## Тема 1.2. Классификация ПО

**1.2.1. Принципы классификации ПО**

**1.2.2. Операционные системы**

**1.2.3. Системы программирования**

**1.2.4. Инструментальные программы и пакеты прикладных программ**

**1.2.5. Прикладное ПО и Интегрированные программные продукты**

### Принципы классификации ПО

***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.

Любая классификация подразумевает выбор некоторого **признака** (или нескольких признаков), на основании которого объекты относят к тому или иному классу. Так, при классификации программного обеспечения по способу распространения можно выделить следующие категории:

* ***CommercialSoftware*** – ***коммерческоеПО*** (с ограниченными лицензией возможностями на использование), разрабатываемое для получения прибыли.
* ***Freeware*** – ***свободное ПО***, распространяемое без ограничений на использование, модификацию и распространение.
* ***Shareware – условно-бесплатное ПО*** (с частичными ограничениями при работе в ознакомительном режиме (например, определенное количество запусков программы).

С точки зрения расположения в сети, программное обеспечение подразделяют на Серверное и Клиентское:

* Серверное ПО — выполняющее коллективные задачи, получая задания от нескольких пользователей одновременно, например интернет сайт, файловый сервер, сервер баз данных.
* Клиентское ПО — служащее для работы с серверами: почтовая программа, программа обмена мгновенными сообщениями, интернет обозреватель.

При классификации программного обеспечения по назначению в качестве критерия используют уровень представления ПК, на который ориентирована та или иная программа. Соответственно выделяют следующие классы ПО:

* ***Системное ПО*** – решает задачи общего управления и поддержания работоспособности системы в целом. К этому классу относят операционные системы, менеджеры загрузки, драйверы устройств, программные кодеки, утилиты и программные средства защиты информации;
* ***Прикладное ПО*** – предназначено для решения прикладных задач конечными пользователями;
* ***Инструментальное ПО*** - включает средства разработки (трансляторы, отладчики, интегрированные среды, различные SDK и т.п.) и системы управления базами данных (СУБД).

***Прикладное ПО*** – самый обширный класс программ, в рамках которого возможна дальнейшая классификация, например по предметным областям. В этом случае признаком является ***класс задач***, решаемых программой. Приведем несколько примеров:

* ***Математические пакеты***
* ***Офисные прикладные программы*** - предназначены для автоматизации офисной деятельности (текстовые редакторы и процессоры, электронные таблицы, редакторы презентаций и т.п.);
* ***Интегрированные ППП для разработки ПО***
* ***Графические пакеты***
* ***Браузеры***
* ***Корпоративные информационные системы*** - бухгалтерские программы, системы корпоративного управления, системы управления проектами (ProjectManagement), инструменты автоматизации документооборота (EDM-системы) и управления архивами документов (DWM-системы);
* ***Системы проектирования и производства*** - системы автоматизированного проектирования (САПР, CAD/CAM-системы), системы управления технологическими (SCADA) и производственными (MES) процессами;
* ***Научное ПО*** - системы математического и статистического расчета, анализа и моделирования;
* ***Геоинформационные системы*** (ГИС);
* ***Системы поддержки принятия решений*** (СППР);
* Приложения, осуществляющие доступ к сетевым сервисам (электронная почта, веб-браузеры, передача сообщений, чат-каналы, клиенты файл обменных сетей и т.п.);
* ***Мультимедийное ПО*** - компьютерные игры, средства просмотра и редактирования аудио- и видеоинформации, графические редакторы и анимационные редакторы и т.п.

|  |  |  |
| --- | --- | --- |
|  | ПО |  |
| по способу распространения |  | уровень представления ПК |
|  |  |  |
| ***Freeware***–***свободное ПО*** |  | ***Системное ПО*** |
| ***Shareware– условно-бесплатное ПО*** |  | ***Прикладное ПО*** |
| ***CommercialSoftware***–***коммерческоеПО*** |  | ***Инструментальное ПО*** |
|  |  |  |
|  | * ***Математические пакеты*** |  |
|  | * ***Офисные приложения*** |  |
|  | * ***Графические пакеты*** |  |
|  | * ***Интегрированные ППП для разработки ПО*** |  |
|  | * ***Мультимедийное ПО*** |  |
|  | * ***Браузеры*** |  |
|  | * ***Научное ПО*** |  |
|  | * ***ППП, обеспечивающие информационную безопасность*** |  |

Рис. 1.2-1.

С точки зрения конечного пользователя такая классификация оправданна и наглядна, однако, для разработчика же более значимым фактором является структура прикладной программы, в общем случае состоящей из нескольких компонентов. Назначение этих компонентов, связи между ними и способность к взаимодействию определяют интеграцию прикладного ПО. Чем теснее связаны программные компоненты, тем выше степень интеграции.

В зависимости от степени интеграции многочисленные прикладные программные средства можно классифицировать следующим образом:

* ***отдельные прикладные программы;***
* ***библиотеки прикладных программ;***
* ***пакеты прикладных программ;***
* ***интегрированные программные системы.***

***Отдельная прикладная программа*** пишется, как правило, на некотором высокоуровневом языке программирования (Pascal, Basic и т.п.) и предназначается для решения конкретной прикладной задачи. Такая программа может быть реализована в виде набора модулей, каждый из которых выполняет некоторую самостоятельную функцию (например, модуль пользовательского интерфейса, модуль обработки ошибок, модуль печати и т.п.). При этом доступ к функциям модулей из внешних программ невозможен.

***Библиотека*** представляет собой набор отдельных программ, каждая из которых решает некоторую прикладную задачу или выполняет определенные вспомогательные функции (управление памятью, обмен с внешними устройствами и т.п.). Библиотеки программ зарекомендовали себя эффективным средством решения вычислительных задач. Они интенсивно используются при решении научных и инженерных задач с помощью ПК. Условно их можно разделить на библиотеки общего назначения и специализированные библиотеки.

***Пакет прикладных программ (ППП)*** - это комплекс взаимосвязанных программ, ориентированный на решение определенного класса задач. Формально такое определение не исключает из числа пакетов и библиотеки программ, однако у ППП, как отдельной категории, есть ряд особенностей, среди которых: ориентация на решение классов задач, унифицированный интерфейс, наличие языковых средств.

***Интегрированная программная система*** – это комплекс программ, элементами которого являются различные пакеты и библиотеки программ. Примером служат системы автоматизированного проектирования, имеющие в своем составе несколько ППП различного назначения. Часто в подобной системе решаются задачи, относящиеся к различным классам или даже к различным предметным областям.

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.**

### Операционные 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 **ОС** **DOS**. 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.

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,** котораябыла 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.

### 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.(команду за командой).

После того, как программа откомпилирована, ни сама исходная программа, ни компилятор более 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 распространен на ЭВМ различных типов и очень популярен 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 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 не базовых 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)*** (LISt Processing – 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)***(PROgramming in LOGic – 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.
* **Visual Basic** – удобный и популярный инструмент для создания **Windows**-программ с использованием визуальных средств.
* **Visual C++** – одно из самых распространённых средств для разработки **Windows** приложений.

### 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 – **MS Word**.

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**  **Office Visio** – 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 немавтоматизированы 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 **E-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 **Mathematic, MathCAD, Maple и MatLab**.

### Прикладное ПО и Интегрированные программные продукты

***Прикладные программы*** (приложения, application) предназначены для того, чтобы обеспечить применение вычислительной техники в различных сферах деятельности человека. Из-за огромного разнообразия ***прикладного программного обеспечения*** (ППО) существует множество вариантов его классификации. Наиболее общая классификация предполагает разделение ППО на два основных класса:

1. Прикладные программы ***общего назначения***. К таким относятся программы, обеспечивающие выполнение наиболее часто используемых, универсальных задач (текстовые редакторы, табличные процессоры, графические редакторы, Интернет-клиенты и т.п.).
2. Прикладные программы ***специального (профессионального) назначения***. Программы этого класса ориентированы на достаточно узкую предметную область, (издательские системы; САПР - системы автоматизированного проектирования; банковские, бухгалтерские программы; программы 3D-графики; программы видеомонтажа; нотные редакторы и т.д.).

***Средства обработки текстовой информации***

Несмотря на широкие возможности использования компьютеров для обработки самой разной информации, самыми популярными по-прежнему остаются программы, предназначенные для работы с текстом. При подготовке текстовых документов на компьютере используются три основные группы операций:

* ***Операции ввода*** позволяют перенести исходный текст из его внешней формы в электронный вид, то есть в файл, хранящийся на компьютере. Ввод может осуществляться не только набором с помощью клавиатуры, но и путем сканирования бумажного оригинала c и последующегоим переводаом документа из графического формата в текстовый (распознавание).
* ***Операции редактирования*** (правки) позволяют изменить уже существующий электронный документ путем добавления или удаления его фрагментов, перестановки частей документа, слияния нескольких файлов, разбиения единого документа на несколько более мелких и т.д.
* ***Форматирование текста.*** Команды форматирования позволяют точно определить, как будет выглядеть текст на экране монитора или на бумаге после печати на принтере.

Программы, предназначенные для обработки текстовой информации, называют ***текстовыми редакторами***. Все многообразие современных текстовых редакторов условно можно разбить на три основные группы:

1. ***Простейшие текстовые редакторы***, обладающие минимумом возможностей и способные работать с документами в обычном текстовом формате .txt, который, как известно, при всей своей простоте обладает довольно слабыми возможностями форматирования. К этой группе редакторов можно отнести как входящие в комплект поставки ОС семейства Windows редакторы WordPad и совсем малофункциональный*NotePad (Блокнот)*, и множество аналогичных продуктов других производителей (Atlantis, EditPad, AditorPro, Gedit и т.д.).
2. ***Промежуточный класс текстовых редакторов*** включает в себя достаточно широкие возможности по части оформления документов. Они работают со всеми стандартными текстовыми файлами(TXT, RTF, DOC). К таким программам можно отнести *MicrosoftWorks, Лексикон*.
3. ***Мощные текстовые процессоры***, такие, как *MS Word* или *StarOfficeWriter* позволяют выполнять практически все операции с текстом. Большинство пользователей использует именно эти редакторы в повседневной работе.

Основными функциями текстовых редакторов и процессоров являются:

* ввод и редактирование символов текста;
* возможность использования различных шрифтов символов;
* копирование и перенос части текста с одного места на другое или из одного документа в другой;
* контекстный поиск и замена частей текста;
* задание произвольных параметров абзацев и шрифтов;
* автоматический перенос слов на новую строку;
* автоматическую нумерацию страниц;
* обработка и нумерация сносок;
* создание таблиц и построение диаграмм;
* проверка правописания слов и подбор синонимов;
* построение оглавлений и предметных указателей;
* распечатка подготовленного текста на принтере и т.п.

Также практически все текстовые процессоры обладают следующими функциями:

* поддержка различных форматов документов;
* многооконность, т.е. возможность работы с несколькими документами одновременно;
* вставка и редактирование формул;
* автоматическое сохранение редактируемого документа;
* работа с многоколоночным текстом;
* возможность работы с различными стилями форматирования;
* создание шаблонов документов;
* анализ статистической информации.

Сегодня практически все мощные текстовые редакторы входят в состав интегрированных программных пакетов, предназначенных для нужд современного офиса. Так, например, *MS Word* входит в состав самого популярного офисного пакета *MS Office*.

Аналогичные MS Office программы: — OpenOffice.org Writer, StarOffice Writer, Corel WordPerfect, Apple Pages.

***Средства обработки табличной информации***

Табличное представление данных имеет свои особенности. Многие виды данных намного удобнее хранить и обрабатывать в табличной форме, особенно числовые. Автоматизация табличных расчетов во много раз повышает эффективность и качество работы. Компьютерные программы, предназначенные для хранения и обработки данных, представленных в табличном виде, называют ***электронными таблицами*** или ***табличными процессорами***.

Первая электронная таблица VisiCalc была выпущена фирмой VisiCorporation в 1981 году, и именно с этого момента принято вести отсчет истории электронных таблиц как самостоятельного вида программного обеспечения. Идея выделения таблиц в особый класс документов и создание специализированной программы, выполняющей всевозможные операции с табличными данными, оказалась весьма удачной и была подхвачена многими фирмами. Популярность электронных таблиц стремительно росла.

В 1983 году фирма LotusDevelopmentCorporation выпустила электронную таблицу 1-2-3, ставшую на долгие годы фактическим стандартом в своей области.

В 1985 году появилась первая наиболее распространенный на сегодня табличный редактор MicrosoftExcel. Спустя год данный сектор desktop-приложений пополнился пакетом Quattro, созданным компанией BorlandInternationalCorporation.

***Электронные таблицы позволяют решать целый комплекс задач:***

1. ***Выполнение вычислений****.* Издавна многие расчеты выполняются в табличной форме, особенно в области делопроизводства: многочисленные расчетные ведомости, табуляграммы, сметы расходов и т.п. Кроме того, решение численными методами целого ряда математических задач удобно выполнять в табличной форме. Электронные таблицы представляют собой удобный инструмент для автоматизации таких вычислений. Решения многих вычислительных задач на ЭВМ, которые раньше можно было осуществить только путем программирования, стало возможно реализовать на электронных таблицах.
2. Математическое моделирование. Использование математических формул в ЭТ позволяет представить взаимосвязь между различными параметрами некоторой реальной системы. Основное свойство ЭТ - мгновенный пересчет формул при изменении значений входящих в них операндов. Благодаря этому свойству, таблица представляет собой удобный инструмент для организации эксперимента: подбор параметров, прогноз поведения моделируемой системы, анализ зависимостей, планирование. Дополнительные удобства для моделирования дает возможность графического представления данных.
3. Использование электронной таблицы в качестве базы данных. Конечно, по сравнению с СУБД электронные таблицы имеют меньшие возможности в этой области. Однако некоторые операции манипулирования данными, свойственные реляционным СУБД, в них реализованы. Это поиск данных по заданным условиям и сортировка данных.

Одним из самых популярных табличных процессоров сегодня является MS Excel, входящий в состав пакета MicrosoftOffice. Главные конкуренты — OpenOffice.org Calc, StarOfficeCalcи Corel Quattro Pro.

***Средства обработки графической информации***

Для работы с компьютерной графикой существует множество классов ПО. Различают обычно два основных вида компьютерной графики, которые отличаются принципами формирования изображения: ***растровая*** и ***векторная***. Ниже приведено сравнение растрового и векторного изображения.

. Компьютерное ***растровое изображение*** представляется в виде прямоугольной матрицы, каждая ячейка которой - цветная точка. Т.е. основным элементом растрового изображения является точка. Если изображение экранное, то эта ***точка называется пикселем.***

При создании растровых изображений необходимо задавать разрешение и размеры изображения. В зависимости от того, какое графическое разрешение экрана используется операционной системой, на экране могут размещаться изображения, имеющие 640х480, 800х600, 1024х768 и более пикселей.

***Разрешение*** изображения измеряется в точках на дюйм (dotsperinch - dpi) (1 дюйм = 25,4 мм). Полиграфическая печать полноцветного изображения требует разрешения не менее 200-300 dpi.

С помощью растровой графики можно отразить и передать всю гамму оттенков и тонких эффектов, присущих реальному изображению. Растровое изображение ближе к фотографии, оно позволяет более точно воспроизводить основные характеристики фотографии: освещенность, прозрачность и глубину резкости. Чаще всего растровые изображения получают с помощью сканирования фотографий и других изображений, с помощью цифровой фотокамеры или путем "захвата" кадра видеосъемки.

*Основным недостатком растровых изображений является невозможность их увеличения* для рассмотрения деталей. При увеличении изображения точки становятся крупнее, но дополнительная информация не появляется. Этот эффект называется *пикселизацией*

К числу простейших растровых редакторов относятся *PaintBrush*, *Paint*, *Painter*, которые позволяют непосредственно рисовать простейшие растровые изображения.

Основной класс растровых графических редакторов предназначен для обработки готовых растровых изображений с целью улучшения их качества и создания собственных изображений из уже имеющихся. К таким редакторам относятся такие мощные программы, как *AdobePhotoshop*, *CorelPhotoPaint*,*Gimp* и другие.

К основным растровым форматам относятся:

* *BMP (WindowsDeviceIndependentBitmap)* - самый простой растровый формат является форматом Windows, он поддерживается всеми графическими редакторами, работающими под ее управлением. В BMP данные о цвете хранятся только в модели RGB, поддерживаются как индексированные цвета (до 256 цветов), так и полноцветные изображения. Благодаря примитивнейшему алгоритму записи изображения, при обработке файлов формата BMP очень мало расходуется системных ресурсов, поэтому этот формат часто используется для хранения логотипов, экранных заставок, иконок и прочих элементов графического оформления программ.
* *GIF (GraphicsInterchangeFormat)* - является одним из самых популярных форматов изображений, размещаемых на веб-страницах. Отличительной его особенностью является использование режима индексированных цветов (не более 256), что ограничивает область применения формата изображениями, имеющими резкие цветовые переходы. Небольшие размеры файлов изображений обусловлены применением алгоритма сжатия без потерь качества, благодаря чему изображения в этом формате наиболее удобны для пересылки по каналам связи глобальной сети. В GIF реализован *эффект прозрачности* и возможности хранить в одном файле несколько картинок с указанием времени показа каждой, что используется для создания *анимированных* изображений.
* *PNG (PortableNetworkGraphics)* - формат PNG, являющийся плодом трудов сообщества независимых программистов, появился на свет как ответная реакция на переход популярнейшего формата GIF в разряд коммерческих продуктов. Этот формат, в отличие от GIF сжимает растровые изображения не только по горизонтали, но и по вертикали, что обеспечивает более высокую степень сжатия. Как недостаток формата часто упоминается то, что он не дает возможности создавать анимационные ролики. Зато формат PNG позволяет создавать изображения с 256 уровнями прозрачности что, безусловно, выделяет его на фоне всех существующих в данный момент форматов. Так как формат создавался для Интернета, в его заголовке не предназначено место для дополнительных параметров типа разрешения, поэтому для хранения изображений, подлежащих печати, PNG плохо подходит, для этих целей лучше подойдет PSD или TIFF.
* *JPEG (JointPhotographicExpertsGroup)* - самый популярный формат для хранения фотографических изображений, является общепризнанным стандартом. JPEG может хранить только 24-битовые полноцветные изображения. Хотя JPEG отлично сжимает фотографии, но это сжатие происходит с потерями и портит качество, тем не менее, он может быть легко настроен на минимальные, практически незаметные для человеческого глаза, потери. Однако не стоит использовать формат JPEG для хранения изображений, подлежащих последующей обработке, так как при каждом сохранении документа в этом формате процесс ухудшения качества изображения носит лавинообразный характер. Наиболее целесообразно будет корректировать изображение в каком-нибудь другом подходящем формате, например TIFF, и лишь по завершению всех работ окончательная версия может быть сохранена в JPEG. Формат JPEG не поддерживает анимацию или прозрачный цвет, и пригоден в подавляющем большинстве случаев только для публикации полноцветных изображений, типа фотографических, в Интернете.
* *TIFF (Tag Image File Format)*. Как универсальный формат для хранения растровых изображений, TIFF достаточно широко используется, в первую очередь, в издательских системах, требующих изображения наилучшего качества. Кстати, возможность записи изображений в формате TIFF является одним из признаков высокого класса современных цифровых фотокамер. В этом формате поддерживаются такие чисто профессиональные возможности, как обтравочные контуры, альфа-каналы, возможность сохранять несколько копий изображения с разным разрешением и даже включать в файл слои. Благодаря своей совместимости с большинством профессионального ПО для обработки изображений, формат TIFF очень удобен при переносе изображений между компьютерами различных типов.
* *PSD (AdobePhotoshop)* - является стандартным форматом пакета AdobePhotoshop и отличается от большинства обычных растровых форматов возможностью хранения слоев (layers). Он содержит много дополнительных переменных (не уступает TIFF по их количеству) и сжимает изображения иногда даже сильнее, чем PNG (в тех случаях, когда размеры файла измеряются не в килобайтах, а в десятках или даже сотнях мегабайт).

Основным логическим элементом векторной графики является***геометрический объект****.* В качестве объекта принимаются простые геометрические фигуры (так называемые примитивы - прямоугольник, окружность, эллипс, линия). Благодаря этому форму, цвет и пространственное положение составляющих изображение объектов можно описывать с помощью***математических формул***.

Преимуществами векторной графики принято считать следующее:

* Она экономна в плане дискового пространства, необходимого для хранения изображений: это связано с тем, что сохраняется не само изображение, а только некоторые основные данные, используя которые, программа всякий раз воссоздает изображение заново.
* Объекты векторной графики легко трансформируются и масштабируются, что не оказывает практически никакого влияния на качество изображения. Масштабирование, поворот, искривление сводятся к элементарным преобразованиям над векторами.
* Программы векторной графики имеют развитые средства интеграции изображений и текста, единый подход к ним. Поэтому программы векторной графики незаменимы в области дизайна, технического рисования, для чертежно-графических и оформительских работ.

К недостаткам векторной графики относят:

* Векторная графика ограничена в живописных средствах: в программах векторной графики практически невозможно создавать фотореалистичные изображения.
* Векторный принцип описания изображения не позволяет автоматизировать ввод графической информации, как это делает сканер для растровой графики.

Для работы с векторной имеется огромное количество программных средств, однако его освоение сложнее по сравнению с растровыми. К основным относятся программным средствам относят:

* *CorelDraw* - это профессиональный графический редактор с богатыми настройками и развитой системой управления.
* *AdobeIllustrator*- основное достоинство программы в том, что она вместе с AdobePhotoshop и AdobePageMaker образует достаточно мощный пакет для выполнения компьютерной верстки полиграфических изданий и разработки сложных документов.
* *MacromediaFreehand* - один из самых дружественных и интуитивно понятных векторных редакторов. Программа отличается простотой системы управления и высоким быстродействием, но ее возможности несколько скромнее, чем у предыдущих редакторов.

Своего рода стандартом векторные графики стали форматы двух наиболее популярных профессиональных графических пакетов - AdobeIllustrator и CorelDRAW:

* *AI (AdobeIllustratorDocument)* - поддерживают практически все программы, так или иначе связанные с векторной графикой. Этот формат является наилучшим посредником при передаче изображений из одной программы в другую, с РС на Macintosh и наоборот. В целом, несколько уступая CorelDRAW по иллюстративным возможностям, (может содержать в одном файле только одну страницу, имеет маленькое рабочее поле - этот параметр очень важен для наружной рекламы - всего 3х3 метра) тем не менее, он отличается наибольшей стабильностью и совместимостью с языком PostScript, на который ориентируются практически все издательско-полиграфические приложения.
* *CDR (CorelDRAWDocument)* - основной рабочий формат популярного пакета CorelDRAW, являющимся неоспоримым лидером в классе векторных графических редакторов на платформе РС. Имея сравнительно невысокую устойчивость и проблемы с совместимостью файлов разных версий формата, тем не менее, формат CDR можно без натяжек назвать профессиональным. В файлах этих версий применяется раздельная компрессия для векторных и растровых изображений, могут внедряться шрифты, файлы CDR имеют огромное рабочее поле 45х45 метров, поддерживается многостраничность.
* *PDF (PortableDocumentFormat)* - первоначально проектировался как компактный формат электронной документации, но в последнее время все больше используется для передачи по сетям графических изображений и смешанных документов, содержащих как текст, так и графику. Формат PDF является в полной мере платформонезависимым форматом, в текстовой части которого возможно использование множества шрифтов (которые содержатся непосредственно в документе, поэтому документ будет выглядеть так, как задумал его автор, на любом компьютере) и гипертекстовых ссылок, а также графические иллюстрации любого типа (векторные или растровые). Для достижения минимального размера PDF-файла используется компрессия, причем каждый вид объектов сжимается по наиболее выгодному для него алгоритму. Просматривать документы в формате PDF и распечатывать их на принтере можно с помощью утилиты *AcrobatReader*, распространяемой компанией Adobe бесплатно. Если не популярны, то не надо описывать!

Особый класс программ для работы с любыми видами изображений представляют ***программы-просмотрщики***. Они позволяют просматривать графические файлы различных форматов, создавать фотоальбомы на жестком диске, перемещать, переименовывать, изменять размеры, а также конвертировать из одного формата в другой изображения. Лидером в данной области является программа *ACDSee*.

Быстро развивается область ***трехмерной векторной*** (или 3D) графики. К данному классу ПО относится, например, программа *3D-MAX Studio*. По сравнению с традиционными для графических программ двумерными векторными и растровыми объектами работа с 3D-графикой предполагает использование более сложных понятий и процедур, таких как сцена, камеры, источники света.

Возможности программ для работы с трехмерной графикой достаточно широки. Это и изготовление спецэффектов для кино и телевидения, получение реалистичных фотоизображений, технических иллюстраций в программах автоматизированного проектирования для разработки новых реальных объектов и т.д.

Использование 3D-программ напоминает съемку с помощью видеокамеры комнаты, полной сконструированных вами объектов. Они позволяют смоделировать комнату и ее содержимое с использованием разнообразных базовых трехмерных объектов (кубы, сферы, цилиндры, конусы). После того, как модели всех объектов созданы и размещены на сцене, можно выбрать им оформление с помощью имеющихся в программе встроенных средств или создать собственное оформление. Затем можно создать и расставить воображаемые камеры, которые будут наблюдать и снимать виртуальный трехмерный мир. После всех приготовлений можно анимировать сцену, заставив двигаться объекты, источники света и камеры. В завершении можно визуализировать анимацию и зафиксировать результат в виде видеофайла.

Системы управления базами данных (СУБД)

***База данных*** - совместно используемый набор логически связанных данных. Это единое хранилище данных, которое однократно определяется, а затем используется одновременно многими пользователями.

***Система управления базами данных*** *(СУБД)* - это программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ.

В ***реляционных*** базах данных (БД самого распространенного типа) данные хранятся в *таблицах*. На первый взгляд, эти таблицы подобны электронным таблицам Excel, поскольку они тоже состоят из строк и столбцов. Столбцы называются ***полями*** (fields) и содержат данные определенного типа. Строки именуются ***записями*** (records). В одной строке хранится один набор данных, описывающих определенный объект. Например, если в таблице хранятся данные о клиентах, она может содержать поля для имени, адреса, города, почтового индекса, номера телефона и т.д. Для каждого клиента будет создана отдельная запись.

***Таблицы*** – это не единственный тип объектов, из которых состоят базы данных. Помимо таблиц, существуют ***формы***, ***отчеты*** и ***запросы***.

* ***Формы*** (forms) применяются для добавления новых данных и изменения уже существующих. Формы облегчают добавление и редактирование информации, а также позволяют контролировать тип водимых данных и избегать при вводе ряда ошибок.
* ***Отчеты*** используются для отображения данных в удобном для чтения виде (reports). Ознакомиться со всей информацией, хранящейся в таблице, сложно по той причине, что текст не умещается в полях целиком. Существует возможность включать в отчет не все данные, а только некоторые, что значительно повышает удобство использования.
* ***Запросы*** применяются для вывода в отчеты определенных данных (queries). Использование запросов похоже на процесс поиска, – задаются конкретные критерии отбора, на основе которых база данных формирует и возвращает отчет. Например, если база данных содержит информацию о телефонных номерах, то можно запросить вывести в отчете только те телефоны, которые относятся к конкретному адресу, или только те, которые относятся к конкретной фамилии, или начинающиеся с определенных цифр и т.п. Запросы записываются на языке SQL (StructuredQueryLanguage — язык структурированных запросов).

В основе реляционных баз данных лежит понятие ***связей*** (*отношений*, *relationships*). Они позволяют разработчикам связывать несколько таблиц в базе посредством общих данных. При помощи взаимосвязей разработчики баз данных моделируют таблицы, отражающие взаимодействие объектов в реальности.

Модель реляционной СУБД была разработана в 70-80 годы XX века. К реляционным СУБД относится целый ряд программных продуктов, среди них *MicrosoftAccess из пакета MicrosoftOffice*, *MySQL* или более мощные системы промышленного уровня, таких как *Microsoft SQL Server* или *Oracle*.

В последнее время активно развивается и другая модель представления баз данных – ***объектная***. Реляционная модель акцентирует свое внимание на структуре и связях сущностей, объектная - на их свойствах и поведении.

***Средства разработки презентаций***

***Компьютерная презентация*** представляет собой набор слайдов (электронных страниц), последовательность показа которых может меняться в процессе демонстрации презентации, т.е. презентация является интерактивным документом. Презентация является мультимедийным документом, т.к. каждый слайд может включать в себя различные формы представления информации (текст, таблицы, диаграммы, изображения, звук, анимацию и др.). Презентации обычно используют в процессе выступлений на конференциях, для рекламы товаров на выставках, при объяснении нового материала на уроке и т.д. Наиболее популярной программой для создания презентаций является *MicrosoftPowerPoint*. Основныеконкуренты — *OpenOffice.org Impress, StarOffice Impress, Corel WordPerfect*и*Apple Keynote*.

***Автоматизация ввода информации в компьютер***

Основным методом перевода бумажных документов в электронную форму является ***сканирование***. В результате сканирования получается***графическое изображение***, состоящее из точек, т.е. растровое изображение. Количество точек определяется как размером изображения, так и разрешением сканера.

Существуют специальные стандарты программного интерфейса, обеспечивающего связь между сканерами и операционной системой. Этот интерфейс основан на специальном протоколе *TWAIN*.

Графический образ, получаемый после сканирования документа, иногда необходимо перевести в текст. Для этого используются специальные программные средства, называемые ***средствами распознавания образов***. Из программ, способных распознавать текст на русском языке наиболее известной является *ABBYY FineReader*.

Преобразование документа в электронный вид происходит в три основных этапа. Каждый из этих этапов может выполняться программами как автоматически, так и под контролем пользователя.

1. Сканирование. Запускается сканирующий модуль, настраиваются параметры сканирования (разрешение, размер, тип сканирования) и происходит собственно сканирование.

2. Сегментация и распознавание текста. Прежде чем получить готовый текст, необходимо разбить фрагменты документа на блоки (текст, рисунок, таблица и т.д.), для того, чтобы правильно их распознать (преобразовать в текстовый документ).

3. Проверка орфографии и передача текстового документа в нужное приложение для дальнейшей работы или сохранение в файл.

***Автоматизация перевода текста***

Идея автоматического перевода текстов с одного языка на другой зародилась с появлением первых компьютеров. Если бы полноценный перевод был возможен, то значительно упростилось бы общение между народами. Но это очень сложная задача, о полном решении которой пока говорить рано.

Программы автоматического перевода позволяют переводить отдельные слова и строить смысловые связи в предложениях, не всегда учитывая те или иные особенности языка. Поэтому они предназначены лишь для общего ознакомления с содержанием документа.

Программные средства автоматического перевода можно условно разделить на две основные категории:

1. Компьютерные словари. Назначение их - предоставить значения неизвестных слов быстро и удобно для пользователя.

2. Системы автоматического перевода - позволяют выполнять автоматический перевод связного текста. В ходе работы программа использует словари и наборы грамматических правил, обеспечивающих наилучшее качество перевода.

***Издательские системы***

***Компьютерные издательские системы*** - это комплекс аппаратных и программных средств, предназначенных для компьютерного набора, верстки и издания текстовых и иллюстративных материалов. Главным отличием настольных издательских систем от текстовых редакторов является то, что они предназначены, в первую очередь, для оформления документов, а не для ввода и редактирования. Процесс верстки состоит в оформлении текста и задании условий взаимного расположения текста и иллюстраций. Целью верстки является создание оригинал-макета, пригодного для размножения документа полиграфическим способом.

Существуют различные программные системы, среди которых можно выделить следующие:

1. *AdobeInDesign* - недавно появившийся пакет фирмы Adobe, оптимизированный под верстку документов самого широкого профиля, от одностраничных буклетов до толстых книг, обогащенный набором специфических визуальных инструментов.

2. *AdobePageMarker*- еще один пакет фирмы Adobe, с довольно сложным интерфейсом и системой команд, но в то же время с высокой производительностью и богатыми возможностями, особенно при работе с цветом.

3. *CorelVenturaPublisher* - альтернативный пакет фирмы Corel, несколько утративший в последнее время свои позиции, но вследствие своей универсальности (имеет широкие функции обычных текстовых и графических редакторов, интеграция с Web, поддержка различных платформ) не потерявший актуальности.

4. *QuarkXPress* - достаточно легкая в освоении и гибкая издательская система, которая традиционно используется многими издательствами газет, журналов, рекламными агентствами.

***Системы автоматизации бухгалтерской деятельности***

К этому виду систем относятся как простейшие программы для подготовки бухгалтерских документов и отчетности, так и очень сложные распределенные системы комплексного бухгалтерского учета *1C, Парус, Галактика, Инфо-бухгалтер* и т.д.

Программа "*1С:Предприятие*" является универсальной системой автоматизации деятельности предприятия и может применяться на различных участках бухгалтерского учета.

Система содержит три основных компонента:

1. бухгалтерский учет, отражающий финансовые операции;

2. оперативный учет, предназначенный для автоматизированного учета движения и остатков товарных, материальных, денежных средств предприятия;

3. расчет, позволяющий выполнять периодические расчеты различной сложности.

***Прочее ППО***

Можно перечислить еще множество прикладных программ как общего, как и специального назначения. В их числе:

* *Браузеры (browsers)* - средства просмотра гипертекста, используемые для отображения информации, получаемой с веб-сайтов ––*MS InternetExplorer, FireFox, Opera*;
* Клиенты электронной почты - *MS OutlookExpress, TheBat*;
* Аудио/ видеопроигрыватели - *Winamp, DivX Player, Real Player, Windows Media Рlayer*;
* Справочно-правовые системы - *Консультант Плюс, Гарант, Кодекс*;
* Офис-менеджеры (органайзеры) - *MS Outlook*;
* ПО для инженерных расчетов - *CAD/CAM/CAE AutoCAD*.