

Введение в программирование

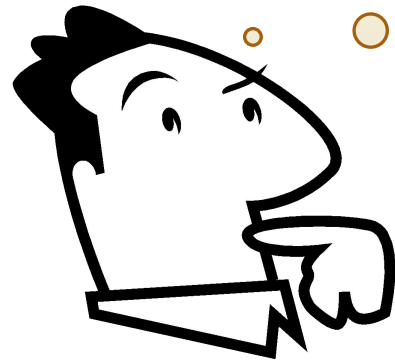
ЗАНЯТИЕ №1

Что же это?



В широком смысле - процесс создания компьютерных программ.
В узком смысле – написание инструкция на языке программирования.

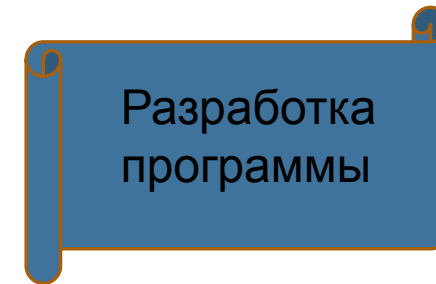
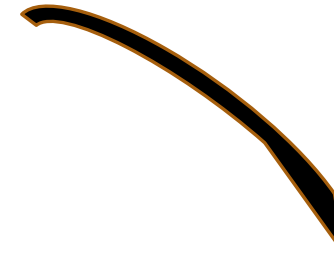
Программирование для ЭВМ основывается на использовании языков программирования, на которых записывается программа.



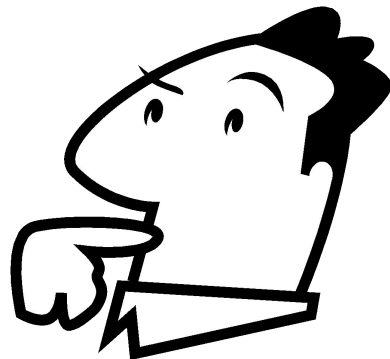
Постановка задачи



Поиск алгоритма
решения



Разработка
программы



Анализ результатов



Понятие алгоритма и программы

Алгоритм – описание последовательности действий над некоторыми объектами (данными), ведущее к цели за определенное время.

Программа – это представление алгоритма и обрабатываемых им данных в виде, предназначенном для выполнения компьютером.

Представления программ

Программа создается человеком, а исполняется компьютером

Поэтому она должна иметь, по крайней мере, два представления:

- *внешнее* - в виде текста, понятного человеку
- *внутреннее* - в виде двоичного кода, с которым работает компьютер

Языки программирования

Язык программирования – это средство для текстового представления программы

Языки программирования делятся на

- *языки высокого уровня*
- *языки низкого уровня (ассемблер)*

Языки высокого уровня

В языках высокого уровня используются способы записи, близкие к средствам естественного языка (обычно, английского)

Язык высокого уровня может быть ориентирован на тот или иной класс задач, либо быть универсальным

Языки низкого уровня

Языки низкого уровня позволяют записать текст программы в виде последовательности машинных команд

Язык низкого уровня ориентирован на конкретный тип процессора; для каждого типа процессоров существует свой язык ассемблера

Языки высокого уровня

Языки высокого уровня относятся к классу *формальных языков*.

Формальным называется искусственный язык, любой элемент которого может быть построен из точно определенных начальных понятий с помощью заданных правил

Эти правила делятся на *синтаксические* (*синтаксис языка*) и *семантические* (*семантика языка*)

Синтаксис и семантика

Синтаксическими называются правила построения отдельных языковых конструкций – слов и предложений – из набора основных символов, образующих *алфавит языка*

Семантические правила показывают смысловое содержание текста

Структура программы

Программа как текст на языке высокого уровня обладает определенной *синтаксической структурой*, в рамках которой можно выделить

- *информационную,*
- *управляющую*
- *и контролирующую составляющие*

Информационная часть структуры

Информационная составляющая программы представлена обрабатываемыми ею данными – *константами* и *переменными*

Для каждой переменной задается *имя*, а в процессе выполнения программы она может получать различные *значения*.

Для константы задается значение, которое не может быть изменено в процессе выполнения программы; кроме того, константа может иметь имя

Значения, которых хранят переменные, определяются *типами данных*

Управляющая часть структуры

Управляющей частью структуры программы является описание действий над данными

Действия над данными, связанные с получением и изменением значений переменных, называются *операциями*

Операции

Для обозначения операций в языках программирования используются *специальные символы*, например $+$, $-$, $*$, $/$ и т.д.

Данные, над которыми выполняется операция, называются ее *операндами*, а полученное значение - *результатом операции*. Примерами операций являются арифметические действия над числами

Операторы

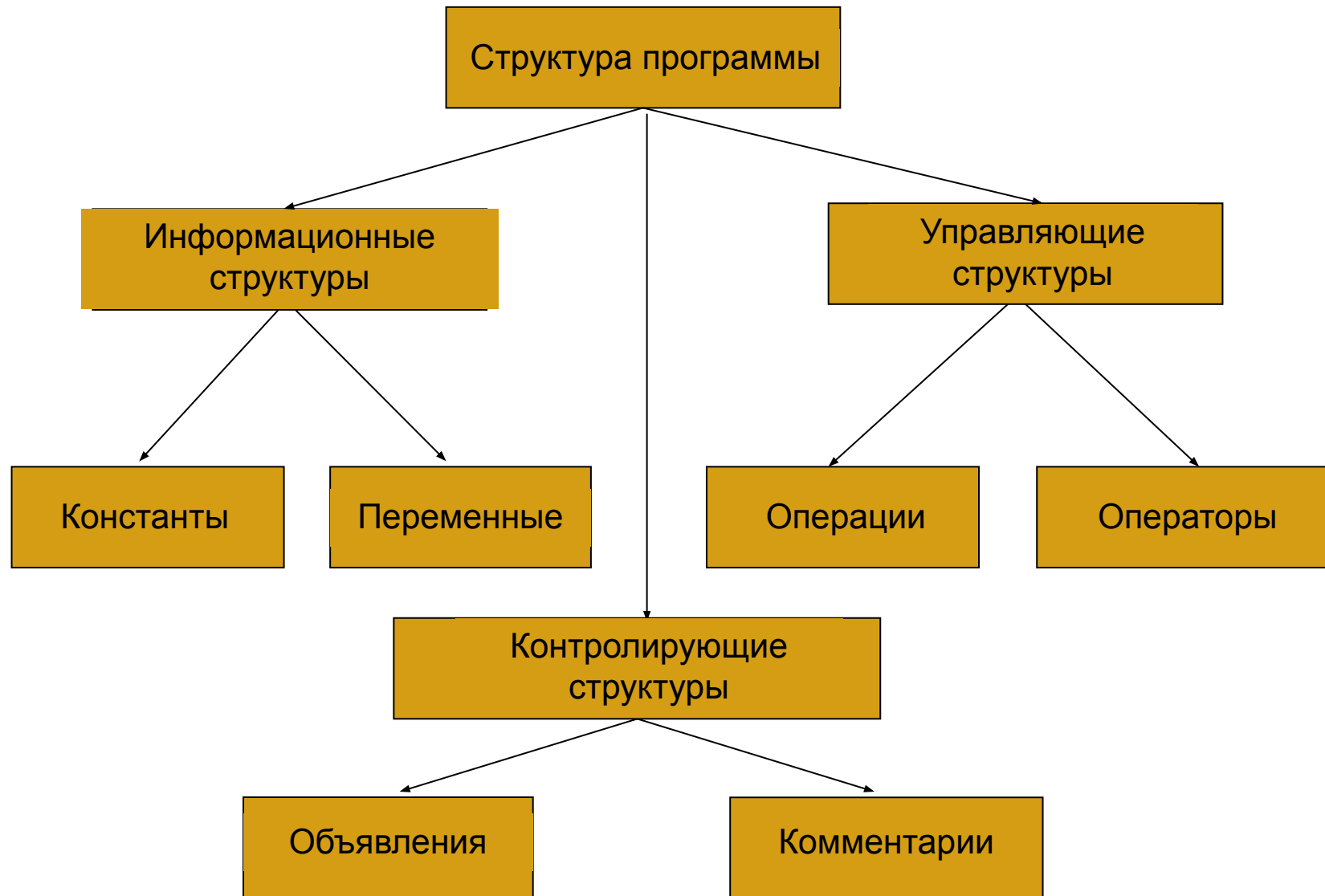
Более сложные действия описываются *операторами языка программирования*, имеющими свой синтаксис и семантику

Составной частью операторов могут быть *выражения*, каждое из которых представляет собой описание некоторой последовательности операций над значениями констант и переменных

После выполнения этих действий получается *значение выражения*

Контролирующая часть структуры

Контролирующая часть структуры программы содержит описания свойств констант и переменных, а также структурообразующие предложения в виде различного рода заголовков, специальных скобок и т.д.



Трансляция программы

Преобразование программы из внешнего представления во внутреннее называется ее *трансляцией*

Трансляция осуществляется специальными программами – трансляторами, которые делятся на

- *компиляторы*
- *интерпретаторы*

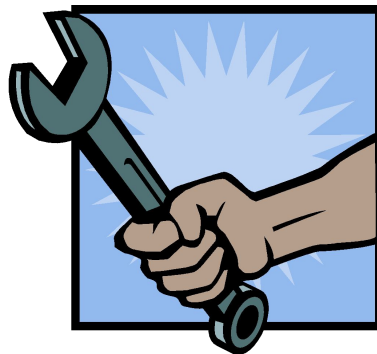
```
class Program
{
    static void Main()
    {
        Console.WriteLine("Hello, World!");
        ...
    }
}
```

Трансляция
→
Программа-транслятор

```
10011100
10101010
00100111
11111010
00100011
11111010
10000011
```

Инструментальные средства

Программы-трансляторы представляют собой пример *инструментальных средств*, т.е. программного обеспечения, предназначенного для разработки других программ



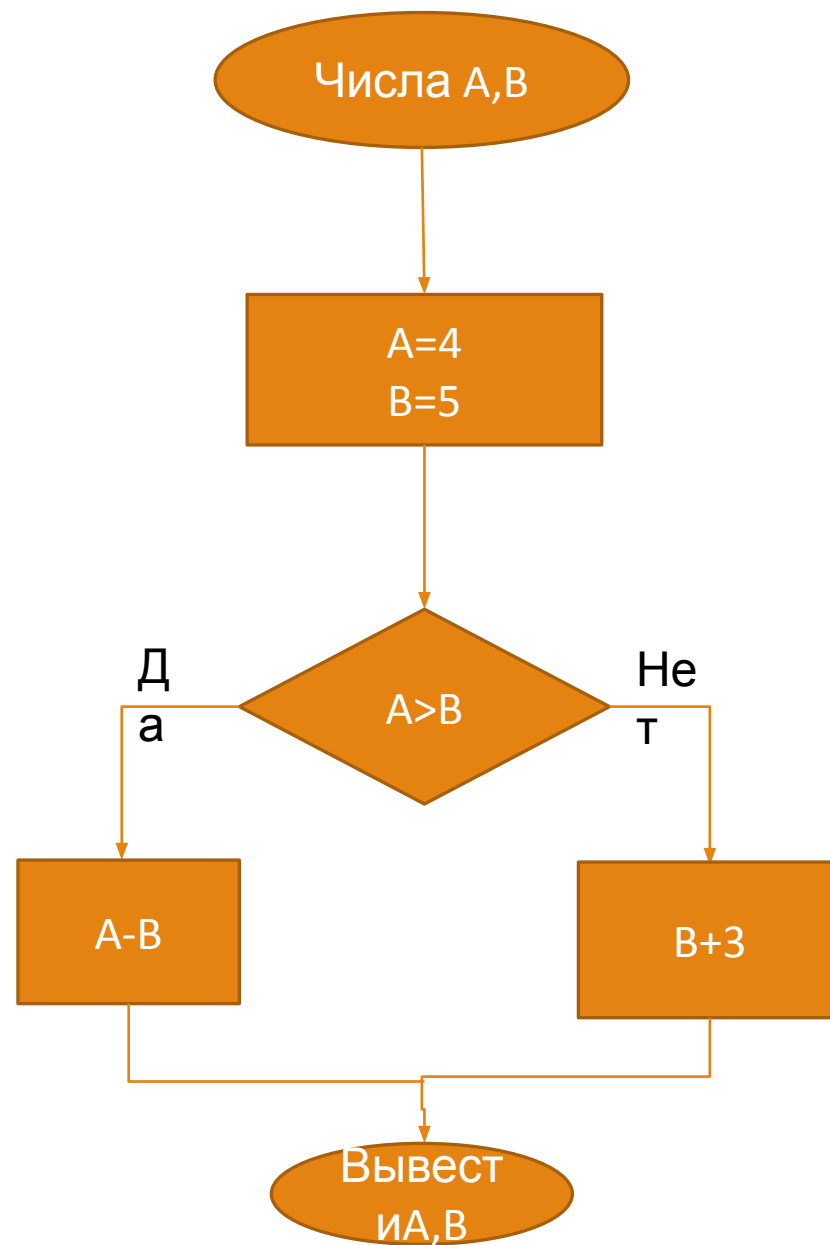
Стадии программирования

1. Анализ
2. Проектирование — разработка комплекса алгоритмов
3. Кодирование и компиляцию — написание исходного текста программы и преобразование его в исполнимый код с помощью компилятора
4. Тестирование и отладку — выявление и устранение ошибок в программах
5. Испытания и сдачу программ

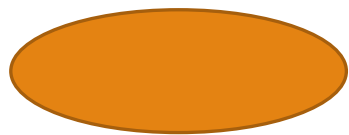
Блок-схема

Схэма — графическое представление определения, анализа или метода решения задачи, в котором используются символы для отображения данных

Блок-схема — распространенный тип, описывающих алгоритмы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление выполнения действий.



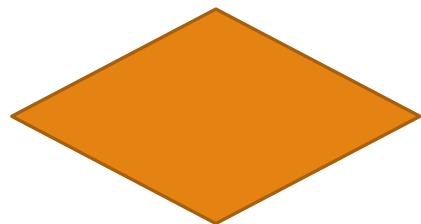
Основные элементы блок-схем



- Блок начало-конец: Показывает начало и конец алгоритма



- Блок действия: Служит для записи операции

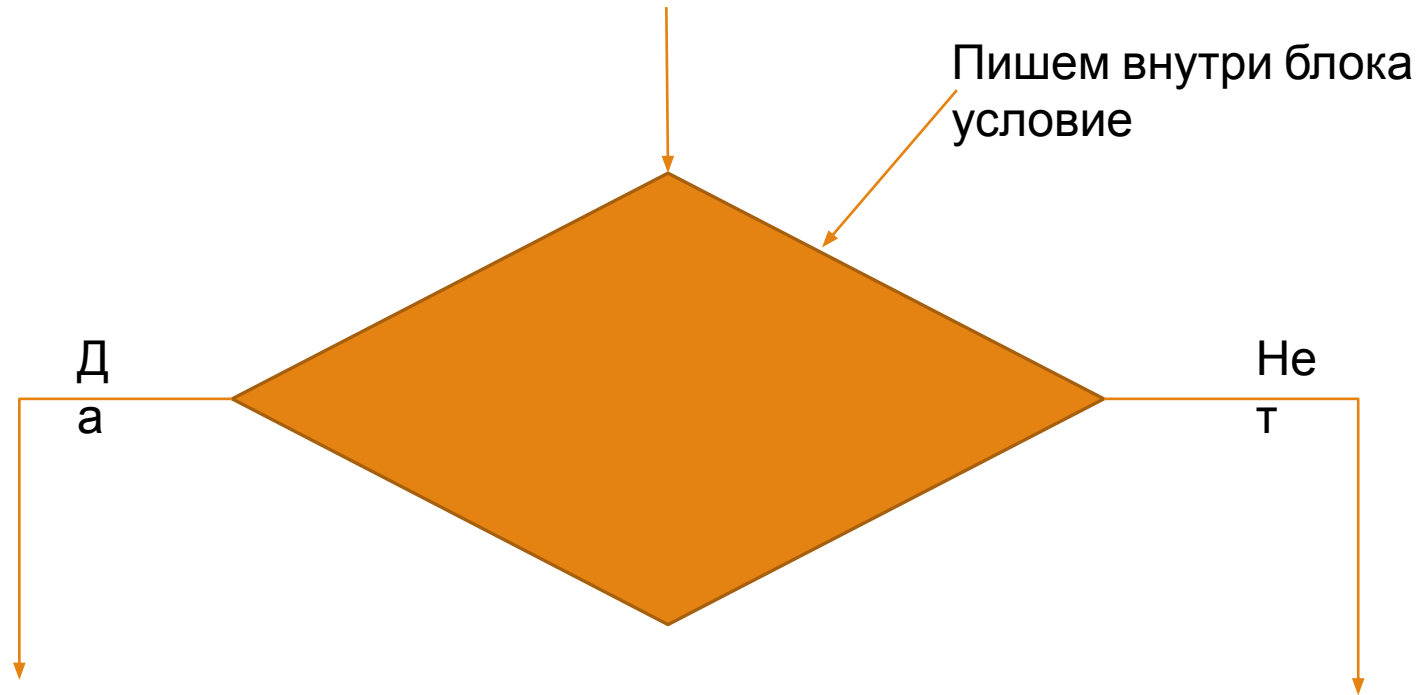


- Логический блок: Переключатель с одним входом и несколькими выходами

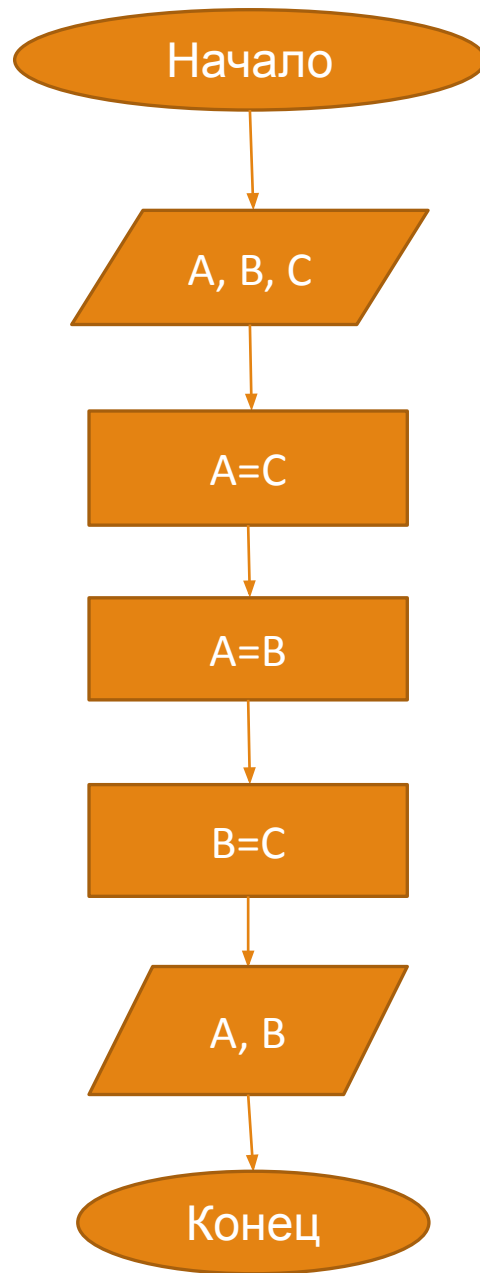


- Блок ввода-вывода: Обозначает показ данных

Логический блок



В зависимости от выполнения условия – следуем по одной из веток



Домашнее задание

Решить задачи. Решение изобразить в виде блок-схем.

1. Подсчитать, сколько очков набрала команда в первом круге чемпионата по хоккею, если известно, что m встреч она выиграла, n встреч проиграла, k встреч закончились ничьими. (Сумма очков: $m*2+k$)
2. Из трех чисел выбрать наименьшее.
3. Значения переменных X , Y , Z поменять местами так, чтобы они оказались упорядоченными по возрастанию. ($X>Y>Z$)