

*Система программирования* —  
это система для разработки новых  
программ на конкретном языке  
программирования.

# В систему программирования входят:

- компилятор или интерпретатор;
- интегрированная среда разработки;
- средства создания и редактирования текстов программ;
- библиотеки стандартных программ и функций;

- отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
- "дружественная" к пользователю диалоговая среда;
- многооконный режим работы;
- встроенная справочная служба;

# *Транслятор* (англ. *translator* — переводчик)

- — это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.
- Транслятор состоит:

- *Компилятор* (англ. *compiler* — составитель, собиратель) читает всю программу *целиком*, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

- *Интерпретатор* (англ. *interpreter* — истолкователь, устный переводчик) переводит и выполняет программу *строка за строкой*.

# системы программирования, ориентированные на создание *Windows-приложений*:

- пакет *Borland Delphi* (Дельфи) — наследник семейства компиляторов Borland Pascal, предоставляющий качественные и очень удобные средства визуальной разработки. Его компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.

- пакет *Microsoft Visual Basic* — удобный и популярный инструмент для создания Windows-программ с использованием визуальных средств. Содержит инструментарий для создания *диаграмм и презентаций*.



- пакет *Borland C++* — одно из самых распространённых средств для разработки DOS и Windows приложений.

Популярные системы  
программирования – *Turbo  
Basic, Quick Basic, Turbo  
Pascal, Turbo C.*

# Язык Бейсик

- был создан в 1965 г. Дж. Кемени и Т. Курцем как язык для начинающих, облегчающий написание простых программ.

# Язык Паскаль

- был разработан в 1970г. Никласом Виртом как язык обучения студентов программированию.  
Паскаль вырабатывает навыки соблюдения хорошего строгого стиля программирования (называемого структурным программированием), упрощающего разработку сложных программ.  
В своем первоначальном виде Паскаль имел довольно ограниченные возможности, но расширенный вариант этого языка – Turbo Pascal, является очень мощным языком программирования.

# Язык Си

- разработан Деннисом Ритчи в 1972 г. соединяет свойства языка высокого уровня с возможностями эффективного использования ресурсов компьютера, которые обычно достигаются только при программировании на языке Ассемблера.

Си не очень прост в изучении и требует тщательности в программировании, но позволяет создавать сложные и весьма эффективные программы.

для иллюстрации приведены на  
языках Бейсик, Паскаль и Си  
программы решения одной и той  
же простой задачи —  
вычисления суммы  $S$  элементов  
одномерного массива  $A=(a_1, a_2,$   
 $\dots, a_n)$ .

# *Программа на Бейсике*

- 

```
10 INPUT "N = "; N
20 DIM A(N)
30 FOR I = 1 TO N
40 PRINT "A("; I; ") =";
50 INPUT A(I)
60 NEXT I
70 S = 0
80 FOR I = 1 TO N
90 S = S + A(I)
100 NEXT I
110 PRINT "Сумма ="; S
```

# *Программа на Паскале*

- 

```
Program Summa;  
  Type Mas = Array [1..100] of Real;  
  Var A : Mas;  
      i, n: Integer;  
      S : Real;  
BEGIN  
  ReadLn(n);  
  For i := 1 To 10 Do  
    ReadLn(A[i]);  
  S := 0;  
  For i := 1 To 10 Do  
    S := S + A[i];  
  WriteLn('S = ', S:8:2);  
END.
```



# Програма на СИ

- 

```
# include <stdio.h>
main()
{
    float a[100],s;
    int i,n = 100;
    for (i = 0; i <= n; i++)
        get(a[i]);
    for (i = 0; i <= n; i++)
        s = s + a[i];
    printf("s = % f \ n", s);
}
```

# **Работа с интегрированной средой разработки (ИСР) ТУРБО ПАСКАЛЬ 7.0**

# Меню ИСЗ

- **File** - включает команды работы с файлами.
- **Edit** - включает команды редактирования текста
- **Search** - включает команды быстрого поиска компонент программы и контекстной замены фрагментов программы

- **Run** - включает команды запуска программы выполнения и трассировки её работы.
- **Compile** - включает команды, задающие различные режимы компиляции.
- **Debug** - включает команды управления отладочной информацией.

- **Tools** - включает команды работы с сообщениями ИСР и команды вызова отладочных утилит.
- **Options** - включает команды, обрабатывающие диалоговые окна для установки опций ИСР.
- **Window** - включает команды управления окнами ИСР.
- **Help** - включает команды вызова справочной информации по работе в ИСР и по языку Turbo Pascal.

# Как обнаружить ошибки?

- Проверяйте наличие точки с запятой в конце операторов;
- Проверяйте соответствие типа и размера используемых в программе переменных и массивов их описанию;
- Пишите операторы вывода промежуточных значений переменных, чтобы понять, где программа начала работать неверно;

- Пишите операторы вывода информации, локализующей место ошибки. Для проверки того, исполняется ли какая-то группа операторов, можно поставить среди них оператор `writeln('*****')`. Если при выполнении программы на экране нет звездочек, значит, эта группа операторов почему-то не выполняется.

- Если при отладке возникла новая ошибка, проверьте самое последнее изменение, которое Вы внесли в программу;



Использование операторов  
присваивания, read, write, writeln,  
описание переменных и констант  
различных типов,  
форматированный вывод.

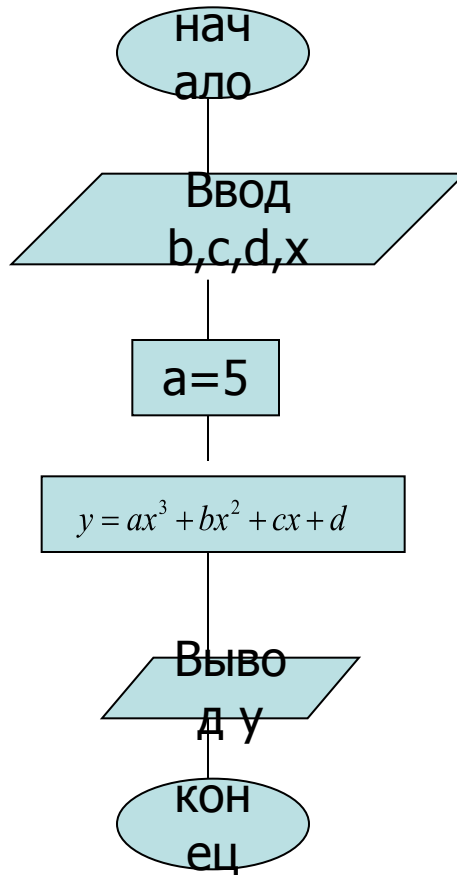
**1. Задача:** Найти значение функции при любых значениях переменных  $b, c, d, x$  и  $a=5$ .

$$y = ax^3 + bx^2 + cx + d$$

# План:

- 1. Описание констант и переменных  $a, b, c, d, x, y$
- 2. Ввод значений  $b, c, d, x$
- 3. Вычисление  $y$
- 4. Вывод значения  $y$ .

# Линейный алгоритм



# Состав программы на Паскале

- Заголовок
- Раздел описания
- Исполняемая часть

# Программа

- [program function.doc](#)

- TP\BIN\TURBO.EXE