



# **СТРУКТУРЫ ДАННЫХ**

**Лекция №2**

**09.02.03**

**2 курс**

# ВВЕДЕНИЕ

*Они служат базовыми элементами  
любой машинной программы.*

*В организации структур данных и  
процедур их обработки заложена  
возможность проверки правильности  
работы программы.*

*Никлас Вирт*



# ПОНЯТИЕ СТРУКТУР ДАННЫХ И АЛГОРИТМОВ

В основе работы всякого компьютера лежит умение оперировать только с одним видом данных - с отдельными битами, или двоичными цифрами.

Компьютер работает с этими данными только в соответствии с неизменным набором алгоритмов, которые определяются системой команд центрального процессора.



# ПОНЯТИЕ СТРУКТУР ДАННЫХ И АЛГОРИТМОВ

Для точного описания абстрактных структур данных и алгоритмов программ используются такие системы формальных обозначений, называемые языками программирования, в которых смысл всякого предложения определяется точно и однозначно.



# ПОНЯТИЕ СТРУКТУР ДАННЫХ И АЛГОРИТМОВ

- Среди средств, представляемых почти всеми языками программирования, имеется возможность ссылаться на элемент данных, пользуясь присвоенным ему именем, идентификатором.
- Одни именованные величины являются константами, которые сохраняют постоянное значение в той части программы, где они определены, другие - переменными, которым с помощью оператора в программе может быть присвоено любое новое значение.
- До тех пор, пока программа не начала выполняться, их значение не определено.



# Типы данных

- Для компьютера все типы данных сводятся в к последовательности битов, поэтому различие в типах следует делать явным.
- Типы данных, принятые в языках программирования, включают натуральные и целые числа, вещественные (действительные) числа (в виде приближенных десятичных дробей), литеры, строки и т.п.



## ЗАЩИТА ТИПОВ

- язык PASCAL, изначально являвшийся, прежде всего, инструментом для иллюстрирования структур данных и алгоритмов, сохраняет от своего первоначального назначения весьма строгую защиту типов.
- PASCAL-компилятор в большинстве случаев расценивает смешение в одном выражении данных разных типов или применение к типу данных несвойственных ему операций как фатальную ошибку.



# ЗАЩИТА ТИПОВ

- язык С, предназначенный прежде всего для системного программирования, является языком с весьма слабой защитой типов.
- С-компиляторы в таких случаях лишь выдают предупреждения.
- Отсутствие жесткой защиты типов дает системному программисту, разрабатывающему программу на языке С, дополнительные возможности, но такой программист сам отвечает за правильность своих действий.





# ПОНЯТИЕ СТРУКТУР ДАННЫХ И АЛГОРИТМОВ

Структуру данных можно свести к *схеме* организации информации в памяти компьютера.

Алгоритм является соответствующим процедурным элементом в структуре программы - он служит *рецептом* расчета.



# ХРАНЕНИЕ ИНФОРМАЦИИ

В цифровых вычислительных машинах можно выделить три основных вида запоминающих устройств:

- сверхоперативная,
- оперативная,
- внешняя память.

Обычно сверхоперативная память строится на регистрах.

Регистры используются для временного хранения и преобразования информации.



# ХРАНЕНИЕ ИНФОРМАЦИИ

- Центральный процессор содержит регистры (иногда называемые аккумуляторами), в которые помещаются аргументы (т.е. операнды) арифметических операций.
- С целью проверки необходимости изменения нормальной последовательности передач управления в аккумуляторах могут анализироваться отдельные биты.
- Регистры используются также для временного хранения команд программы и управляющей информации о номере следующей выполняемой команды.



# ХРАНЕНИЕ ИНФОРМАЦИИ

- Оперативная память предназначена для запоминания более постоянной информации.
- Важнейшим свойством оперативной памяти является адресуемость. Каждая ячейка памяти имеет свой идентификатор, однозначно идентифицирующий ее в общем массиве ячеек памяти (адрес).
- Адреса ячеек являются операндами тех машинных команд, которые обращаются к оперативной памяти. В подавляющем большинстве современных вычислительных систем единицей адресации является байт .
- Определенная ячейка оперативной памяти или множество ячеек могут быть связаны с конкретной переменной в программе.



# ХРАНЕНИЕ ИНФОРМАЦИИ

- Внешняя память служит прежде всего для долговременного хранения данных.
- Характерным для данных на внешней памяти является то, что они могут сохраняться там даже после завершения создавшей их программы и могут быть впоследствии многократно использованы той же программой при повторных ее запусках или другими программами.



Под СТРУКТУРОЙ ДАННЫХ в общем случае понимают множество элементов данных и множество связей между ними.

Понятие "ФИЗИЧЕСКАЯ структура данных" отражает способ физического представления данных в памяти машины и называется еще структурой хранения, внутренней структурой или структурой памяти. Рассмотрение структуры данных без учета ее представления в машинной памяти называется абстрактной или ЛОГИЧЕСКОЙ структурой.



# СТРУКТУРЫ ДАННЫХ

- ▣ ПРОСТЫЕ (базовые, примитивные) структуры (типы) данных
- ▣ ИНТЕГРИРОВАННЫЕ (структурированные, композитные, сложные).
- ▣ *Простыми* называются такие структуры данных, которые не могут быть расчленены на составные части, большие, чем биты.
- ▣ *Интегрированными* называются такие структуры данных, составными частями которых являются другие структуры данных - простые или в свою очередь интегрированные.



В зависимости от отсутствия или наличия явно заданных связей между элементами данных следует различать:

НЕСВЯЗНЫЕ структуры (векторы, массивы, строки, стеки, очереди)

СВЯЗНЫЕ структуры (связные списки).





Первый признак структуры данных — ее ***изменчивость*** - изменение числа элементов и (или) связей между элементами структуры.

В определении изменчивости структуры *не отражен* факт изменения значений элементов данных, поскольку в этом случае все структуры данных имели бы свойство изменчивости.

По признаку изменчивости различают структуры  
СТАТИЧЕСКИЕ, ПОЛУСТАТИЧЕСКИЕ,  
ДИНАМИЧЕСКИЕ.



# КЛАССИФИКАЦИЯ СТРУКТУР ДАННЫХ



Второй признак структуры данных –  
*характер упорядоченности* ее элементов.

По этому признаку структуры можно делить на  
ЛИНЕЙНЫЕ И НЕЛИНЕЙНЫЕ структуры.



В зависимости от характера взаимного расположения элементов в памяти линейные структуры можно разделить на структуры

с ПОСЛЕДОВАТЕЛЬНЫМ распределением элементов в памяти (векторы, строки, массивы, стеки, очереди)

с ПРОИЗВОЛЬНЫМ СВЯЗНЫМ распределением элементов в памяти ( односвязные, двусвязные списки).

Пример нелинейных структур - многосвязные списки, деревья, графы.



В языках программирования понятие "структуры данных" тесно связано с понятием "типы данных".

Любые данные, т.е. константы, переменные, значения функций или выражения, характеризуются своими типами.



Информация по каждому типу однозначно определяет :

- 1) структуру хранения данных указанного типа, т.е. выделение памяти и представление данных в ней, с одной стороны, и интерпретирование двоичного представления, с другой;
- 2) множество допустимых значений, которые может иметь тот или иной объект описываемого типа;
- 3) множество допустимых операций, которые применимы к объекту описываемого типа.



# ОПЕРАЦИИ НАД СТРУКТУРАМИ ДАННЫХ

Над любыми структурами данных могут выполняться четыре общие операции:

- ▣ **Создание** (выделении памяти для структуры данных), память может выделяться в процессе выполнения программы или на этапе компиляции либо при активизации процедурного блока, в котором объявляются соответствующие переменные.
- ▣ **Уничтожение** (противоположна по своему действию операции создания),
- ▣ **Выбор** (доступ) (используется программистами для доступа к данным внутри самой структуры),
- ▣ **Обновление** (позволяет изменить значения данных в структуре данных).



Помимо этих общих операций для каждой структуры данных могут быть определены операции *специфические*, работающие только с данными данного типа (данной структуры).

Специфические операции рассматриваются при рассмотрении каждой конкретной структуры данных.

