рассматриваемых в данном учебном пособии.

Центральным понятием в логическом программировании является ***отношение****.* Программа представляет собой совокупность определений отношений между объектами (в терминах условий или ограничений) и цели (запроса). Процесс выполнения программы трактуется как процесс общей значимости логической формулы, построенной из программы по правилам, установленным семантикой используемого языка. Результат вычисления является побочным продуктом этого процесса. В реляционном программировании нужно только специфицировать факты, на которых алгоритм основывается, а не определять последовательность шагов, которые требуется выполнить. Это свидетельствует о декларативности языка логического программирования. Она метко выражена в формуле Р. Ковальского: «алгоритм = логика + управление». Языки логического программирования характеризуются:

* высоким уровнем;
* строгой ориентацией на символьные вычисления;
* возможностью инверсных вычислений, то есть переменные в процедурах не делятся на входные и выходные;
* возможной логической неполнотой, поскольку зачастую невозможно выразить в программе определенные логические соотношения, а также невозможно получить из программы только правильные выводы.

Логические программы, в принципе, имеют небольшое быстродействие, так как вычисления осуществляются методом проб и ошибок, поиском с возвратами к предыдущим шагам.

Прототипом **объектно-ориентированного программирования** послужил ряд средств, входящих в состав языка **SIMULA*-*67**. Но в самостоятельный стиль оно оформилось с появлением языка **SMALLTALK**, разработанного А. Кеем в 1972 году и первоначально предназначенного для реализации функций машинной графики.

В основе объектно-ориентированного стиля программирования лежит понятие объекта, а суть его выражается формулой: «объект - данные + процедуры». Каждый объект интегрирует в себе некоторую структуру данных и доступные только ему процедуры обработки этих данных, называемые ***методами***. Объединение данных и процедур в одном объекте называется ***инкапсуляцией*** и присуще объектно-ориентированному программированию.

Для описания объектов служат классы. ***Класс*** определяет свойства и методы объекта, принадлежащего этому классу. Соответственно, любой объект можно определить как ***экземпляр*** класса.

Программирование рассматриваемого стиля заключается в выборе имеющихся или создании новых объектов и организации взаимодействия между ними. При создании новых объектов свойства объектов могут добавляться или ***наследоваться*** от объектов-предков. В процессе работы с объектами допускается ***полиморфизм*** – возможность использования методов с одинаковыми именами для обработки данных разных типов.

К наиболее современным объектно-ориентированным языкам программирования относятся C++ и Java.

**Язык C++** был разработан в начале 80-х годов Б. Страуструпом, сотрудником лаборатории Bell корпорации AT&T. Им была создана компактная компилирующая система, в которой за основу был взят язык **С**, дополненный элементами языков BCPL, Simula-67 и Algol-68. К июлю 1983 года появился язык **С** с классами, а чуть позднее — C++. К 1990 году была выпущена третья версия языка **C++**, принятая комитетом ANSI в качестве исходного материала для его стандартизации.

В 1990 году сотрудник корпорации Sun Д. Гослинг на основе расширения C++ разработал объектно-ориентированный язык Oak, основным достоинством которого было обеспечение сетевого взаимодействия различных по типу устройств. Новая интегрируемая в Internet версия языка, получила название Java. Первый ***браузер,*** который поддерживал язык Java, разработан программистом корпорации Sun П. Нафтоном и получил название HotJava. С января 1995 года Java получает распространение в Internet.

Согласно официальному определению авторов, Java является простым объектно-ориентированным и архитектурно-нейтральным языком интерпретирующего типа, обеспечивающим надежность, безопасность и переносимость, обладающим высокой производительностью в сочетании с многопоточностью и динамичностью.

Принципиальной разницей между Java и C++ является то, что первый из них является ***интерпретируемым,*** а второй – ***компилируемым.*** Синтаксис языков практически полностью совпадает.

С точки зрения возможностей собственно объектно-ориентированных средств язык Java обладает рядом преимуществ перед языком C++. Так, язык Java демонстрирует более гибкую и мощную систему инкапсуляции информации. Механизм наследования, реализованный в Java, обязывает к более строгому подходу к программированию, что улучшает надежность и понимаемость кода. Язык же C++ обладает сложной, неадекватной и трудной для понимания системой наследования. Возможности динамического связывания объектов одинаково хорошо представлены в обоих языках, однако, синтаксическая избыточность C++ заставляет и здесь отдать предпочтение языку Java.

В силу своей конструктивности идеи объектно-ориентированного программирования используются во многих универсальных процедурных языках. Так, например, в состав интегрированной системы программирования на языке Pascal (корпорации Borland International) версии 5.5 входит специальная библиотека объектно-ориентированного программирования Turbo Vision.

В последнее время многие программы, в особенности объектно-ориентированные, реализуются как ***системы визуального программирования.*** Отличительной особенностью таких систем является мощная среда разработки программ из готовых «строительных блоков», позволяющая создать интерфейсную часть программного продукта в диалоговом режиме, практически без кодирования программных операций. К числу объектно-ориентированных систем визуального программирования относятся; **Visual Basic, Delphi, C++ Builder и Visual C++.**

В учебном процессе чаще всего используются системы программирования на базе языков - **Basic**, **Pascal**, **C++.**

**Язык Basic** был создан в 1965 г. как язык, облегчающий написание простых программ.  Существует много различных версий этого языка: от очень простых до усовершенствованных. Наибольшее распространение получила версия **Quick Basic 4.5**. Это очень простой, но в то же время эффективный язык, унаследовавший от раннего Бейсика все его достоинства, но избавившийся от всех его недостатков и впитавший целый ряд передовых идей начала 90-х годов. Одна из удач – это использование своеобразного режима работы. Для быстрой работы в среде используется режим **интерпретатора,** а для окончательного перевода отлаженных программ на машинный язык используется **компилятор.**

В последнее время получили распространение системы программирования, ориентированные на создание **Windows**-приложений:

* **Delphi** – наследник семейства компиляторов **Pascal,** предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.
* **VisualBasic** – удобный и популярный инструмент для создания **Windows**-программ с использованием визуальных средств.
* **Visual C++** – одно из самых распространённых средств для разработки **Windows** приложений.

### 2.1.4. Инструментальные программы и пакеты прикладных программ

**Инструментальные программные средства** – это программы, которые используются в ходе разработки, корректировки или развития **ПО** и часто являются составными частями систем программирования.

К инструментальным программам, например, относятся:

* редакторы;
* средства компоновки программ;
* отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
* вспомогательные программы, реализующие часто используемые системные действия;
* графические пакеты программ и т.п.

**Текстовый редактор** – это программа, используемая специально для ввода и редактирования текстовых данных. Этими данными могут быть программа или какой-либо документ или же книга. Редактируемый текст выводится на экран, и пользователь может в диалоговом режиме вносить в него свои изменения.

Возможности текстовых редакторов различны — от программ, предназначенных для подготовки небольших документов простой структуры, до программ для набора, оформления и полной подготовки к типографскому изданию книг и журналов (издательские системы).

Наиболее известный текстовый редактор –**MSWord**.

Полнофункциональные издательские системы – **MS Publisher**, **Corel Ventura** и **Adobe PageMaker**. Издательские системы незаменимы для компьютерной верстки и графики. Значительно облегчают работу с многостраничными документами, имеют возможности автоматической разбивки текста на страницы, расстановки номеров страниц, создания заголовков и т.д. Создание макетов любых изданий – от рекламных листков до многостраничных книг и журналов – становится очень простым, даже для новичков.

**Графический редактор** – это программа, предназначенная для автоматизации процессов построения на экране дисплея графических изображений. Предоставляет возможности рисования линий, кривых, раскраски областей экрана, создания надписей различными шрифтами и т.д. Большинство редакторов позволяют обрабатывать изображения, полученные с помощью сканеров, а также выводить картинки в таком виде, чтобы они могли быть включены в документ, подготовленный с помощью текстового редактора. Некоторые редакторы позволяют получать изображения трёхмерных объектов, их сечений, разворотов, каркасных моделей и т.п.

В настоящее время также пользуется известностью система **Corel DRAW**– мощный графический редактор с функциями создания публикаций, снабжённый инструментами для редактирования графики и трёхмерного моделирования.

**MS**  **OfficeVisio** – это средство построения диаграмм, которое позволяет создавать технические и бизнес - диаграммы для документирования и лучшего понимания и понимания сложных процессов и систем.

**Электронная таблица**– это компьютерный эквивалент обычной таблицы, состоящей из строк и граф, на пересечении которых располагаются клетки, в которых содержится числовая информация, формулы или текст.

Значение в числовой клетке таблицы может быть либо записано, либо рассчитано по соответствующей формуле; в формуле могут присутствовать обращения к другим клеткам.

Табличные процессоры представляют собой удобное средство для проведения бухгалтерских и статистических расчетов. В каждом пакете имеются сотни встроенных математических функций и алгоритмов статистической обработки данных. Кроме того, имеются мощные средства для связи таблиц между собой, создания и редактирования электронных баз данных.

Специальные средства позволяют автоматически получать и распечатывать настраиваемые отчеты с использованием десятков различных типов таблиц, графиков, диаграмм, снабжать их комментариями и графическими иллюстрациями.

Табличные процессоры имеют встроенную справочную систему, предоставляющую пользователю информацию по конкретным командам меню и другие справочные данные. Многомерные таблицы позволяют быстро делать выборки в базе данных по любому критерию.

Самый популярный табличный процессор – **MS Excel**. В нем автоматизированы многие рутинные операции, специальные шаблоны помогают создавать отчёты, импортировать данные и многое другое.

**База данных** – это один или несколько взаимосвязанных файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации. Базы данных используются под управлением систем управления базами данных.

**Система управления базами данных (СУБД)** - это система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ конечных пользователей.  Системы управления базами данных позволяют объединять большие объемы информации и обрабатывать их, сортировать, делать выборки по определённым критериям и т.п.

Современные **СУБД** дают возможность включать в них не только текстовую и графическую информацию, но и звуковые фрагменты и даже видеоклипы.

Простота использования **СУБД** позволяет создавать новые базы данных, не прибегая к программированию, а пользуясь только встроенными функциями. **СУБД** обеспечивают правильность, полноту и непротиворечивость данных, а также удобный доступ к ним.

Популярные СУБД– FoxPro, Access for Windows, Oracle.

**Библиотека стандартных подпрограмм** - это совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих определенным единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм и т.п.

Стандартные подпрограммы имеют единую форму обращения, что обеспечивает простоту и удобство настройки параметров подпрограммы на решение конкретной задачи.

**Пакеты прикладных программ (ППП)**– это специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

В зависимости от характера решаемых задач различают следующие **ППП**:

* пакеты для решения типовых инженерных, планово-экономических, общенаучных задач;
* пакеты системных программ;
* пакеты для обеспечения систем автоматизированного проектирования и систем автоматизации научных исследований;
* пакеты обучающих программных средств и другие.

Чтобы пользователь мог использовать **ППП** для решения конкретной задачи, пакет должен обладать средствами настройки на определенную предметную область.

**ППП** обеспечивают значительное снижение требований к уровню профессиональной подготовки пользователей в области программирования, вплоть до возможности эксплуатации пакета без программиста.

Часто пакеты прикладных программ располагают базами данных для хранения данных и передачи их прикладным программам.

**Интегрированные пакеты** представляют собой набор нескольких программных продуктов, объединенных в единый удобный инструмент. Наиболее развитые из них включают в себя текстовый редактор, органайзер, электронную таблицу, **СУБД**, средства поддержки электронной почты, программу создания презентационной графики.

Результаты, полученные отдельными подпрограммами, могут быть объединены в окончательный документ, содержащий табличный, графический и текстовый материал.

Интегрированные пакеты, как правило, содержат некоторое ядро, обеспечивающее возможность тесного взаимодействия между составляющими.

Пример: **интегрированный пакет для написания книг, содержащих иллюстрации.** Он содержит:

* текстовый редактор;
* орфографический корректор (программу обнаружения орфографических ошибок);
* программу слияния текстов;
* программу формирования оглавлений и составления указателей;
* автоматический поиск и замену слов и фраз;
* средства телекоммуникации;
* электронную таблицу;
* систему управления базами данных;
* модули графического оформления;
* графический редактор;
* возможность печати сотнями разных шрифтов и т.д.

Наиболее известным интегрированным пакетом является **Microsoft Office.** В этот мощный профессиональный пакет вошли такие необходимые программы, как текстовый редактор **MS Word**, табличный процессор **MS Excel**, программа создания презентаций **PowerPoint, СУБД Access,** средство поддержки электронной почты **е-Mail*.*** Мало того, все части этого пакета составляют единое целое, и даже внешне все программы выглядят единообразно, что облегчает как их освоение, так и ежедневное использование.

Для инженерных и научных расчетов активно используются универсальные математические пакеты символьной и численной математики, такие, например, как **Mathematica, MathCAD, Maple и MatLab**.

### 2.1.5. Контрольные вопросы по теме

### «Программное обеспечение компьютера»

1. Что такое программа?
2. Что включает в себя понятие "программное обеспечение"?
3. Охарактеризуйте основные категории **ПО**.
4. В чем отличие прикладных программ от системных и инструментальных?
5. В чем состоит назначение операционной системы?
6. Что такое файл и как организована файловая система?
7. Приведите пример иерархической файловой структуры.
8. Что такое базовая система ввода-вывода (**BIOS**)?
9. Назовите основные разновидности программ-утилит.
10. К каким категориям программного обеспечения относятся программные пакеты: **Norton Commander, MS-DOS, Windows-2000, Windows XP,; MSWord, Adobe PageMaker, TPascal, Basic, MS Excel, MS Access, Microsoft Office**?
11. Для чего предназначен пакет программ **Norton Commander**?
12. В чем отличие командного и графического интерфейса?
13. Характеризуйте основные особенности операционных систем **Windows-2000 и Windows XP**.
14. Какие языки и системы программирования вы знаете и в чем их особенности?
15. В чем отличие процесса интерпретации от процесса компиляции?
16. Назовите основные функции текстовых редакторов.
17. Какие дополнительные возможности редактирования текстов обеспечивают полнофункциональные издательские системы по сравнению с текстовыми редакторами?
18. Назовите функциональные возможности табличного процессора.
19. Дайте определение и опишите назначение базы данных.
20. Приведите пример возможного наполнения базы данных вашего учебного заведения.
21. Каковы основные функциональные возможности систем управления базами данных?
22. Что такое информационно-поисковые системы?
23. Дайте определение пакета прикладных программ.
24. Каково назначение библиотек стандартных программ? Дайте определения интегрированного пакета программ.

### 2.1.6. Тестовые задания по теме

### «Программное обеспечение компьютера»

1. **Программой архиватором называют**
   1. программу для уменьшения информационного объема (сжатия) файлов
   2. транслятор
   3. программу резервного копирования файлов
   4. интерпретатор
   5. систему управления базами данных
2. **Компьютерные вирусы**
   1. пишутся людьми специально для нанесения ущерба пользователям персональных компьютеров
   2. зарождаются при работе неверно написанных программных продуктов
   3. являются следствием ошибок в операционной системе
   4. возникают в связи со сбоями в работе аппаратных средств компьютера
   5. имеют биологическое происхождение
3. **Браузеры (например, Microsoft Internet Explorer) являются**
4. средством просмотра web-страниц
5. серверами Интернет
6. антивирусными программами
7. трансляторами языка программирования
8. **Web-страница – это**
9. документ, в котором хранится информация пользователя
10. документ, в котором хранится информация сервера
11. документ, в котором хранится вся информация по сети
12. сводка меню программных продуктов
13. **Домен – это**
14. часть адреса, определяющая адрес компьютера пользователя в сети
15. единица измерения информации
16. название программы, для осуществления связи между компьютерами
17. название устройства, осуществляющего связь между компьютерами
18. **Гипертекст - это**
19. структурированный текст, в котором могут осуществляться переходы по выделенным меткам
20. очень большой текст
21. текст, набранный на компьютере
22. текст, в котором используется шрифт большого размера
23. **HTML (Hyper Text Markup Language) является**
24. средством создания Web-страниц
25. средством просмотра Web-страниц
26. транслятором языка программирования
27. сервером Интернет
28. **Для передачи в сети web-страниц используется протокол**
29. www
30. http
31. ftp
32. dns
33. **Текстовый редактор - это**
34. программа, предназначенная для работы с текстовой информацией в процессе делопроизводства, редакционно-издательской деятельности и др
35. программа автоматического перевода текста на символических языках в текст, записанный с использованием машинных кодов
36. программа обработки изображений при создании мультимедийных игровых программ
37. программа управления ресурсами персонального компьютера при создании документов
38. работник издательства, осуществляющий проверку и исправление ошибок в тексте при подготовке рукописи к печати
39. **Поиск слова в тексте по заданному образцу является процессом**
40. обработки информации
41. хранения информации
42. передачи информации
43. уничтожения информации
44. получения информации
45. **Windows – это**
46. операционная система
47. графическая программа
48. текстовый редактор
49. хорошая вещь
50. **Windows-приложения это**
51. компоненты Windows
52. DOS-программы
53. документы по Windows
54. калькулятор, блокнот, графический редактор
55. **Экран монитора называют**
56. рабочий стол Windows
57. Окно Windows
58. панель Windows
59. обои Windows
60. **Под программным обеспечением (SoftWare) понимается**
61. совокупность программ, выполняемых вычислительной техникой
62. правила техники безопасности
63. последовательность символов, слов, абзацев
64. инструкция по получению денег в банкомате
65. **Укажите правильные категории программного обеспечения**
66. прикладное, системное, инструментальное
67. системное, файловое, оптическое
68. электронный собеседник, электронные оболочки, команды
69. программы, коды, алгоритмы
70. **Антивирусные программы предназначены для**
71. предотвращения заражения компьютерными вирусами и ликвидации последствий заражения вирусами
72. архивации (сжатия) информации на диске
73. вывода телевизионных сигналов на монитор компьютера на фоне работы других программ
74. воспроизведения последовательности картинок, создающих впечатление движущегося изображения
75. **Программы вспомогательного назначения из класса системных называются**
76. Утилиты (лат utilitas - польза)
77. программы контроля, тестирования и диагностики
78. антивирусные программы
79. программы-драйверы
80. системные программы
81. **Операционная система выполняет функции по**
82. организации диалога с пользователем, управления аппаратурой и ресурсами компьютера
83. обеспечению организации и хранения файлов
84. подключению устройств ввода/вывода
85. организации обмена данными между компьютером и различными периферийными устройствами
86. **BIOS находится**
87. в постоянно-запоминающем устройстве (ПЗУ)
88. в оперативно-запоминающем устройстве (ОЗУ)
89. на винчестере
90. на CD-ROM
91. **Текущий диск – это**
92. диск, с которым пользователь работает в данный момент времени
93. CD-ROM
94. жесткий диск
95. диск, в котором хранится операционная система
96. **Системные программы**
97. управляют работой аппаратных средств и обеспечивают услугами нас и наши прикладные комплексы
98. управляют работой ЭВМ с помощью электрических импульсов
99. игры, драйверы, трансляторы и тд
100. программы, которые хранятся на жестком диске
101. **Hardware это**
102. одна из составляющих информационной технологии - аппаратное обеспечение
103. одна из составляющих информационной технологии - программное обеспечение
104. жесткий диск, находящийся внутри системного блока
105. **Software это**
106. одна из составляющих информационной технологии - программное обеспечение
107. одна из составляющих информационной технологии - аппаратное обеспечение
108. жесткий диск, находящийся внутри блока
109. компактный диск
110. **Программа**
111. это набор инструкций на машинном языке, который хранится в виде файла на магнитном диске и по вашей команде загружается в компьютер для выполнения
112. это игры, предназначенные для использования на ЭВМ
113. это набор инструкций, предназначенный для запуска компьютера
114. это набор инструкций, предназначенных для работы компьютера
115. **Драйверы устройств**
116. это программные средства, предназначенные для подключения устройств ввода/вывода
117. это аппаратные средства, подключенные к компьютеру для осуществления операций ввода/вывода
118. это программа, переводящая языки высокого уровня в машинный код
119. это программа, позволяющая повысить скорость работы пользователя на ЭВМ
120. **Прикладные программы**
121. программы, предназначенные для решения конкретных задач
122. управляют работой аппаратных средств и обеспечивают услугами нас и наши прикладные комплексы
123. игры, драйверы, трансляторы и тд
124. программы, которые хранятся на различного типа дискетах
125. **В состав ОС не входит**
126. драйверы
127. BIOS
128. программа-загрузчик
129. ядро ОС
130. **Ярлык – это**
131. графическое изображение файла, папки или программы
132. копия файла, папки или программы
133. директория
134. перемещенный файл, папка или программа
135. **Жесткому диску соответствует соответствует имя**
136. C:
137. B:
138. A:
139. **Текущий каталог - это каталог**
140. с которым работает или работал пользователь на данном диске
141. в котором хранятся все программы операционной системы
142. объем которого изменяется при работе компьютера
143. в котором находятся файлы, созданные пользователем
144. **DOS предназначена для**
145. организации управления компьютером и его ресурсами через набор элементарных операций
146. работы с внешними устройствами
147. только для обработки и хранения файлов
148. обмена данными между компьютером и различными периферийными устройствами
149. **Графическим редактором называется программа, предназначенная для**
150. работы с графическим изображением
151. создания графического образа текста
152. редактирования вида и начертания шрифта
153. построения диаграмм
154. **Примитивами в графическом редакторе называются**
155. линия, круг, прямоугольник
156. карандаш, кисть, ластик
157. выделение, копирование, вставка
158. наборы цветов (палитра)
159. **Инструментами в графическом редакторе являются**
160. карандаш, кисть, ластик
161. линия, круг, прямоугольник
162. выделение, копирование, вставка
163. наборы цветов (палитра)
164. **Минимальным объектом, используемым в векторном графическом редакторе, является**
165. точка экрана (пиксель)
166. объект (прямоугольник, круг и т.д.)
167. палитра цветов
168. знакоместо (символ)
169. **Редакционно-издательские технологии предназначены для**
170. обработки текстовой информации
171. обработки больших объемов структурированной информации
172. решения вычислительных задач и обеспечения экономической деятельности
173. обработки реальных изображений и звука
174. создания инструментальных программных средств информационных технологий
175. **Технологии баз данных предназначены для**
176. обработки больших объемов структурированной информации
177. обработки текстовой информации
178. решения вычислительных задач и обеспечения экономической деятельности
179. обработки реальных изображений и звука
180. создания инструментальных программных средств информационных технологий
181. **Офисные технологии предназначены для**
182. решения вычислительных задач и обеспечения экономической деятельности
183. обработки больших объемов структурированной информации
184. обработки текстовой информации
185. обработки реальных изображений и звука
186. создания инструментальных программных средств информационных технологий
187. **Графическими редакторами являются следующие программы**
188. Coreldraw
189. Excel
190. FoxPro
191. Basic
192. Access
193. Word
194. Pascal
195. **Системами управления базами данных являются следующие программы**
196. Access
197. FoxPro
198. Excel
199. Basic
200. Coreldraw
201. Word
202. Pascal
203. **Электронными таблицами являются следующие программы**
204. Excel
205. Supercalc
206. FoxPro
207. Basic
208. Access
209. Lexicon
210. Coreldraw
211. Word
212. Pascal
213. **Операционная система(ОС) – это**
214. система программ, осуществляющая общее управление работой устройств компьютера
215. главный электронный блок компьютера
216. программа, выполняющая арифметические и логические операции
217. программа, управляющая работой компьютера в каждый конкретный момент времени
218. программа, обеспечивающая доступ пользователя к ресурсам компьютера
219. **В состав ОС входят следующие подсистемы**
220. управления устройствами
221. управления файлами
222. управления программами
223. управления памятью компьютера
224. обработки ошибок
225. распознавания образов
226. управления базами данных
227. расширения комплекта компьютера
228. **К числу операционных систем относятся**
229. MS DOS
230. Windows31
231. Windows 95
232. UNIX
233. OS/2
234. Norton Commander
235. **Диалоговая оболочка - это**
236. программа, обеспечивающая доступ пользователя к ресурсам компьютера
237. главный электронный блок компьютера
238. система программ, осуществляющая общее управление работой устройств компьютера
239. программа, выполняющая арифметические и логические операции
240. программа, управляющая работой компьютера в каждый конкретный момент времени
241. **Назначением графических редакторов является**
242. построение графических изображений
243. создание анимационных изображений (мультипликации)
244. создание графического представления таблицы (диаграмм)
245. обработка текстовой информации
246. **Операционная система – это** 
     1. набор программ, обеспечивающий работу всех аппаратных устройств компьютера и доступ пользователя к ним
     2. система программирования на языке низкого уровня
     3. совокупность программ, используемых для операций с документами
     4. совокупность основных устройств компьютера
     5. программа для уничтожения компьютерных вирусов
247. **Принцип программного управления работой компьютера предполагает** 
     1. необходимость использования операционной системы для синхронной работы аппаратных средств
     2. двоичное кодирование данных в компьютере
     3. возможность выполнения без внешнего вмешательства целой серии команд
     4. моделирование информационной деятельности человека при управлении компьютером
     5. использование формул исчисления высказываний для реализации команд в компьютере
248. **Технология программирования предназначена для**
249. создания инструментальных программных средств информационных технологий
250. обработки текстовой информации
251. решения вычислительных задач и обеспечения экономической деятельности
252. обработки реальных изображений и звука
253. обработки больших объемов структурированной информации
254. **Языками программирования являются следующие программы**
255. Pascal
256. Basic
257. Excel
258. FoxPro
259. Access
260. Lexicon
261. Supercalc
262. Coreldraw
263. Word
264. **Текстовыми редакторами являются следующие программы**
265. Word
266. Lexicon
267. Excel
268. FoxPro
269. Basic
270. Access
271. Supercalc
272. Coreldraw
273. Pascal
274. **Графическими редакторами являются следующие программы**
275. Coreldraw
276. Excel
277. FoxPro
278. Basic
279. Access
280. Word
281. Pascal
282. **Системами управления базами данных являются следующие программы**
283. Access
284. FoxPro
285. Excel
286. Basic
287. Coreldraw
288. Word
289. Pascal
290. **Электронными таблицами являются следующие программы**
291. Supercalc
292. Excel
293. FoxPro
294. Basic
295. dAccess
296. Lexicon
297. Coreldraw
298. Word
299. Pascal
300. **В состав ОС входят следующие подсистемы**
301. управления устройствами
302. управления файлами
303. управления памятью компьютера
304. управления программами
305. обработки ошибок
306. распознавания образов
307. управления базами данных
308. расширения комплекта компьютера