



Государственное образовательное учреждение  
высшего профессионального образования  
Санкт-Петербургский  
государственный технологический институт  
(технический университет)



**ДОБРО ПОЖАЛОВАТЬ**



**Кафедра системного анализа**

# Основные структуры данных

Существует три основных типа структур данных:

- *Линейная*
- *Табличная*
- *Иерархическая*

*Линейные структуры (списки)* - это упорядоченная структура данных, в которой адрес элемента данных однозначно определяется своим номером в массиве. (список студентов группы, все студенты группы зарегистрированы под своим уникальным номером)

*Табличные структуры (матрицы)* – это упорядоченные структуры, в которых адрес элемента определяется номером строки и номером столбца, на пересечении которых находится ячейка, содержащая искомый элемент.

В *иерархической структуре* адрес каждого элемента определяется путем доступа (маршрутом), ведущим от вершины структуры к данному элементу.

*Пуск → Программы → Стандартные → Калькулятор*

Хранение файлов организуется в иерархической структуре, которая называется *файловой структурой*. Файловая система включает в себя помимо самих файлов, правила образования имен и способов обращения к ним, иерархическую систему оглавления файлов и структуру хранения файлов на дисках. В качестве вершины файловой структуры служит имя носителя на котором сохраняются файлы.

Полное имя файла считается собственное имя файла с путем доступа к нему.

Пример: *C:\\1 курс\группа 321\Список студентов.doc*

## Технические средства

**Компьютер** – многофункциональное устройство предназначенное для обработки и хранения информации.

В основу построения большинства компьютеров положены общие принципы, сформулированные в 1945 г. американским ученым *Джоном фон Нейманом*.

- 1. Принцип программного управления.** Из него следует, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.
- 2. Принцип однородности памяти.** Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

**3. Принцип адресности.** Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

При рассмотрении компьютерных устройств принято различать их *архитектуру* и *структуру*.

**Архитектурой компьютера** называется его логическая организация, структура и ресурсы, то есть средства компьютерной системы, которые участвуют в процессе обработки данных.

**Структура компьютера** — это совокупность его функциональных элементов и связей между ними.

Архитектура современных компьютеров основана на **магистрально-модульном** принципе и является *открытой*. Модульный принцип позволяет потребителю подобрать нужную ему конфигурацию компьютера и при необходимости производить его модернизацию

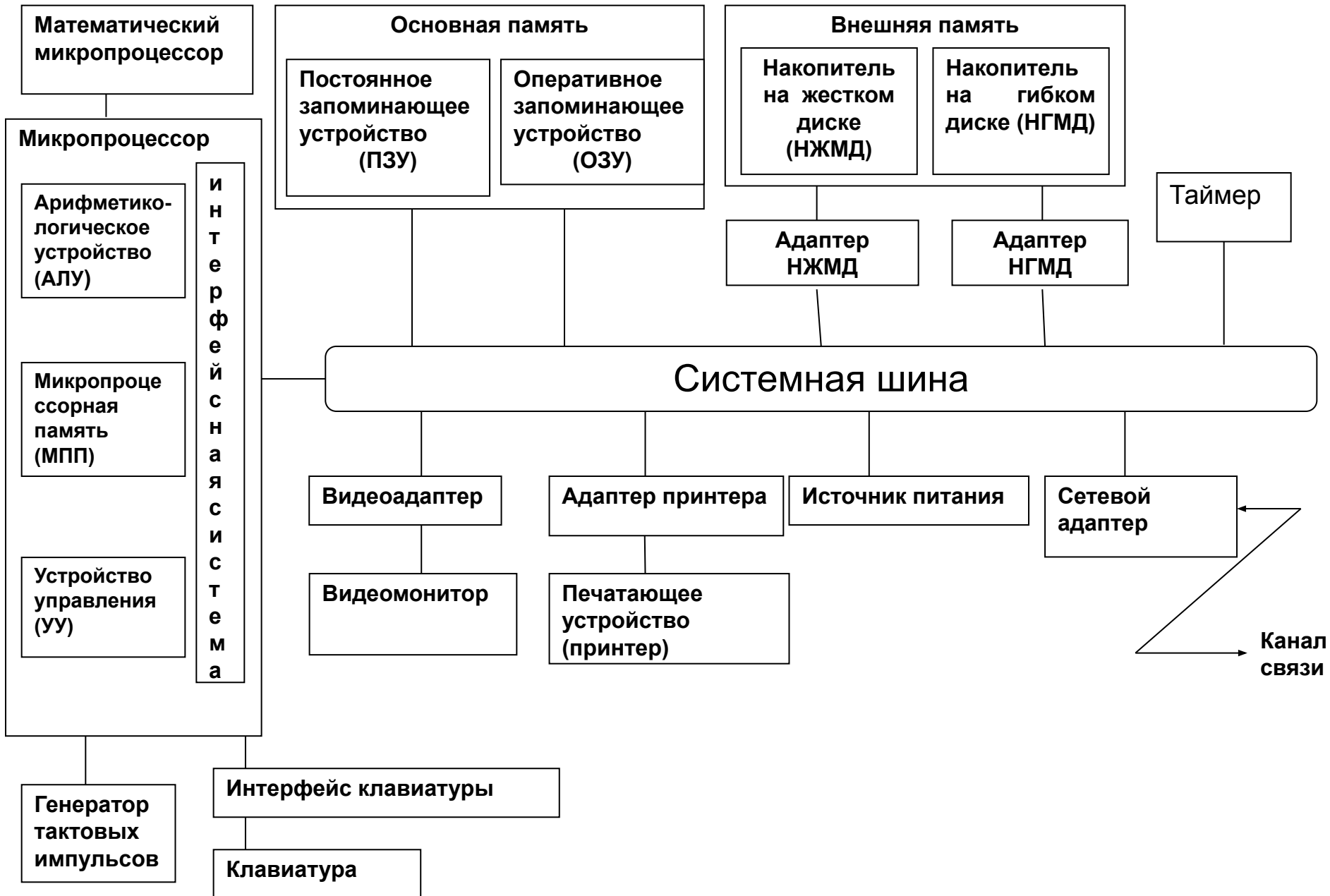
Модульная организация системы опирается на **магистральный (шинный)** принцип обмена информации. Магистральная или системная шина – набор электрических линий связывающих воедино по адресации памяти, передачи данных и служебных сигналов процессор, память и периферийные устройства.

Обмен информацией между отдельными устройствами производится по трем многозарядным шинам, соединяющим все модули – **шине данных, шине адресов, шине управления.**

Подключение отдельных модулей компьютера к магистралям на физическом уровне осуществляется с помощью **контроллеров**, а на программном осуществляется **драйверами.**



# Структурная схема персонального компьютера



## Программное обеспечение

*Программное обеспечение компьютера* – совокупность программ, позволяющих осуществлять на компьютере автоматическую обработку информации.

Программное обеспечение делится на:

- ***Системное (базовое)***
- ***Прикладное (специальное)***
- ***Инструментарий технологии программирования***

*Системное* – минимальный набор программных средств, обеспечивающих работу компьютера.

*Прикладное* программное обеспечение предназначено для выполнения конкретных задач пользователя.



# Системное программное обеспечение



```
graph TD; A[Системное программное обеспечение] --> B[Базовое обеспечение]; A --> C[Сервисное программное обеспечение]; B --> D[Операционные системы]; B --> E[Операционные оболочки]; B --> F[Сетевая операционная система]; C --> G[Программы диагностики работоспособности компьютера]; C --> H[Антивирусные программы]; C --> I[Программы обслуживания дисков]; C --> J[Программы архивирования данных]; C --> K[Программы обслуживания сети];
```

*Базовое обеспечение*

**Операционные системы**

**Операционные оболочки**

**Сетевая операционная система**

*Сервисное программное обеспечение*

**Программы диагностики работоспособности компьютера**

**Антивирусные программы**

**Программы обслуживания дисков**

**Программы архивирования данных**

**Программы обслуживания сети**

**Операционная система** - это программа, которая загружается при включении компьютера. Операционная система предназначена для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами ЭВМ.

Операционная система осуществляет:

- *диалог с пользователем;*
- *управление компьютером, его ресурсами (оперативной памятью, местом на дисках и т.д.);*
- *запускает другие (прикладные) программы на выполнение.*

Операционная система обеспечивает пользователю и прикладным программам удобный способ общения (интерфейс) с устройствами компьютера.

## Инструментарий технологии программирования

Это направление связано с технологией создания программных продуктов. Сюда можно отнести средства создания приложений, которые включают в себя совокупность языков и систем программирования, а также различные программные комплексы для отладки и поддержки создаваемых программ.

Языки программирования делятся на языки *структурного программирования* (ТВ, ТР) и *объектно-ориентированные*, использующие понятия *объектов*, *свойств* и *методов* (VB, C<sup>++</sup>). Программа, подготовленная на языке программирования, проходит этап трансляции, т.е. преобразования в программу, состоящую из машинных команд. Для этой цели были разработаны программы-переводчики (трансляторы).

В зависимости от функционального назначения трансляторы подразделяются на интерпретаторы и компиляторы.

Интерпретатор – это транслятор, анализирующий каждую конструкцию исходной программы в отдельности и сразу ее выполняющий. Программы хранятся в том виде, в каком они хранятся на бумаге. С такой программой пользователю легко работать.

Компилятор – это транслятор, обеспечивающий перевод программы на язык, близкий к машинному, но он ее не выполняет. До выполнения программа должна пройти этап редактирования, для поиска и исправления ошибок. Другими словами компилятор не выполняет программы, а он их строит. VB одновременно является и *интерпретатором*, и *компилятором*.

## Основные понятия объектно-ориентированного проектирования

Объектно-ориентированный подход использует следующие базовые понятия:

- **объект;**
- **свойство объекта;**
- **метод;**
- **событие;**
- **класс объектов.**

**Объект** – некая сущность, которая четко проявляет свое поведение и является представителем некоторого класса подобных себе объектов.

Объект содержит **инструкции** (программный код), определяющие действия, которые может выполнять объект, и обрабатываемые **данные**.

Объекты могут содержать в своем описании другие объекты, тогда говорят, что он является **контейнером**.

**Свойство** – характеристика объекта, его параметр. Все объекты наделены определенными свойствами, которые в совокупности выделяют объект из множества других объектов.

*Например, свойствами объекта «Командная кнопка» являются высота, ширина, цвет и т.д.*

Значения свойств можно задавать на стадии проектирования в окне свойств (Properties Windows). При этом нет необходимости описывать все свойства каждого объекта: в большинстве случаев используются значения свойств установленные по умолчанию. Чтобы изменить характеристики объекта, нужно изменить значение его свойства.

Синтаксис установки значения свойства:

**Объект.Свойство=ЗначениеСвойства**

*Например:* Form1.BackColor=vbRed

**Метод** – программа действия над объектом или его свойствами.

Синтаксис применения метода:

### **Объект.Метод**

Метод можно применить ко всем объектам семейства. Методы объекта определяют те задачи, которые может выполнить данный объект. Например, объект **Форма** обладает методом PRINT. Например, чтобы отобразить на Форме строку «ПРИВЕТ», можно написать следующий код:

***Форма1.Print "Привет".***

**Событие** – это действие, распознаваемое объектом (например, щелчок мыши или нажатие клавиши), для которого можно запрограммировать **отклик**.

События происходят в результате действий пользователя или программы, или же они могут быть вызваны системой.

Суть программирования на VB заключается в этих двух понятиях: *событие* и *отклик* на него. Если пользователь производит какое-то воздействие на систему, например, нажимает кнопку, тогда в качестве отклика выполняется код созданной пользователем процедуры. Если такой отклик не создан, т.е. не написана соответствующая процедура, то система никак не реагирует на данное событие, и оно остается без ответа.

Таким образом, действия, происходящие в системе, являются *событиями*, а отклики на них – *процедурами*. Этот специальный вид процедур, генерирующих отклик на события, называется *процедурами обработки событий*. В целом программирование на VB состоит в создании кода программ, которые генерируют прямо или косвенно отклики на события.



Объекты могут объединяться в *классы*. К одному классу принадлежат объекты с одинаковым набором свойств, методов и событий.

**Класс** – совокупность объектов, характеризующихся общностью применяемых методов обработки или свойств.

Каждый *конкретный* объект, имеющий структуру этого класса, называется *экземпляром* класса.

Примером *класса* может служить:

- **класс студентов**. Все студенты сдают экзамены. Каждый студент (экземпляр класса) сдает экзамены;
- **класс Командная кнопка** – общее описание кнопок в окнах приложения. Эти кнопки могут быть всех цветов и размеров, но должны иметь множество *общих свойств* и других характеристик, например, *событий*, которые для этих объектов одинаковы (например, щелчок по ним мышью)