

Государственное бюджетное образовательное учреждение  
среднего профессионального образования  
«Новороссийский колледж строительства и экономики»  
Краснодарского края

## Презентация по дисциплине «Основы программирования»

На тему:

«Основы языка программирования С»  
для специальности  
230115 «Программирование в компьютерных  
системах»

Преподаватель: Белова С.В.

## **Задачи:**

- 1. Образовательные:** Обобщение знаний по синтаксису языка программирования С, семантике его языковых конструкций, среде программирования и структуре программы.
- 2. Воспитательные:** Формирование информационной культуры, понятия о языке С, как наиболее распространенном языке программирования.
- 3. Развивающие:** Развитие навыков усвоения теоретического материала.

Язык программирования должен отражать  
фундаментальные и наиболее важные концепции  
алгоритмов в очевидной и легко воспринимаемой форме...  
Н. Вирт

Язык программирования является искусственным языком, в котором строго определены синтаксис (способ соединения слов в словосочетания и предложения) и семантика (значение слов), поэтому, в отличие от естественного языка, здесь недопустимы никакие произвольные интерпретации (истолкования).

Язык C возник как универсальный язык системного программирования. Но он не остался в этих рамках. К концу 80-х годов язык C, оттеснив Fortran с позиции лидера, завоевал массовую популярность среди программистов во всем мире и стал использоваться в самых различных прикладных задачах. Немалую роль здесь сыграло распространение Unix (а значит и C) в университетской среде, где проходило подготовку новое поколение программистов.

Как и все языки, C постепенно совершенствовался, но большинство усовершенствований не носило радикального характера. Наиболее существенным из них, пожалуй, следует считать введение строгой спецификации типов функций, которая значительно повысила надежность межмодульного взаимодействия на C. Все такие усовершенствования были в 1989 году закреплены в стандарте ANSI который и поныне определяет язык C.



Язык программирования включает алфавит, из символов которого формируются слова, выражения и операторы.

Алфавит – это набор допустимых символов языка, из которых формируются все конструкции языка по определенным правилам.

К конструкциям языка относятся:

Слова – идентификаторы (последовательность символов, которая начинается с буквы и служит для именования элементов программы. Идентификаторы, зарезервированные в служебных целях – ключевые слова) и константы,

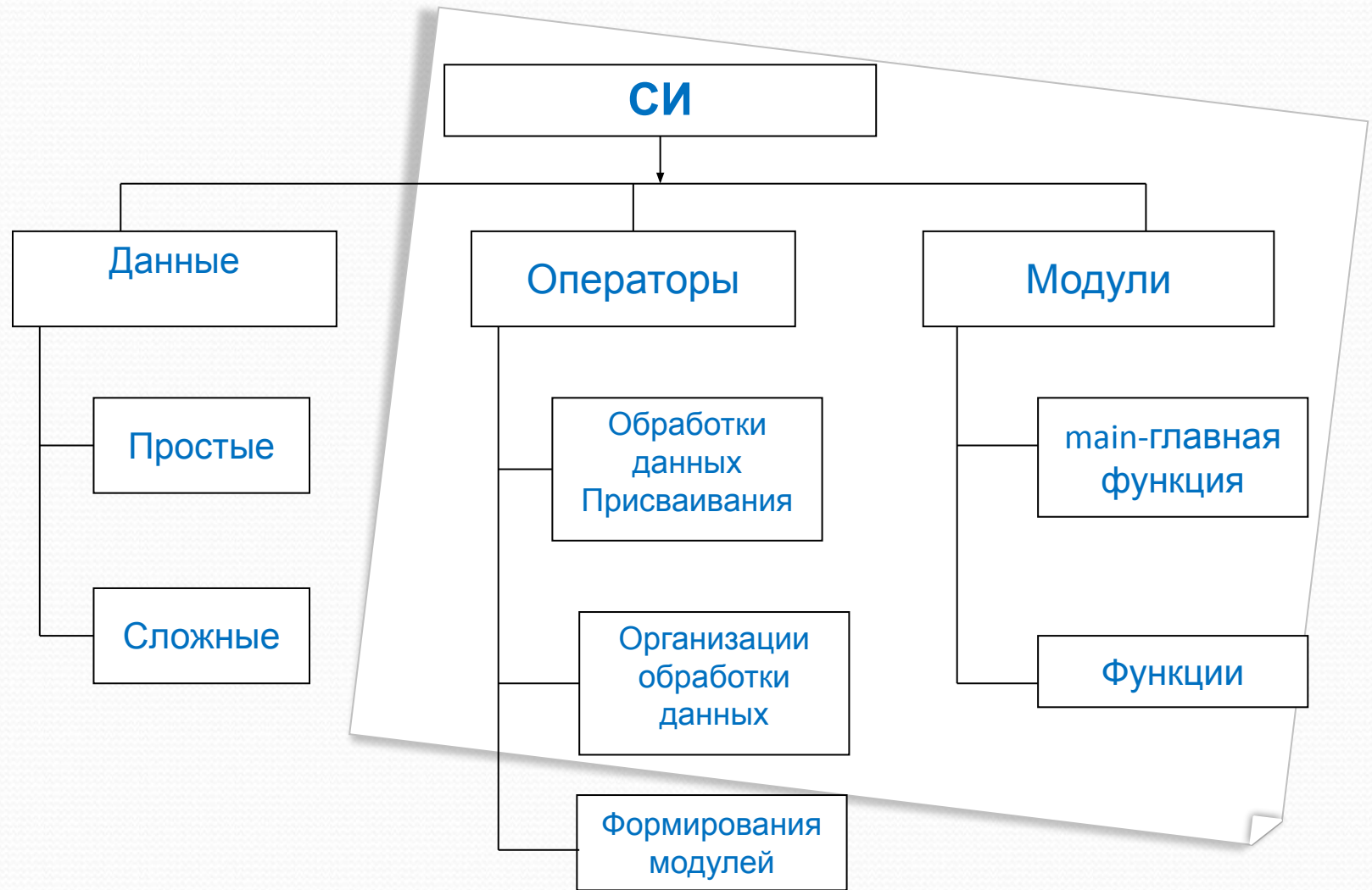
Выражения – группы слов с определенным смыслом, операторы предложения для описания некоторых действий.

Из операторов формируются программы. Программы содержат модули – относительно самостоятельные фрагменты для функционально законченной обработки данных (ввод-вывод). В С – это функции.

Программа состоит из :

1. директив препроцессора;
2. объявления глобальных переменных;
3. одной главной функции;
4. ряда неглавных функций.

# Схема структуры языка:





После того как алгоритм разработан, его необходимо реализовать посредством языка программирования. Это можно сделать с помощью специальных комплексов программ, которые называются системами программирования.

Систем программирования является совокупностью средств, которые обеспечивают автоматизацию разработки и отладки программ. Система включает:

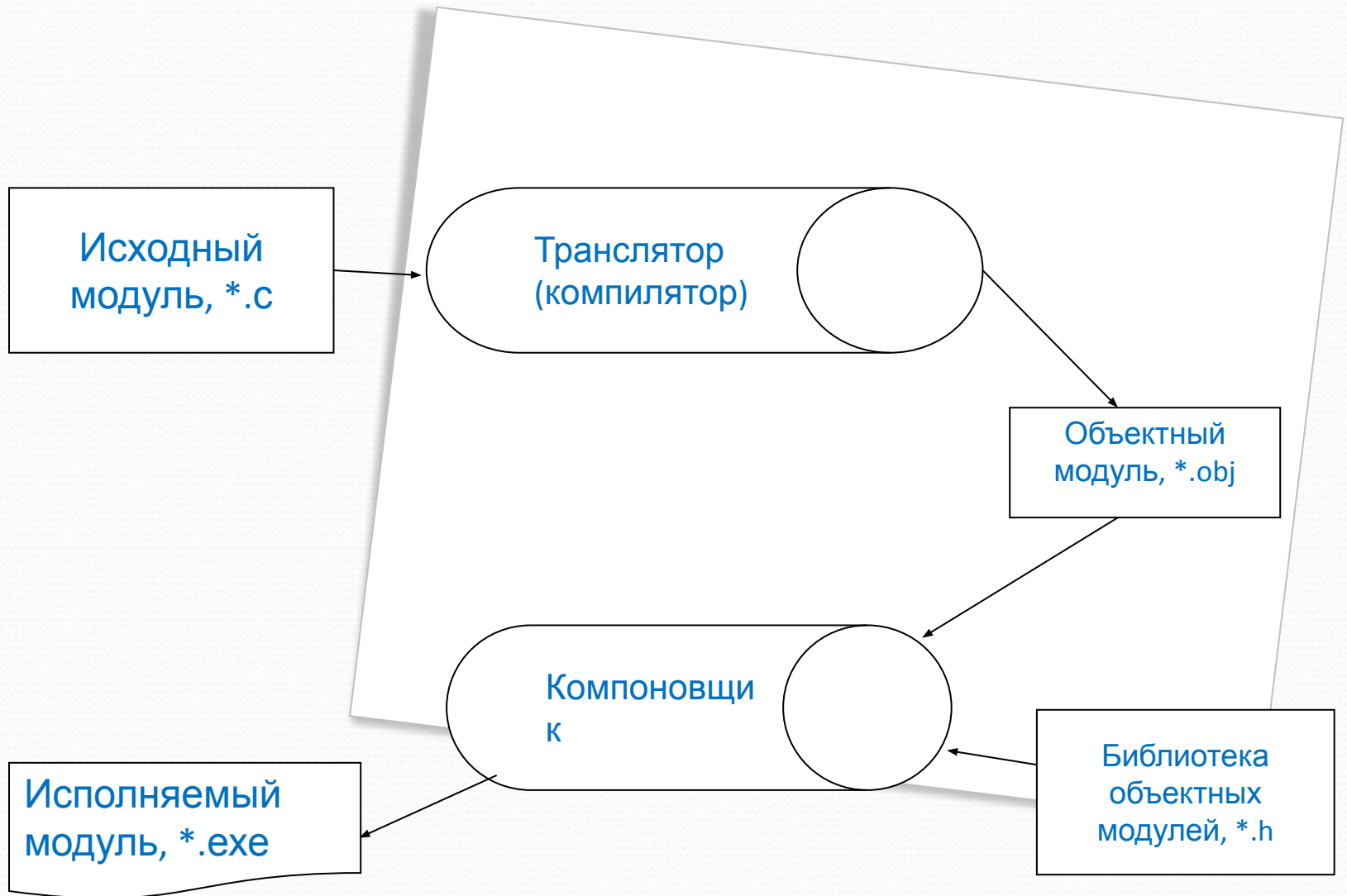
1. язык программирования;
2. транслятор;
3. компоновщик;
4. отладчик;
5. библиотеку программ;
6. справочную систему.

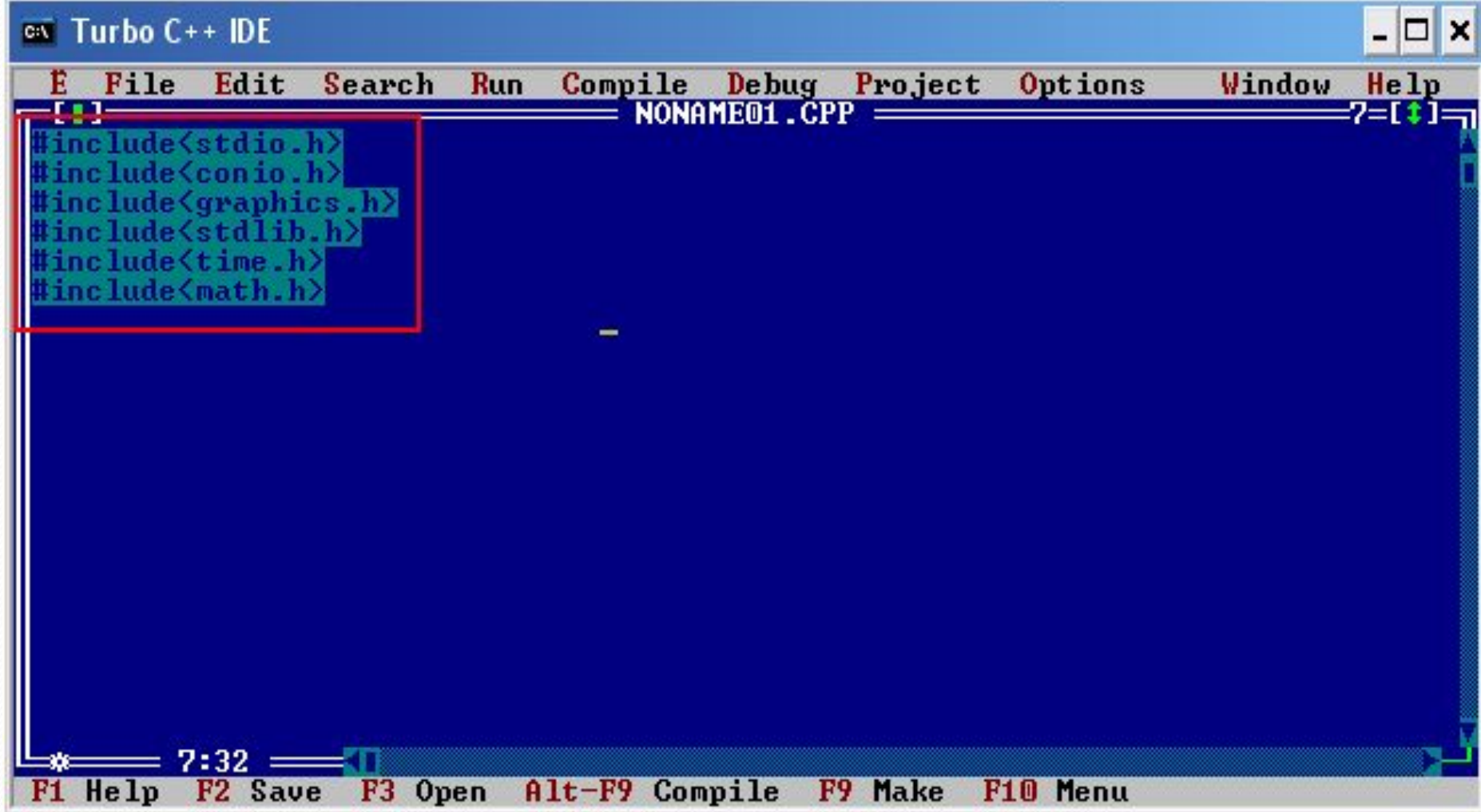


## Процесс создания программы включает следующие этапы:

- 1.написание и редактирование исходного текста программы с сохранением ее в виде исходного файла или модуля ;
- 2.компиляция программы и получение ее объектного файла или модуля. Чтобы программу выполнить, ее необходимо преобразовать в форму, которую может воспринимать компьютер – трансляция. Трансляторы делятся на ассемблеры, интерпретаторы, компиляторы;
- 3.компоновка – построение исполнимого файла или модуля путем объединения полученного объектного модуля программы с объектными модулями стандартных и специальных библиотек. Исполнимый модуль содержит программу на машинном языке с конкретными адресами памяти, готовую к выполнению;
- 4.отладка программы происходит с помощью специального отладчика среды и позволяет найти ошибки, просмотреть отдельные части программы, проследить за изменением переменных.

# Схема выполнения программы:



The image shows a screenshot of the Turbo C++ IDE. The window title is "C:\ Turbo C++ IDE". The menu bar includes "E File", "Edit", "Search", "Run", "Compile", "Debug", "Project", "Options", "Window", and "Help". The current file is "NONAME01.CPP". The code in the editor is:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<stdlib.h>
#include<time.h>
#include<math.h>
```

The code is highlighted in a light blue color. The status bar at the bottom shows the time "7:32" and function key shortcuts: "F1 Help", "F2 Save", "F3 Open", "Alt-F9 Compile", "F9 Make", and "F10 Menu".

Стандартной библиотекой языка Си (также известная как libc) называется нестандартизованная коллекция заголовочных файлов и библиотек, вызываемых как подпрограммы для реализации общих операций, таких как обработка ввода/вывода и строк в языке программирования Си.



```
C:\ Turbo C++ IDE
File Edit Search Run Compile Debug Project Options Window Help
D:ZAPOLNEN.C
void main()
{
int x[5]={1,4,7,8,9};
int i,j,a=0,b=10;
float o=2.12;
double t1,t2;
char cat;
}
* 37:11
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

## переменных

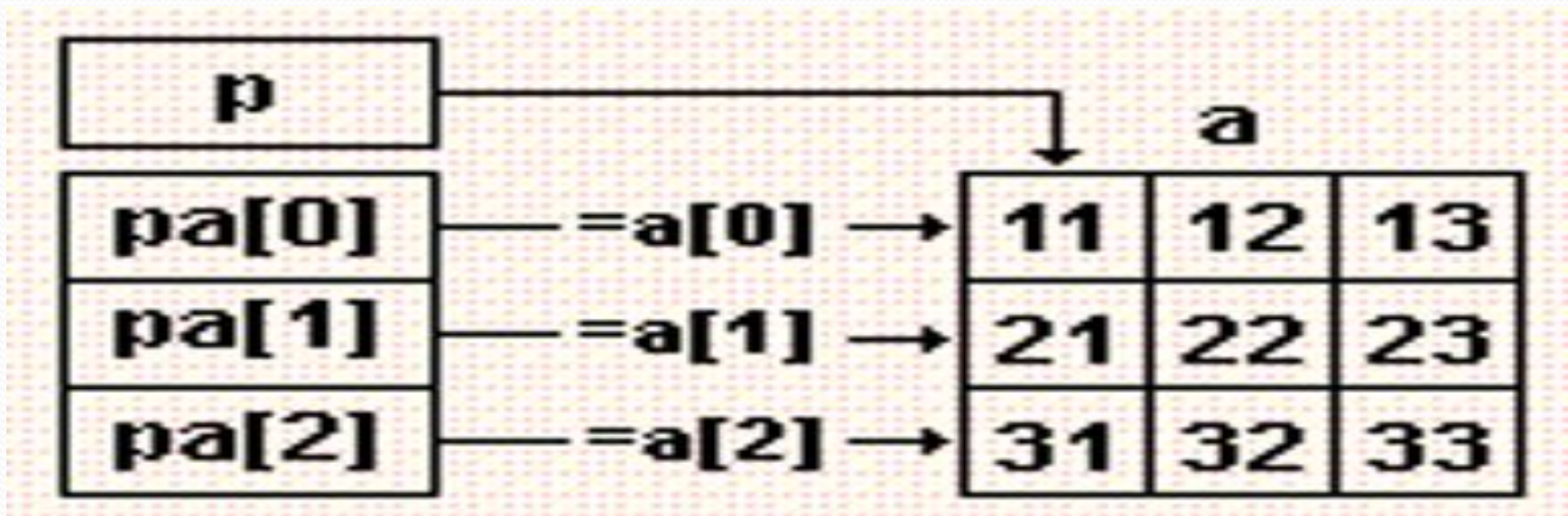
Int-Для целое число.

Float-Вещественный тип.Для чисел с плавающей точкой.

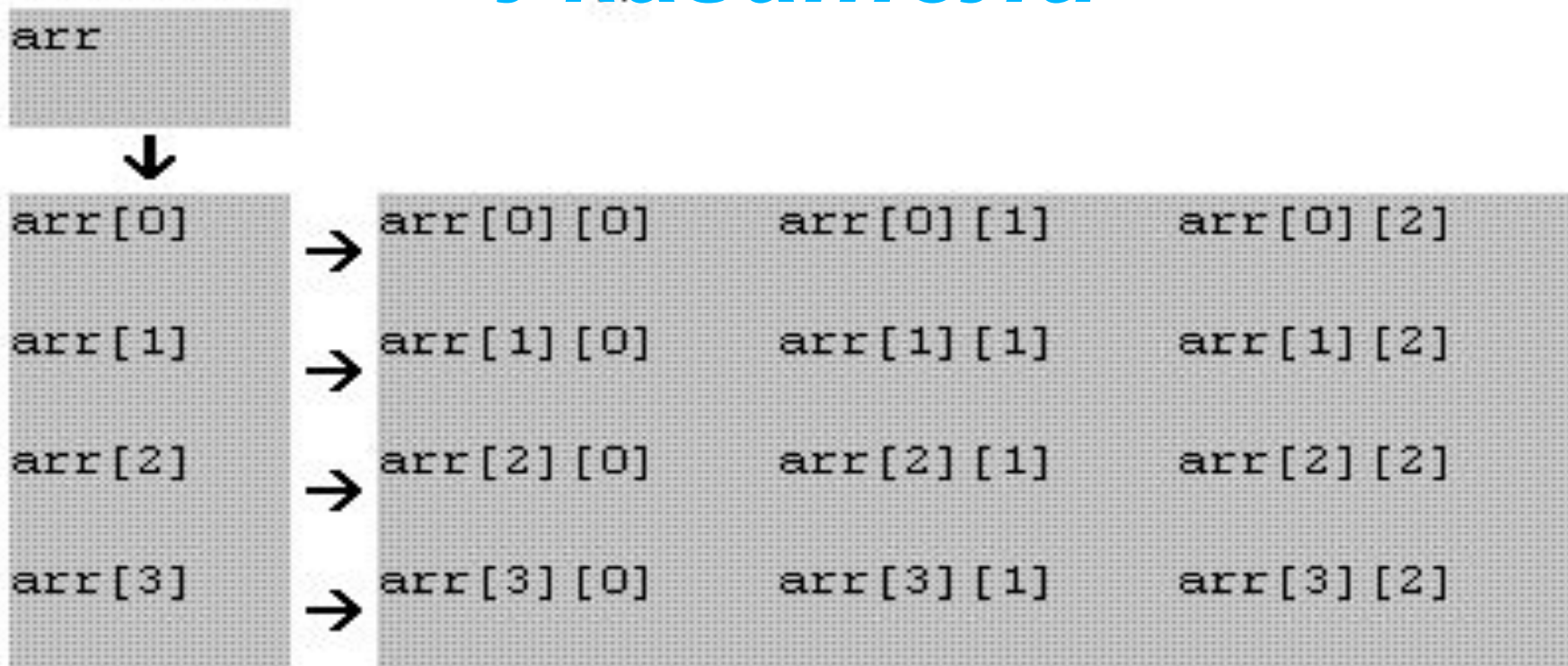
Double-Вещественный тип двойной точности. Больше или равен типу float.

Char-Символьный тип (один символ)

Void-пустой



## указатели





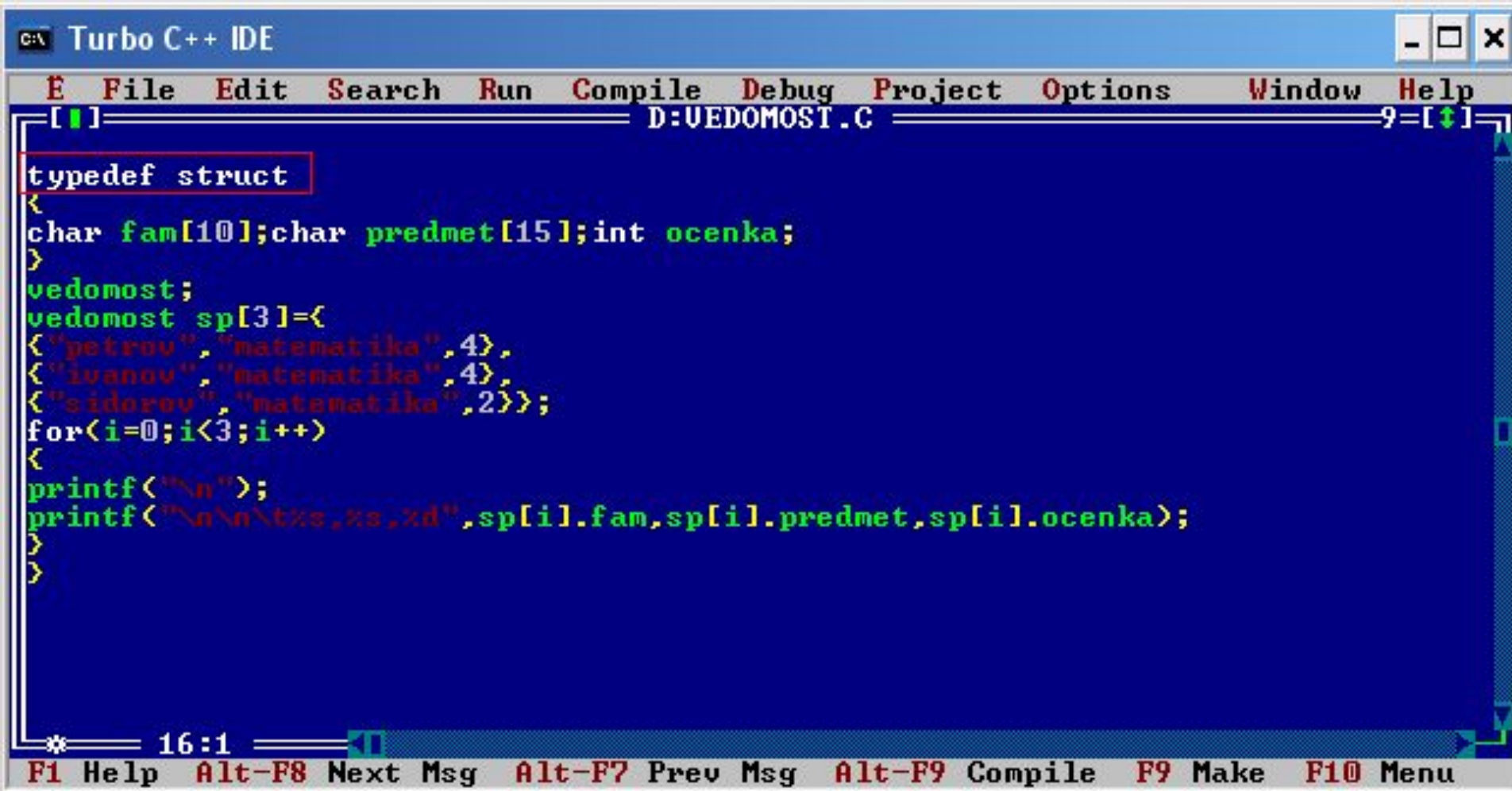
```
C:\ Turbo C++ IDE
E File Edit Search Run Compile Debug Project Options Window Help
D:TABLICA_.C
scanf("%d",&c);
w=c*2;
printf("\t%d",w);
scanf("%d",&c);
x=c*3;
printf("\t%d",x);
y=c*4;
scanf("%d",&c);
printf("\t%d",y);
q=c*5;
scanf("%d",&c);
printf("\t%d",q);
v=c*6;
printf("\t%d",v);
33:1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

## Функции ввода/вывода

Printf-функция форматного вывода — вывода в различные потоки значений разных типов, отформатированных согласно заданному шаблону.

Scanf - функция форматированного ввода. С её помощью можно вводить данные со стандартного устройства ввода (клавиатуры).





```
 Turbo C++ IDE
File Edit Search Run Compile Debug Project Options Window Help
D:\VEDOMOST.C
typedef struct
{
char fam[10];char predmet[15];int ocenka;
}
vedomost;
vedomost sp[3]={
{"petrov", "matematika", 4},
{"ivanov", "matematika", 4},
{"sidorov", "matematika", 2}};
for(i=0; i<3; i++)
{
printf("\n");
printf("\n\n\t%s, %s, %d", sp[i].fam, sp[i].predmet, sp[i].ocenka);
}
}
* 16:1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

Структуры предназначены для хранения в одном месте различных данных, которые относятся к одному понятию.

C:\ Turbo C++ IDE

File Edit Search Run Compile Debug Project Options Window Help

NONAME01.CPP

```
Syntax:
* if ( <expression> ) <statement1>;
* if ( <expression> ) <statement1>;
  else <statement2>;

if (x < y)
  z = x;
else
  z = y;
```

3:1

F1 Help F2 Save F3 Open

# Оператор условного перехода If

Выполнение  
оператора if  
начинается с  
вычисления  
выражения.

(выражение)  
оператор-1;  
[else оператор-2;]

C:\ Turbo C++ IDE

File Edit Search Run Compile Debug Project Options Window Help

D:HHH.C

```
if(c==a-b)
{
printf("%s", " ");
e=e+1;
}
else
{
c=a-b;
printf("%s", "no,verno,pravilnij otvet ");
printf("%d",c);
};
```

51:1

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu



The image shows a screenshot of the Turbo C++ IDE. The window title is "C:\ Turbo C++ IDE". The menu bar includes "E File", "Edit", "Search", "Run", "Compile", "Debug", "Project", "Options", "Window", and "Help". The file name is "NONAME01.CPP". The code in the editor is as follows:

```
Syntax:
* switch ( <expression> ) <statement>
* case <constant expression> :
* default :
switch (operand)
<
case MULTIPLY:    x *= y; break;
case DIVIDE:      x /= y; break;
case ADD:         x += y; break;
case SUBTRACT:   x -= y; break;
case INCREMENT2: x++;
case INCREMENT1: x++;      break;
case EXPONENT:
case ROOT:
case MOD:         printf("Not done\n"); break;
default:          printf("Bug!\n");
-
* 17:40
```

The status bar at the bottom shows "F1 Help", "F2 Save", "F3 Open", "Alt-F9 Compile", "F9 Make", and "F10 Menu".

## Оператор выбора `switch`

Оператор `switch` предназначен для организации выбора из множества различных вариантов.



```
NONAME01.CPP
for (i=0; i<100; i++)
    sum += x[i];

for (i=0, t=string; i < 40 && *t; i++, t++)
    putchar(*t);
    putchar('\n');
```

# Цикл со счётчиком

10:25  
F1 Help F2 Save F3 Open

```
D:DREAD.C
for(i=1;i<101;i++)
{
    a[i]=rand()%83;
    printf("%td",a[i]);
}
```

Цикл, в котором некоторая переменная изменяет своё значение от заданного начального значения до конечного значения с некоторым шагом

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

The screenshot shows the Turbo C++ IDE with a file named NONAME01.CPP. The code defines a function `*do*` with an example of a `do` loop. The loop body contains `i = 1; n = 1;`, `n *= i;`, and `i++;`. The loop terminates when `i > factorial`. The status bar at the bottom shows the cursor is at line 1, column 4.

```
*do*  
  
Example:  
i = 1; n = 1;  
do {  
    n *= i;  
    i++;  
} while (i <= factorial);
```

## Цикл с постусловием M

Цикл с  
постусловием —  
цикл, в котором  
условие  
проверяется  
после  
выполнения  
тела цикла:  
do...while.

The screenshot shows the Turbo C++ IDE with a file named D:SORT.C. The code implements a sorting algorithm using a `do` loop. The loop body contains a `for` loop that compares adjacent elements `x[i]` and `x[i+1]` and swaps them if they are in the wrong order. The loop terminates when `c > 0`. The status bar at the bottom shows the cursor is at line 38, column 1.

```
do  
{  
    for(i=0;i<10;i++)  
    {  
        if(x[i]>x[i+1])  
        {  
            h=x[i];  
            x[i]=x[i+1];  
            x[i+1]=h;  
        }  
        c++;  
    }  
}while(c>0);  
  
printf("\n sortirovka po vozrastaniju ");  
for(i=0;i<10;i++)  
{  
    printf("%t\t\td",x[i]);  
}  
};
```

```
 Turbo C++ IDE
File Edit Search Run Compile Debug Project Options Window Help
D:SORT.C
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
void main()
{
int *x;
clrscr();
for(i=0;i<10;i++)
{
x[10]=calloc(sizeof(int),i);
}
printf("%zd\n",x[i]);
}
* 20:1
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

## Выделение памяти

Оператор sizeof позволяет выяснить, сколько байт занимает тот или иной тип. В качестве параметра он принимает или название типа, или переменную соответствующего типа.



```
C:\DOCUME~1\belova\0016~1\TC\BIN\_TC.EXE
E File Edit Search Run Compile Debug Project Options Window Help
14.CPP 2=I+
#include <stdio.h>
#include <conio.h>
#include <math.h>
main ()
<
int mas[7];
int n,c;
clrscr();
for (n=0; n<7; n++)
<
printf ("\n vvedite choslo = ");
scanf ("%d",&mas[n]);
}

for (n=0;n<7;n++)
<

if (mas[n]>0)
<
c=mas[n]*mas[n];
printf (" %d ",c);
* 8:10
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

Программа на языке С

Результат  
выполнения  
программы

```
C:\DOCUME~1\belova\0016~1\TC\BIN\_TC.EXE

vvedite choslo = 4
vvedite choslo = 6
vvedite choslo = 7
vvedite choslo = 8
vvedite choslo = 9
vvedite choslo = 2
vvedite choslo = 4
16 36 49 64 81 4 16
```