

# Основные понятия и определения

- **Компьютер на аппаратном уровне** – совокупность технических средств для автоматизированной обработки дискретных данных по заданному алгоритму.
- **Алгоритм** — конечный упорядоченный набор четко определенных правил для решения проблемы  
[Международная организация стандартов ISO 2382/1-93]
- **Алгоритм** — это точное предписание исполнителю, определяющее содержание и порядок действий, которые необходимо выполнить над исходными и промежуточными данными для получения конечного результата.

# Принцип программного управления

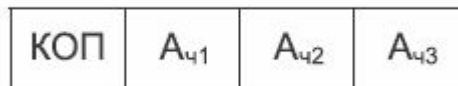
- Вычислительная машина является исполнителем алгоритмов → свойства алгоритмов определяют ее организацию.
- Универсальную формулировку принципа программного управления предложил американский ученый Джон фон Нейман (1945):
  - Обрабатываемая информация кодируется двоичными цифрами (0, 1) и разделяется на единицы, называемые словами.

# Принцип программного управления

- ...принципы программного управления Джона фон Неймана:
  - Алгоритм вычислений представляется в виртуальной машине в машинной форме — в форме программы, состоящей из последовательности *команд*. Команды тоже записываются в двоичном виде. Каждая команда предписывает некоторую операцию (из набора операций вычислительной машины) и указывает слова данных (числа), над которыми ее нужно выполнить.

# Принцип программного управления

- ...принципы программного управления Джона фон Неймана:
  - Операция задается значением кода операции КОП, а числа — адресами ячеек памяти  $A_{ч1}$ , в которых они хранятся. Адрес  $A_{ч}$  является машинным именем числа Ч.
    - Адрес — единственное средство, с помощью которого можно найти нужное число в памяти.
    - В типовой команде  $A_{ч1}$ ,  $A_{ч2}$  обозначают адреса аргументов, а  $A_{ч3}$  — адрес результата операции.



# Понятие машинного языка

- В совокупности команды аппаратного процессора составляют встроенный ***машинный язык*** (МЯ).
- Состав команд машинного языка зависит от назначения компьютера:
  - команды пересылки данных,
  - арифметической и логической обработки (сложение, вычитание, умножение и деление),
  - ввода/вывода
  - управления потоком команд.

# Языки высокого уровня

- **Язык высокого уровня (ЯВУ)** - команды, более удобные для человека, чем машинные команды
- ЯВУ аппаратный компьютер напрямую «не понимает»!
- Существует два способа решения проблемы, их основная цель - заменять высокоуровневые команды эквивалентными наборами машинных команд

# Языки высокого уровня и трансляция

- **Трансляция** - преобразование одной программы, написанной на ЯВУ, в другую программу, записываемую в терминах машинного языка.
- Способ преобразования: каждая команда из ЯВУ заменяется на эквивалентный набор команд из машинного языка.

# Языки высокого уровня и интерпретация

- **Интерпретация** - создание программы на машинном языке, которая поочередно обрабатывает каждую команду программы на языке высокого уровня и заменяет каждую команду ЯВУ на эквивалентный набор машинных команд и сразу же выполняет этот набор.
- Программу, выполняющую этот процесс, называют *интерпретатором*.



# Языки высокого уровня и виртуальная машина

- ***Виртуальная машина*** — это программная надстройка над аппаратным компьютером.
- В качестве машинного языка здесь выступает язык высокого уровня.
- Всю работу по-прежнему будет выполнять аппаратный компьютер, обеспечиваемый транслятором или интерпретатором, но теперь это «остаётся за кадром».

# Трансляторы и интерпретаторы

- ***Транслятором*** называют программное приложение, которое в качестве входных данных воспринимает программы на некотором *исходном языке*, а на выходе формирует эквивалентные по своей функциональности программы, но уже на другом *объектном языке*. Как исходный язык, так и объектный язык может быть высокого или низкого уровня.

# Трансляторы и интерпретаторы.

## Наиболее популярные разновидности

- **Ассемблер** — транслятор, у которого объектным языком является некоторая разновидность машинного языка какого-либо аппаратного компьютера, а исходным языком — символическое представление машинного языка.
- Исходный язык обычно называют **ЯЗЫКОМ ассемблера**.
- Чаще всего каждая команда на исходном языке переводится в одну команду на объектном языке.

# Трансляторы и интерпретаторы.

## Наиболее популярные разновидности

- **Компилятор** — это транслятор, для которого исходным считается язык высокого уровня.
- Объектный язык очень близок к машинному языку аппаратного компьютера — им является либо язык ассемблера, либо какой-нибудь вариант машинного языка.
- Например, программы на языке C обычно компилируются в программы на языке ассемблера, которые ассемблер затем переводит в машинный язык.

# Трансляторы и интерпретаторы. Наиболее популярные разновидности

Характерные недостатки механизма трансляции:

- потеря информации об исходной программе на языке высокого уровня. Если при выполнении объектной формы программы появляется ошибка, трудно выявить высокоуровневый оператор, являющийся ее источником;
- существенный рост размера объектной формы программы, поскольку оператор на языке высокого уровня содержит гораздо больше информации, чем команда машинного языка.

# Трансляторы и интерпретаторы.

## Наиболее популярные разновидности

- **Программная интерпретация** - создается виртуальная машина — виртуальный компьютер, для которого машинным языком будет некоторый язык высокого уровня.
- Виртуальный компьютер — это аппаратный компьютер + набор программ на машинном языке, которые моделируют алгоритмы и структуры данных, необходимые для выполнения программ на языке высокого уровня.

# Трансляторы и интерпретаторы.

## Наиболее популярные разновидности

### **Преимущества трансляции:**

- операторы программы остаются в своей исходной форме до тех пор, пока они не понадобятся при выполнении.
- Не расходуются память на хранение нескольких копий длинной цепочки машинных команд; все необходимые команды достаточно сохранить в программе-интерпретаторе лишь один раз.

### **Недостатки интерпретации:**

- необходимость многократного декодирования одного и того же оператора цикла (или вызова подпрограмм). Для минимизации затрат к таким операторам следует применять механизм трансляции.

# Трансляторы и интерпретаторы.

## Наиболее популярные разновидности

- **Объектно-трансляция и интерпретация** применяют совместно, взаимно компенсируя недостатки друг друга.
- **Комбинированное решение:**
  - независимые части программы транслируются в объектные формы (этап трансляции)
  - независимые части объединяются с набором подпрограмм поддержки выполнения. Подпрограммы поддержки обеспечивают программные реализации специфических операций, объединение с ними формирует выполняемую форму программы.
  - операторы этой формы декодируются и интерпретируются на этапе выполнения.



# Трансляторы и интерпретаторы. Комбинированное решение

