



ИНСТИТУТ СИСТЕМОЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Основы программирования
Семинар №1



МИЭТ

НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

1. Типы данных

2. Форматный ввод/вывод данных

2.1. Управляющие символы

2.2. Форматы

3. Структурные операторы

3.1. Оператор IF

3.2. Оператор WHILE/ DO WHILE

3.3. Оператор FOR

3.4. Оператор SWITCH-CASE



Краткая теоретическая справка

БАЗОВЫЕ

ПОЛЬЗОВАТЕЛЬСКИЕ

ЦЕЛОЧИСЛЕННЫЙ

int
short
long int

ВЕЩЕСТВЕННЫЙ

float
double

ЛОГИЧЕСКИЙ

bool

«ПУСТОЙ»

Void

СИМВОЛЬНЫЙ

char

Массивы
Структуры
Классы и пр.

Краткая теоретическая справка

Ввод и вывод информации осуществляется через функции стандартной библиотеки.

Прототипы рассматриваемых функций находятся в файле `stdio.h`.

Эта библиотека содержит функции

`printf()` — для вывода информации

`scanf()` — для ввода информации.

Пример использования функций:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a, b, c;
6      printf("Введите a=");
7      scanf("%d",&a);
8      printf("Введите b=");
9      scanf("%d",&b);
10     c=a+b;
11     printf("\n Ответ: c=%d", c);
12
13     return 0;
14 }
```

Введите a=2

Введите b=2

Ответ: c=4

Краткая теоретическая справка

Управляющие символы не выводятся на экран, а управляют расположением выводимых символов в функции `printf()`. Отличительной чертой управляющего символа является наличие обратного слэша `\` перед ним.

Основные управляющие символы:

- `\n` — перевод строки;
- `\t` — горизонтальная табуляция;
- `\v` — вертикальная табуляция;
- `\b` — возврат на символ;
- `\r` — возврат на начало строки.

Пример использования управляющих СИМВОЛОВ:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a=2, b=2;
6     printf("| \t a= \t | \t b= \t | \t c= \t | \n");
7     printf("| \t %d \t | \t %d \t | \t %d \t |", a,b,a+b);
8     a++;
9     b++;
10    printf("\n | \t %d \t | \t %d \t | \t %d \t |", a,b,a+b);
11    return 0;
12 }
```

	a=		b=		c=	
	2		2		4	
	3		3		6	

Краткая теоретическая справка

Строка форматов содержит форматы для вывода значений. Каждый формат вывода начинается с символа `%`. После строки форматов через запятую указываются имена переменных, которые необходимо вывести.

Основные форматы:

- `%d` — целочисленный формат;
- `%f` — вещественный формат;
- `%c` — символьный формат.

Пример использования форматов:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a=2;
6     double b=4.5;
7     char c='c';
8     printf("\\t целый \\t\\t вещест. \\t\\t символ. \\t\\n");
9     printf("\\t %d \\t\\t %f \\t\\t %c \\t", a,b,c);
10
11     return 0;
12 }
```

целый	вещест.	СИМВОЛ.
2	4.500000	c

Краткая теоретическая справка

Оператор ветвления, проверяет некоторое условие и, в зависимости от результатов, выполняет определенный блок кода. В качестве условий могут использоваться любые операции возвращающие значение типа `bool`. Если при несоблюдении условия надо выполнить какой-то код, то к оператору `if` можно добавить блок `else`.

```
if(условие) {  
    // do something  
} else {  
    // do something else  
}
```

Пример использования оператора:

```
1  #include <stdio.h>  
2  
3  int main()  
4  {  
5      int a;  
6      int b=1;  
7      double c;  
8      printf("Введите a=");  
9      scanf("%d", &a);  
10     if (a!=0){  
11         c= (double)b/2;  
12         printf("c= %2.1f", c);  
13     }  
14     else {  
15         printf("На 0 делить нельзя!");  
16     }  
17 }
```

```
Введите a=2  
c= 0.5
```

```
Введите a=0  
На 0 делить нельзя!
```

Краткая теоретическая справка

● while

Оператор цикла, выполняет блок кода, пока условие принимает значение true. Перед каждым выполнением цикла, выражение **while** проверяется и может выполняться от нуля (то есть вообще не выполняться) до нескольких раз.

```
while (условие) {  
    // do something  
}
```

● do while

Оператор цикла, в отличии от **while** проверка условия происходит после выполнения. То есть оператор выполнится хотя бы один раз.

```
do {  
    // do something  
} while (условие)
```

Цикл может быть прерван операторами **break**, **return**, **throw** или **goto**.

while

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a=0;
6      while (a<10){
7          a++;
8      }
9      printf("a=%d", a);
10
11     return 0;
12 }
```

a=10

do while

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a=0;
6      do{
7          a++;
8      } while (a<10);
9      printf("a=%d", a);
10
11     return 0;
12 }
```

a=10

Краткая теоретическая справка

Оператор цикла, выполняет блок кода, пока условие не примет значение **false**. В каждом операторе определяются секции инициализации, условия и итератора. Они все определяют, сколько раз цикл будет повторяться.

```
for (инициализатор; условие; итератор)
{
    // do something
}
```

Пример использования оператора:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     for(int i=0; i<10; i++){
6         printf(" %d \n", i);
7     }
8
9     return 0;
10 }
11
```

0

1

2

3

4

5

6

7

8

9

Краткая теоретическая справка

Оператор ветвления, более продвинутый, чем `if`, так как позволяет обрабатывать сразу несколько условий. После каждого блока `case` следует явный выход из `switch` путем использования операторов `break`, `goto` или `return`.

```
int i= (необходимое значение)
switch(i) {
    case 0:    //do something
        break;
    case 1:
    default: //do anything
        break;
}
```

Пример использования оператора:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n;
6      printf("Введите n=");
7      scanf("%d", &n);
8      switch (n){
9          case 1: printf("Цифра один"); break;
10         case 2: printf("Цифра два"); break;
11         case 3: printf("Цифра три"); break;
12         default: printf("Цифра больше трех"); break;
13     }
14
15     return 0;
16 }
```

```
Введите n=2
Цифра два
```

Вариант №1

Вариант №2

Разработать программу с переключающимся меню через оператор `switch-case`. Каждый пункт меню – отдельная задача из списка (далее задачи по вариантам):

1 Вычислить значение $x = a*b/c$, где a , b , c – целые числа, x – вещественное, a , b и c ввести с клавиатуры. Проверить ошибки ввода и вывести сообщение об ошибке (ошибка: $c=0$).

2 Для целого числа k (вводить с клавиатуры) от 1 до 130 вывести фразу «Мне k лет», учитывая при этом, что при некоторых значениях k слово «лет» надо заменить словом «год» или «года».

3 Поле шахматной доски определено парой натуральных чисел (значения от 0 до 8). Ввести координаты 2-х полей k, l и m, n . Определить являются ли они полями одного цвета.

1 Подсчитать площадь круга по введенному с клавиатуры радиусу. Проверить ошибки ввода и вывести сообщение об ошибке (ошибка: нулевое или отрицательное значение радиуса.)

2 Для целого числа d (вводить с клавиатуры) от 1 до 1000, обозначающего денежную единицу, дописать слово «копейка» в правильной форме.

3 Дано трехзначное число a . Определить, составляют ли цифры числа возрастающую последовательность.