

Государственное областное автономное образовательное учреждение  
«Центр поддержки одаренных детей «Стратегия»

Рассмотрена и принята на заседании  
Педагогического совета ГООАУ «Центр  
поддержки одаренных детей «Стратегия»

УТВЕРЖДАЮ:

Директор ГООАУ «Центр поддержки  
одаренных детей «Стратегия»

И.А. Шуйкова



Протокол от  
« 31 » 08 20 18 г. № 1

Приказ от  
08 20 18 г. № 140/1-н

Образовательная программа по информатике 7-8 классов, реализуемая в  
форме электронного обучения, с применением дистанционных  
образовательных технологий

Возраст обучающихся: 13-15 лет  
Срок реализации: 1 год.

Авторы программы:  
Шафоростова Е.П.,  
преподаватель

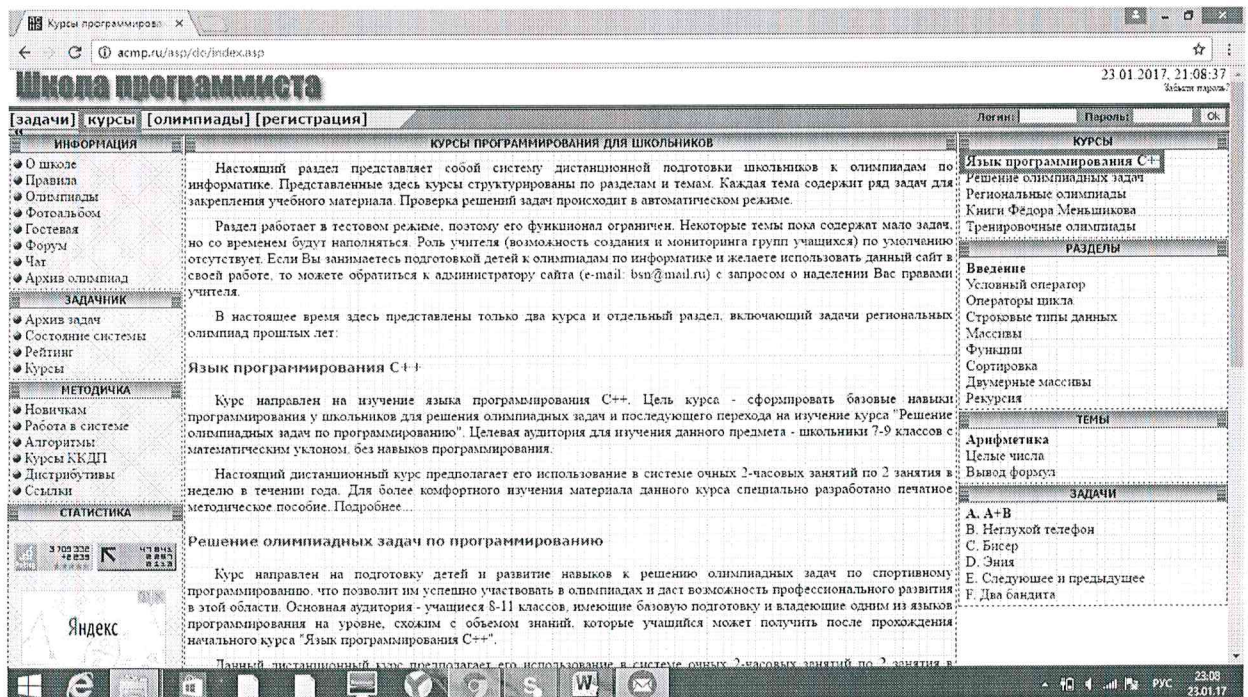
г. Липецк, 2018

## МОДУЛЬ 3. Массивы

Дата	Тема
с 10.12.2018 по 30.12.2018	<p><b>Оглавление</b></p> <p>Ответы на задачи модуля №2..... 3</p> <p>Теоретический материал. Массивы.....8</p>
	<p><b>Практический материал:</b></p> <p>1) Сайт <a href="http://acmp.ru">acmp.ru</a> → Раздел «Массивы» → тема «Линейный поиск» → задачи A-F.</p> <p>2) Сайт <a href="http://acmp.ru">acmp.ru</a> → Раздел «Массивы» → Тема «Преобразования и анализ данных» → задачи A-D.</p>

### АЛГОРИТМ РАБОТЫ

1. Откройте сайт [acmp.ru](http://acmp.ru).
2. В меню, в верхней части окна выберите Курсы, в правой части окна выберите курс под названием «Язык программирования C++». Соответствующие разделы, темы, задачи следует выбирать в меню в правой части окна.



3. При первом посещении на данном сайте нужно зарегистрироваться. При регистрации обязательно используйте реальную

фамилию и имя ученика, написанные русскими буквами. Только в этом случае мы сможем анализировать вашу работу и подводить итоги в виде рейтинга участников ОЗШ по данному курсу. При повторном посещении сайта ученик должен войти в свой личный кабинет под логином и паролем, указанным при регистрации учеником.

4. Приступить к работе по заданию, которое указано в учебном модуле. Проверка задач осуществляется в автоматическом режиме на сайте [asmp.ru](http://asmp.ru).

5. Для того, чтобы преподаватель смог оценить работу школьника, ему необходимо создать файл MS Word, в который нужно добавить скриншоты с экрана с решенными задачами. Например,

Раздел «Введение», Тема «Целые числа», Задача А...

Раздел «Условный оператор», Тема «Алгебра и геометрия», Задача А...  
и так далее.

6. Данный файл MS Word нужно сохранить и загрузить в системе <http://school.strategy48.ru/> в разделе «Домашнее задание».

7. Для того, чтобы посмотреть оценки за модуль ученику необходимо зайти в свой личный кабинет на сайте <http://school.strategy48.ru/> и посмотреть их в разделе «Мои оценки». Перед Вами появится одна или несколько таблиц (в зависимости от того сколько предметов Вы выбрали) с оценками и Вашим средним баллом по выбранному предмету.

8. Размещение нового учебного модуля, сдача домашних заданий и выставление оценок осуществляется согласно расписанию образовательного процесса.

9. По всем возникающим вопросам ученики могут обращаться к преподавателю Шафоростовой Елене Петровне по электронной почте [volkova.lenochka@mail.ru](mailto:volkova.lenochka@mail.ru) с пометкой в теме письма «ОЗШ». Ответ будет направлен ученику в течение 24 часов.

Желаем успехов!

## Ответы на задачи модуля №2

### Раздел «Операторы цикла». Тема «Цикл с параметром for».

#### Задача А. Монетки

```
#include <iostream>
using namespace std;
int main()
{
    int n, x, k1, k2, i;
    cin>>n; //вводим количество монет
    k1=0; //количество монет, которые лежат решкой вверх
    k2=0; // количество монет, которые лежат орлом вверх
    for (i=1;i<=n;i++)
    {
        cin>>x; //вводим монету
        if (x==1)
            k1++;
        else
            k2++;
    }
    if (k1>k2)
        cout<<k2;
    else
        cout<<k1;
    return 0;
}
```

### Раздел «Операторы цикла». Тема «Цикл с параметром for».

#### Задача В. Арбузы

```
#include <iostream>
using namespace std;
int main()
{
    int n, x, max, min, i;
    cin>>n; //вводим количество арбузов
    max=0;
    min=30000;
    for (i=1;i<=n;i++)
    {
        cin>>x; //вводим вес арбуза
        if (x>max)
            max=x;
        if (x<min)
            min=x;
    }
    cout<<min<<' '<<max;
    return 0;
}
```

### Раздел «Операторы цикла». Тема «Цикл с параметром for».

#### Задача С. Конечные автоматы

```
#include <iostream>
using namespace std;
int main()
{
    int k, n, m, i;
    cin>>k;
    for (i=1;i<=k;i++)
```

```

    {
        cin>>n>>m;
        cout<<19*m+(n+239)*(n+366)/2<<endl;
    }

    return 0;
}

```

**Раздел «Операторы цикла». Тема «Цикл с параметром for».**  
**Задача D. Автобусная экскурсия**

```

#include <iostream>
using namespace std;
int main()
{
    int n,x, i;
    cin>>n;
    for (i=1;i<=n;i++)
    {
        cin>>x;
        if (x<=437)
        {
            cout<<"Crash "<<i;
            return 0;
        }
    }
    cout<<"No crash";

    return 0;
}

```

**Раздел «Операторы цикла». Тема «Цикл с предусловием while».**  
**Задача E. Числа Фибоначчи**

```

#include <iostream>
using namespace std;
int main() {
    long long n, a, b, c, i=2;
    cin>>n;
    a=0;
    b=1;
    while (i<=n)
    {
        c=a+b;
        a=b;
        b=c;
        i++;
    }

    if (n<=1)
        cout<<n;
    else
        cout<<c;
    return 0;
}

```

**Раздел «Операторы цикла». Тема «Цикл с предусловием while».**  
**Задача F. Числа Фибоначчи-2**

```

#include <iostream>
using namespace std;

```

```

int main() {
long long A, n, a, b, c, i=2;
cin>>A;
a=0;
b=1;
while(c<A)
{
c=a+b;
a=b;
b=c;
i++;
}

if (A==c)
cout<<1<<endl<<i-1;
else
cout<<0;
return 0;
}

```

## **Раздел «Операторы цикла». Тема «Цикл с предусловием while».**

### **Задача Г. Банковские проценты**

```

#include <iostream>
using namespace std;
int main()
{
double x, p, y, k=0;
cin>>x>>p>>y;
x=x*100;
y=y*100;
while(x<y)
{
x=x*(1+p/100);
x=(int)x;
k++;
}
cout<<k;
return 0;
}

```

## **Раздел «Операторы цикла». Тема «Цикл с предусловием while».**

### **Задача Н. Утренняя пробежка**

```

#include <iostream>
using namespace std;
int main()
{
long double x, y;
cin>>x>>y;
int i=1;
while(x<y)
{
x=x*115/100;
i++;
}
cout<<i;
return 0;
}

```

## **Раздел «Операторы цикла». Тема «Цикл с предусловием while».**

### **Задача I. Сумма максимума и минимума**

```

#include <iostream>
using namespace std;
int main()
{
    int x,y,min,max;
    cin>>min>>max;
    while(1)
    {
        if( cin>>x)
        {
            if (x<min)
                min=x;
        }
        else break;
        if ( cin>>y)
        {
            if(y>max)
                max=y;
        }
        else break;
    }
    cout<<min+max;

    return 0;
}

```

## Раздел «Операторы цикла». Тема «Цикл с постусловием do..while».

### Задача А. Длина последовательности

```

#include <iostream>
using namespace std;
int main()
{
    int x,k=0;
    do
    {
        cin>>x;
        if (x!=0)
            k++;
    }
    while(x!=0);
    cout<<k;
    return 0;
}

```

## Раздел «Операторы цикла». Тема «Цикл с постусловием do..while».

### Задача В. Сумма последовательности

```

#include <iostream>
using namespace std;
int main()
{
    int x,s=0;
    do
    {
        cin>>x;
        if (x!=0)
            s=s+x;
    }
    while(x!=0);
}

```

```
    cout<<s;
    return 0;
}
```

**Раздел «Операторы цикла». Тема «Цикл с постусловием do..while».**

**Задача С. Среднее значение**

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    int x,k=0,s=0;
    do
    {
        cin>>x;
        if (x!=0)
        {
            k++;
            s=s+x;
        }
    }
    while(x!=0);

    cout<<setprecision(3)<<fixed<<(double)s/k;
    return 0;
}
```

**Раздел «Операторы цикла». Тема «Цикл с постусловием do..while».**

**Задача D. Количество четных элементов**

```
#include <iostream>
using namespace std;
int main()
{
    int x,k=0;
    do
    {
        cin>>x;
        if (x!=0 && x%2==0)
            k++;
    }
    while(x!=0);
    cout<<k;
    return 0;
}
```



## Теоретический материал. Массивы

### 1. Понятие массива.

*Массив* – это нестандартный тип данных; структура данных, представляющая собой набор переменных одинакового типа, имеющих общее имя, расположенных в смежных участках памяти, что позволяет получить доступ к каждому элементу по его номеру.

*Пример: одномерный массив*

	0	1	2	3	индексы элементов
T=	{ 7	5	-3	0 }	элементы массива

Количество индексов у одного элемента массива определяет его размерность. Различают массивы:

1) *Одномерный массив* – массив, каждый элемент которого содержит один индекс.

2) *Двумерный массив* – массив, каждый элемент которого содержит два индекса (номер строки и номер столбца).

Индексация элементов в массиве начинается с 0.

*Пример: двумерный массив*

$$M = \begin{pmatrix} 2 & 3 & 8 \\ 5 & -1 & 4 \\ 6 & 7 & 9 \end{pmatrix}$$

M[1][2]=4

### 2. Объявление одномерного массива.

Массив, как и любая переменная программы, перед использованием должен быть объявлен.

Пример объявления одномерного массива, имя массива – а, элементы целого типа, количество элементов равно 100:

```
int a[100];
```

### 3. Способы задания значений элементов одномерного массива.

1) *Инициализация массива в момент его объявления:*

```
int a[16] = { 5, -12, -12, 9, 10, 0, -9, -12, -1, 23, 65, 64, 11, 43, 39, -15 }.
```

2) *Ввод элементов массива с клавиатуры:*

```
int a[100];
int n, i;
cin>>n;
for ( i=0; i<n; i++)
    cin>>a[i];
```

#### 4. Вывод элементов одномерного массива на экран.

1) Вывод элементов в строчку через пробел:

```
for ( i=0; i<n; i++)  
    cout<<a[i]<<' ';
```

2) Вывод элементов в столбик

```
for ( i=0; i<n; i++)  
    cout<<a[i]<<endl;
```

или

```
for ( i=0; i<n; i++)  
    cout<<a[i]<<"\n";
```

#### 5. Типичные алгоритмы обработки массивов.

1) Поиск наибольшего (наименьшего) элемента массива и определение его индекса.

а) поиск наибольшего элемента массива и его индекса:

```
int max=a[0]; //максимальный элемент  
int ind_max=0; //индекс максимального элемента  
for (int i=0; i<n; i++)  
    if (a[i]>max)  
    {  
        max=a[i];  
        ind_max=i;  
    }
```

б) поиск наименьшего элемента массива и его индекса

```
int min=a[1]; //минимальный элемент  
int ind_min=1; //индекс минимального элемента  
for (int i=0; i<n; i++)  
    if (a[i]<min)  
    {  
        min=a[i];  
        ind_min=i;  
    }
```

2) Вычисление суммы (произведения) элементов массива.

а) Сумма элементов массива, например, кратных 5:

```
int s=0;  
for (int i=0; i<n; i++)
```

```

if (a[i]%5==0)
    s=s+a[i];

```

б) Произведение элементов массива, например, кратных 7 и оканчивающихся на 4.

```

int p=1;
for (int i=0; i<n; i++)
    if (a[i]%7==0) && (a[i]%10==4)
        p=p*a[i];

```

3) Подсчет количества элементов массива, удовлетворяющих заданному условию, например, четных и не кратных 3.

```

int k=0;
for (int i=0; i<n; i++)
    if (a[i]%2==0) && (a[i]%3<>0)
        k++;

```

## 6. Пример

Дан массив, состоящий из целых чисел. Напишите программу, которая подсчитывает количество положительных чисел среди элементов массива.

*Входные данные*

Сначала задано число  $N$  — количество элементов в массиве ( $1 \leq N \leq 10000$ ). Далее через пробел записаны  $N$  чисел — элементы массива. Массив состоит из целых чисел.

*Выходные данные*

Необходимо единственное число - количество положительных элементов в массиве.

Примеры

Входные данные	Выходные данные
5 1 2 3 -1 -4	3

```

#include <iostream>
using namespace std;

```

```

int main()
{
    int a[10000];
    int n,i,k=0;
    cin>>n;
    for(int i=0;i<n;i++)

```

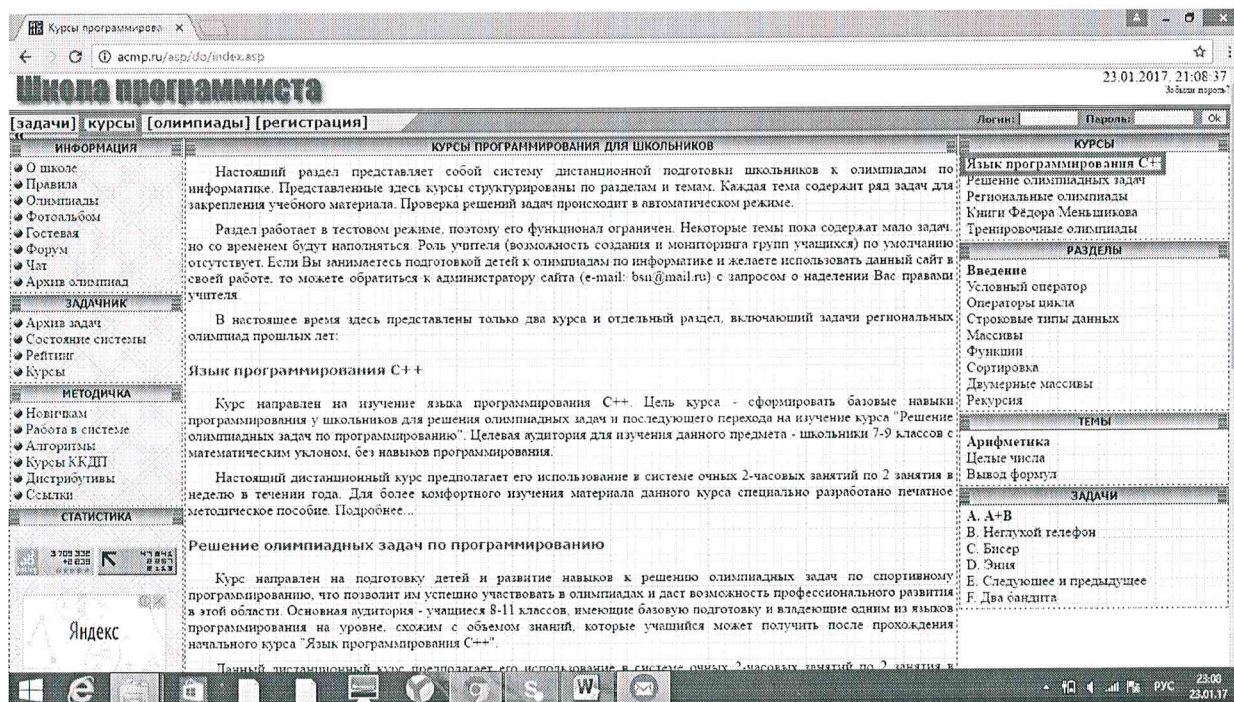
```
        cin>>a[i];
    for(i=1;i<n;i++)
        if(a[i]>0)
            k++;
    cout<<k;
    return 0;
}
```

## МОДУЛЬ 2. Операторы цикла

Дата	Тема
с 12.11.2018 по 02.12.2018	<p>Оглавление</p> <p>Ответы на задачи модуля №1 ..... 3</p> <p>Операторы цикла..... 9</p>
	<p><b>Практический материал:</b></p> <p>1) Сайт <a href="http://asmp.ru">asmp.ru</a> → Раздел «Операторы цикла» → тема «Цикл с параметром for» → задачи A-D.</p> <p>2) Сайт <a href="http://asmp.ru">asmp.ru</a> → Раздел «Операторы цикла» → Тема «Цикл с предусловием while» → задачи E-I.</p> <p>3) Сайт <a href="http://asmp.ru">asmp.ru</a> → Раздел «Операторы цикла» → Тема «Цикл с постусловием do..while» → задачи A-D.</p>

### АЛГОРИТМ РАБОТЫ

1. Откройте сайт [asmp.ru](http://asmp.ru).
2. В меню, в верхней части окна выберите Курсы, в правой части окна выберите курс под названием «Язык программирования C++». Соответствующие разделы, темы, задачи следует выбирать в меню в правой части окна.



3. При первом посещении на данном сайте нужно зарегистрироваться. При регистрации обязательно используйте реальную фамилию и имя ученика, написанные русскими буквами. Только в этом случае мы сможем анализировать вашу работу и подводить итоги в виде рейтинга участников ОЗШ по данному курсу. При повторном посещении сайта ученик должен войти в свой личный кабинет под логином и паролем, указанным при регистрации учеником.

4. Приступить к работе по заданию, которое указано в учебном модуле. Проверка задач осуществляется в автоматическом режиме на сайте [asmp.ru](http://asmp.ru).

5. Для того, чтобы преподаватель смог оценить работу школьника, ему необходимо создать файл MS Word, в который нужно добавить скриншоты с экрана с решенными задачами. Например,

Раздел «Введение», Тема «Целые числа», Задача А...

Раздел «Условный оператор», Тема «Алгебра и геометрия», Задача А...  
и так далее.

6. Данный файл MS Word нужно сохранить и загрузить в системе <http://school.strategy48.ru/> в разделе «Домашнее задание».

7. Для того, чтобы посмотреть оценки за модуль ученику необходимо зайти в свой личный кабинет на сайте <http://school.strategy48.ru/> и посмотреть их в разделе «Мои оценки». Перед Вами появится одна или несколько таблиц (в зависимости от того сколько предметов Вы выбрали) с оценками и Вашим средним баллом по выбранному предмету.

8. Размещение нового учебного модуля, сдача домашних заданий и выставление оценок осуществляется согласно расписанию образовательного процесса.

9. По всем возникающим вопросам ученики могут обращаться к преподавателю Шафоростовой Елене Петровне по электронной почте [volkova.lenochka@mail.ru](mailto:volkova.lenochka@mail.ru) с пометкой в теме письма «ОЗШ». Ответ будет направлен ученику в течение 24 часов.

Желаем успехов!

## Ответы на задачи модуля №1

### Раздел «Введение». Тема «Целые числа».

#### Задача А. Пятью пять – двадцать пять

```
#include <iostream>
using namespace std;

int main()
{
    int n;
    cin>>n;
    if (n<10)
        cout<<n*n;
    else
        cout<<(n/10)*(n/10+1)<<25;
    return 0;
}
```

### Раздел «Введение». Тема «Целые числа».

#### Задача В. Последняя цифра

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    cin >> a;
    cout << a % 10;
    return 0;
}
```

### Раздел «Введение». Тема «Целые числа».

#### Задача С. Число десятков

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    cin >> a;
    cout << (a / 10)%10;
    return 0;
}
```

### Раздел «Введение». Тема «Целые числа».

#### Задача D. Сумма цифр числа

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    cin >> a;
    cout << a/100+(a/10)%10+a%10;
    return 0;
}
```

### Раздел «Введение». Тема «Целые числа».

### **Задача Е. Перевязь**

```
#include <iostream>
using namespace std;
int main()
{
    int a,b;
    cin >> a;
    b=a/10;
    if(a%10>0)
    {
        b=a/10+1;
    }
    else
    {
        b=a/10;
    }
    cout << b;
    return 0;
}
```

### **Раздел «Введение». Тема «Целые числа».**

#### **Задача Ф. Проверьте делимость**

```
#include <iostream>
using namespace std;
int main()
{

    int a,b;
    cin >> a >> b;
    if(a%b==0||b%a==0)
    {
        cout<<"1";
    }
    else
    {
        cout<<"2";
    }
    return 0;
}
```

### **Раздел «Введение». Тема «Целые числа».**

#### **Задача Г. Дележ яблок**

```
#include <iostream>
using namespace std;
int main()
{

    int n,k,a,b,c;
    cin >> n >> k;
    a=k/n;
    b=k%n;
    if (b==0)
    {
        c=0;
    }
    else
    {
        c=n-b;
    }
}
```



```
    cout << a <<" "<< b <<" "<< c;  
    return 0;  
}
```

## **Раздел «Введение». Тема «Целые числа».**

### **Задача Н. МКАД**

```
#include <iostream>  
using namespace std;  
int main()  
{  
  
    int v,t;  
    cin >> v >> t;  
    if(v>0)  
    {  
        cout<<(v*t)%109+1;  
    }  
    else  
    {  
        cout<<109+v*t%109+1;  
    }  
    return 0;  
}
```

## **Раздел «Условный оператор». Тема «Алгебра и геометрия».**

### **Задача А. От перестановки что-то меняется**

```
#include <iostream>  
using namespace std;  
int main()  
{  
  
    int a1,a2,a3;  
    cin >> a1 >> a2 >> a3;  
    if(a1+a2==a3)  
    {  
        cout<< "YES";  
    }  
    else  
        if(a1+a3==a2)  
        {  
            cout<< "YES";  
        }  
        else  
            if(a2+a3==a1)  
            {  
                cout<< "YES";  
            }  
            else  
            {  
                cout<< "NO";  
            }  
    return 0;  
}
```

## **Раздел «Условный оператор». Тема «Алгебра и геометрия».**

### **Задача В. Внеземные гости**

```
#include <iostream>  
using namespace std;
```

```

int main()
{

int r1,r2,r3;
cin >> r1 >> r2 >> r3;
if(r1>=r2+r3)
{
cout<< "YES";
}
else
{
cout<< "NO";
}
return 0;
}

```

## Раздел «Условный оператор». Тема «Алгебра и геометрия».

### Задача С. Длина отрезка

```

#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

int main()
{
int x1,y1,x2,y2;
cin>>x1>>y1>>x2>>y2;
cout<<fixed<<setprecision(5)<<sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
return 0;
}

```

## Раздел «Условный оператор». Тема «Алгебра и геометрия».

### Задача D. Две окружности

```

#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

int main()
{
double x1,y1,r1,x2,y2,r2,r;
cin>>x1>>y1>>r1>>x2>>y2>>r2;
r=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
if(r1+r2>=r && r+r2>=r1 && r+r1>=r2)
{
cout<<"YES";
}
else
{
cout<<"NO";
}
return 0;
}

```

## Раздел «Условный оператор». Тема «Алгебра и геометрия».

### Задача F. Коробки

```
1 using namespace std;
2
3 int main() {
4     int a, b, c, a1, b1, c1;
5     cin >> a >> b >> c >> a1 >> b1 >> c1;
6     if (a > b)
7     {
8         swap(a, b);
9     }
10    if (b > c)
11    {
12        swap(b, c);
13    }
14    if (a > b)
15    {
16        swap(a, b);
17    }
18    if (a1 > b1)
19    {
20        swap(a1, b1);
21    }
22    if (b1 > c1)
23    {
24        swap(b1, c1);
25    }
26    if (a1 > b1)
27    {
28        swap(a1, b1);
29    }
30    if (a == a1 && b == b1 && c == c1)
31    {
32        cout << "Boxes are equal";
33    }
34    else if (a >= a1 && b >= b1 && c >= c1)
35    {
36        cout << "The first box is larger than the second one";
37    }
38    else if (a <= a1 && b <= b1 && c <= c1)
39    {
40        cout << "The first box is smaller than the second one";
41    }
42    else
43    {
44        cout << "Boxes are incomparable";
45    }
46
47    return 0;
48 }
```

**Раздел «Условный оператор». Тема «Алгебра и геометрия».**

**Задача Г. Школьная алгебра**

```
1 using namespace std;
2
3 int main() {
4     int a, b, c;
5     cout << "a = ";
6     int a1;
7     cout << "b = ";
8     int b1;
9     cout << "c = ";
10    int c1;
11    int x1, x2;
12    int D;
13    int k;
14    if ((a == 0) && (b == 0))
15    {
16        cout << "No solution";
17    }
18    else
19    {
20        if ((a == 0) && (b != 0))
21        {
22            cout << "x = " << -c/a;
23            return 0;
24        }
25        else if (a != 0)
26        {
27            D = b*b - 4*a*c;
28            cout << "D = " << D;
29            if (D < 0)
30            {
31                cout << "No solution";
32            }
33            else if (D == 0)
34            {
35                x1 = x2 = -b/(2*a);
36                cout << "x = " << x1;
37            }
38            else if (D > 0)
39            {
40                x1 = (-b + sqrt(D))/(2*a);
41                x2 = (-b - sqrt(D))/(2*a);
42                cout << "x1 = " << x1;
43                cout << "x2 = " << x2;
44            }
45        }
46    }
47
48    return 0;
49 }
```

**Раздел «Условный оператор». Тема «Алгебра и геометрия».**

**Задача Н. Квадратное уравнение**

[Безымянный листок]

```
1 #include <iostream>
2 #include <cmath>
3 #include <iomanip>
4 using namespace std;
5
6 int main() {
7     int a, b, c, d = 0;
8     double x1, x2;
9     cin >> a >> b >> c;
10    b = b * (-1);
11    if(a == 0 && b == 0){
12        cout << "0";
13        return 0;
14    }
15    if(a == 0 && b == 0 && c == 0){
16        cout << "0";
17        return 0;
18    }
19    d = (b*b) - 4*(a * c);
20    if (a == 0)
21    {
22        cout << "1";
23        x1 = (c) / b;
24        cout << endl;
25        cout << fixed << setprecision(6) << x1;
26        return 0;
27    }
28    else
29    if (b == 0 && c == 0){
30        cout << "0";
31        return 0;
32    }
33    }
34    else
35    if(d > 0)
36    {
37
38        cout << "2";
39        cout << endl;
40        x1 = (b + (sqrt(d))) / 2*a;
41        x2 = (b - (sqrt(d))) / 2*a;
42        cout << fixed << setprecision(6) << x2;
43        cout << endl;
44        cout << fixed << setprecision(6) << x1;
45        return 0;
46    }
47    else
48    if(d == 0)
49    {
50
51        cout << "1";
52        cout << endl;
53        x1 = (b + (sqrt(d))) / 2*a;
54        cout << fixed << setprecision(6) << x1;
55        return 0;
56    }
57    }
58    else
59    if(d < 0)
60    {
61
62        cout << "0";
63        return 0;
64    }
65 }
```

## Операторы цикла

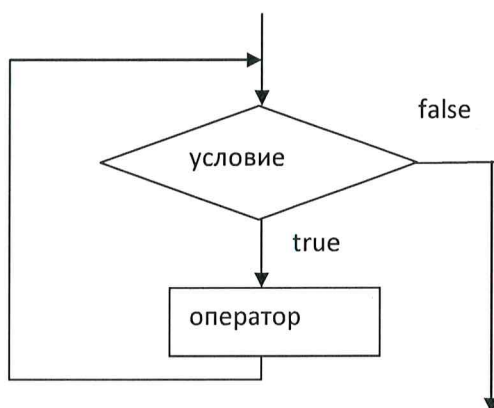
**Цикл** – это базовая алгоритмическая конструкция, в которой некоторая последовательность операторов может повторяться многократно (возможно ни разу) в зависимости от условия.

### Виды циклов:

**а) Цикл с предусловием** – сначала в цикле проверяется условие, если оно принимает значение true (истина), то выполняются операторы тела цикла, затем снова проверяется условие. Выход из цикла происходит, когда условие принимает значение false (ложь).

```
while (условие)
{
    оператор1;
    оператор2;
    .....
    операторN;
}
```

### Блок-схема:



**Пример:** Для заданного  $n$  вычислить сумму ряда  $s=1+2+\dots+n$ .

Решение: Для слагаемого заведем переменную  $a$  с начальным значением 1. Сумму будем накапливать в переменной  $s$ , начальное значение которой равно 0. Цикл будем выполнять до тех пор, пока значение очередного слагаемого не превысит введенного  $n$ . На очередном шаге цикла происходит увеличение значения переменной  $s$  на значение слагаемого, а также увеличиваем значение самого слагаемого на 1.

```
int main() {
    int n, a=1, s=0;
    cin>>n;
```

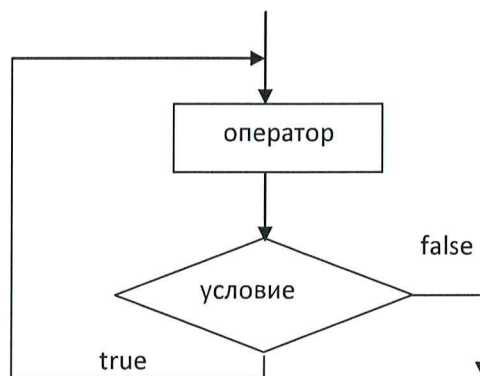
```
while (a<=n)
{
    s=s+a;
    a++;
}
cout<<s;
return 0;
}
```

**б) Цикл с постусловием** – на первом шаге цикла выполняются операторы тела цикла. Затем проверяется условие, если оно принимает значение true (истина), то снова выполняются операторы тела цикла. Выход из цикла происходит, когда условие принимает значение false (ложь).

Отличие цикла с постусловием от цикла с предусловием заключается в том, что тело цикла с постусловием выполнится хотя бы один раз.

```
do
{
    оператор1;
    оператор2;
    .....
    операторN;
}
while (условие);
```

**Блок-схема:**



**Пример:** Для заданного n вычислить сумму ряда  $s=1+2+\dots+n$ .

Продемонстрируем решение предыдущего примера с помощью цикла с постусловием.

```
int main() {
    int n, a=1, s=0;
    cin>>n;
    do
    {
        s=s+a;
        a++;
    }
    while(a<=n);
    cout<<s;
    return 0;
}
```

**в) Цикл со счетчиком (с параметром)**

```
for (выражение_1; выражение_условие; выражение_3)
{
    оператор1;
    оператор2;
    .....
    операторN;
}
```

выражение\_1 вычисляется 1 раз (блок начальных значений),

выражение\_2 играет роль условия, если оно истинно, то цикл выполняется,

выражение\_3 выполняется каждый раз.

*Пример:*

```
a) int s=0;
for (int i=1; i<=n; i++)
{
    s=s+1/i; //или s+=1/i;
}
```

```
б) for (int s=0, i=1; i<=n; i++)
{
```

```

        s=s+1/i; //или s+=1/i;
    }
в) int p=1;
for(int i=10; i>0; i--)
{
    p=p*i;
}
г) int p=1;
for(int i=1; i<=10; i=i+2)
{
    p=p*i;
}

```

### Операторы, используемые в теле цикла

1) **Оператор break** – прерывает исполнение цикла while, do, for.

*Пример:*

```

for (i=1; i<=n; i++)
{
    .....
    while (k!=m)
    {
        .....
        if (k>0)
            break;
    }
}

```

В рассмотренном примере break прерывает вложенный цикл.

2) **Оператор continue** – выполняет переход на следующую итерацию (один виток цикла)

```

while (k!=m)
{
    .....
    if (k>0)
        continue;
}

```

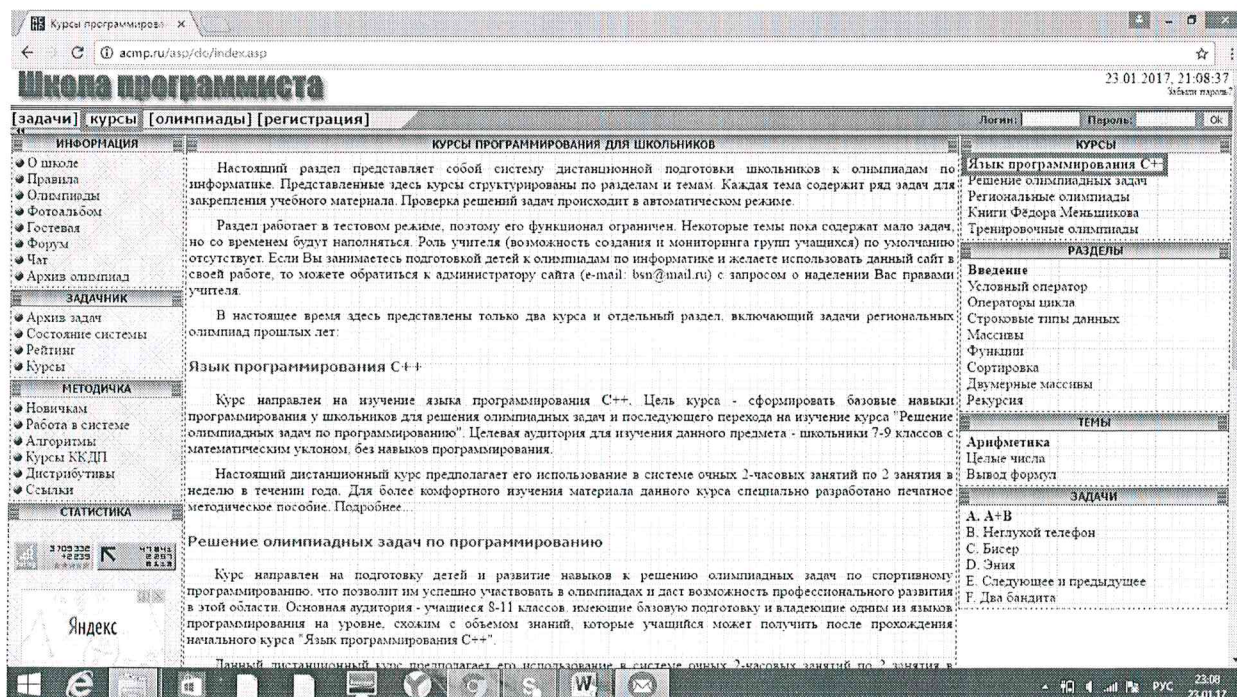


## МОДУЛЬ 1. Введение в язык программирования C++. Условный оператор

Дата	Тема
с 15.10.2018 по 11.11.2018	<b>Теоретический материал:</b> 1) Введение в язык программирования C++: <i>сайт aсmp.ru</i> → Раздел «Введение» 2) Условный оператор: <i>сайт aсmp.ru</i> → Раздел «Условный оператор»
	<b>Практический материал:</b> 1) Сайт <i>aсmp.ru</i> → Раздел «Введение» → тема «Целые числа» → задачи А-Н. 2) Сайт <i>aсmp.ru</i> → Раздел «Условный оператор» → Тема «Алгебра и геометрия» → задачи А-Н.

### АЛГОРИТМ РАБОТЫ

1. Откройте сайт **aсmp.ru**.
2. В меню, в верхней части окна выберите Курсы, в правой части окна выберите курс под названием «Язык программирования C++». Соответствующие разделы, темы, задачи следует выбирать в меню в правой части окна.



3. При первом посещении на данном сайте нужно зарегистрироваться. При регистрации обязательно используйте реальную фамилию и имя ученика, написанные русскими буквами. Только в этом случае мы сможем анализировать вашу работу и подводить итоги в виде рейтинга участников ЗШ по данному курсу. При повторном посещении сайта ученик должен войти в свой личный кабинет под логином и паролем, указанным при регистрации учеником.

4. Приступить к работе по заданию, которое указано в учебном модуле. Проверка задач осуществляется в автоматическом режиме на сайте [aspr.ru](http://aspr.ru).

5. Для того, чтобы преподаватель смог оценить работу школьника, ему необходимо создать файл MS Word, в который нужно добавить скриншоты с экрана с решенными задачами. Например,

Раздел «Введение», Тема «Целые числа», Задача А...

Раздел «Условный оператор», Тема «Алгебра и геометрия», Задача А...  
и так далее.

6. Данный файл MS Word нужно сохранить и загрузить в системе <http://school.strategy48.ru/> в разделе «Домашнее задание».

7. Для того, чтобы посмотреть оценки за модуль ученику необходимо зайти в свой личный кабинет на сайте <http://school.strategy48.ru/> и посмотреть их в разделе «Мои оценки». Перед Вами появится одна или несколько таблиц (в зависимости от того сколько предметов Вы выбрали) с оценками и Вашим средним баллом по выбранному предмету.

8. Размещение нового учебного модуля, сдача домашних заданий и выставление оценок осуществляется согласно расписанию образовательного процесса.

9. По всем возникающим вопросам ученики могут обращаться к преподавателю Шафоростовой Елене Петровне по электронной почте [volkova.lenochka@mail.ru](mailto:volkova.lenochka@mail.ru) с пометкой в теме письма «ЗШ». Ответ будет направлен ученику в течение 24 часов.

Желаем успехов!

## Структура программы на языке C++

Программа на языке C++ – это последовательность четырех частей:

- 1) директивы препроцессору,
- 2) указание компилятору,
- 3) объявления,
- 4) определения.

**Препроцессор** – это специальная программа, которая обрабатывает текст программы раньше компилятора. Все команды препроцессора начинаются знаком #.

**Директива** – это команда, предписывающая препроцессору заменить в тексте программы каждый встретившийся идентификатор на строку замещения. Каждая директива располагается на новой строке и начинается с символа #.

*Виды директив:*

1) **#define идентификатор строка\_замещения** - позволяет объявить именованную константу.

*Пример:*

- а) #define E 2.718282
- б) #define MAXLINE 80
- в) #define SIZE 16

2) **#include <имя файла>** - директива предписывает препроцессору заменить всю строку на содержимое файла, имя которого записано в угловых скобках. Чаще всего подключаются заголовочные файлы с расширением \*.h. Заголовочные файлы содержат объявления макроконстант, типов данных, а также функций (их прототипы - объявления).

*Пример:*

- а) #include <iostream> // библиотека для работы с потоками ввода-вывода
- б) #include <cmath> /\* заголовочный файл содержит прототипы математических функций \*/

**Замечание:** В каждой программе необходимо использовать стандартное пространство имен:

```
using namespace std;
```

**Указания компилятору** – специальные программы, управляющие ходом компиляции.

**Объявления** – объявляться могут переменные, типы данных, функции.

**Определения** – определяться могут переменные, типы данных, функции. В определении переменной может задаваться начальное значение, что приводит к выделению памяти для этой переменной; определение функции содержит заголовок и тело функции.

### Простейшая программа

```
#include <iostream> //подключение функций стандартного потока ввода-вывода
using namespace std;

int main() // начало главной программы
{
    int a,b,c; //объявление целочисленных переменных a,b,c
    cin>>a>>b; //ввод с клавиатуры переменных a и b
    c=a+b; //вычисление суммы двух чисел
    cout<<c; //вывод переменной c на экран
    return 0; /*главная функция main возвращает 0 в случае
успешности своей работы, в противном случае – код ошибки */
```

}

Основная часть программы всегда называется именем **main**. Фигурные скобки обозначают начало и конец основной программы.

**Замечания:**

- язык программирования C++ различает строчные и заглавные буквы, все стандартные операторы C++ записываются строчными буквами;
- однострочный комментарий в программе обозначается //
- многострочный комментарий /\* Комментарий!!! \*/.

## Алфавит, идентификаторы, служебные слова, константы

Алфавит языка C++ состоит из:

- прописные и строчные буквы латинского алфавита;
- цифры;
- специальные знаки: “ , { } | [ ] ( ) + - / % и т.д.

**Идентификатор** – последовательность латинских букв, цифр и символа подчеркивания “\_”, начинающаяся с буквы или символа подчеркивания.

**Служебные (ключевые) слова** – идентификаторы, зарезервированные в языке, т.е. такие, которые нельзя использовать в качестве свободно выбираемых программистом имен.

**Символьная константа** – это лексема, которая состоит из изображения символа и ограничивающих апострофов. Внутри апострофов можно записать любой символ, изображаемый на дисплее или принтере в текстовом режиме. Такие символы в языке Си называют управляющими последовательностями.

*Список управляющих символов:*

- ‘\n’ – перевод строки;
- ‘\t’ – горизонтальная табуляция;
- ‘\r’ – возврат курсора к началу строки;
- ‘\b’ – возврат на одну позицию (на один символ).

## Классификация типов данных

Типы данных определяются:

- 1) множеством значений, которые могут принимать объекты заданного типа;
- 2) множеством операций.

Типы данных бывают: стандартные и нестандартные.

**Типы данных:**

1) **Пустой тип void** – имеет специальное назначение, указывает, что функция не имеет возвращаемого значения.

2) **Простые (скалярные) типы:**

- целые,
- вещественные,
- символьные,
- указатели,
- перечислимый тип.

3) **Составные (структурированные) типы:**

- массив,
- структура,
- объединение.

## Группа целых типов

Тип данных	Размер памяти, бит	Диапазон значений
short int	16	от -32768 до 32767
unsigned int	16	от 0 до 65535 (16-битная платформа)
	32	от 0 до 4294967295 (32-битная платформа)
int	16	От -32768 до 32767 (16-битная платформа)
	32	от -2147483648 до 2147483647 (32-битная платформа)
long	32	от -2147483648 до 2147483647
unsigned long	32	от 0 до 4294967295
long long int	64	от $-(2^{63}-1)$ до $2^{63}-1$
unsigned long long int	64	от 0 до $2^{64}-1$

### Группа вещественных типов

Тип данных	Размер памяти, бит	Диапазон значений
float	32	от $3.4 \cdot E^{-38}$ до $3.4 \cdot E^{38}$
double	64	от $1.7 \cdot E^{-308}$ до $1.7 \cdot E^{308}$
long double	80	от $3.4 \cdot E^{-4932}$ до $3.4 \cdot E^{4932}$

### Логический тип данных

Логический тип данных **bool**. Переменная логического типа может принимать одно из двух значений: true (истина), false (ложь).

*Пример:* объявление переменной:

```
int x; // объявлена переменная целого типа
```

```
double a, b; // объявлены переменные вещественного типа
```

### Средства консольного ввода-вывода

#### 1) Оператор ввода:

```
cin >> переменная;
```

*Пример:*

```
a) cin >> a;
```

```
б) cin >> b >> c;
```

#### 2) Оператор вывода:

```
cout << переменная;
```

*Пример:*

```
a) cout << "v=" << v;
```

```
б) cout << endl; // перенос курсора на следующую строку
```

```
в) вывод вещественного числа с тремя знаками после запятой
```

```
#include <iomanip>
```

```
.....
```

```
int main()
```

```
{
```

```
    float d=2.51234878;
```

```
    cout << fixed; // фиксируем дробную часть числа
```

```
    cout << setprecision(3) << d;
```

```
    return 0;
```

```
}  
В окне вывода будет число 2.512
```

## Русификация программы

```
void main()  
{  
    setlocale(LC_ALL, "Russian");  
}
```

## Выражения. Операции

**Выражение** – грамматическая конструкция языка, состоящая из знаков операций и операндов, задающая порядок вычисления по некоторому правилу, в результате которого получается единственное значение некоторого типа.

Выражение отличается от оператора тем, что в результате выражения получается единственное значение некоторого типа.

По количеству операндов операции бывают:

- 1) унарные (! – отрицание, + унарный плюс);
- 2) бинарные (|| - or, && - and, ++ сложение);

По назначению операции бывают:

1) *Арифметические операции:*

+, -, /, \*, % - остаток от деления, ++(инкремент) – увеличивает значение переменной на 1, --(декремент) – уменьшает значение переменной на 1.

2) *Логические операции:*

&& - логическое «и»;

|| - логические «или»;

! – логическое отрицание;

3) *Операции отношения:*

>, <, >=, <=, ==, !=;

4) *Операции присваивания:*

= простое присваивание,

+= сложение с присваиванием (x+=a или x=x+a),

-= вычитание с присваиванием (x-=a или x=x-a).

5) *Прочие операции*

а) , - бинарная операция (выражение1, выражение 2)

б) операция приведения типа: (тип данных) выражение

*Пример:* (int)k/d;

## Операторы

1) **Пустой оператор ;** - оператор ничего не выполняет, требуется там, где действий выполнять не нужно, но требуется оператор.

*Пример:*

```
for (i=0; i<n; i++)  
    a[i]=0;
```

2) **Составной оператор**

```
{  
    оператор1;  
    оператор2;  
    .....  
    оператор N;  
}
```

```
}
```

### 3) Оператор-выражение

*Пример:*

a) a=1;

б) i++;

### 4) Условный оператор

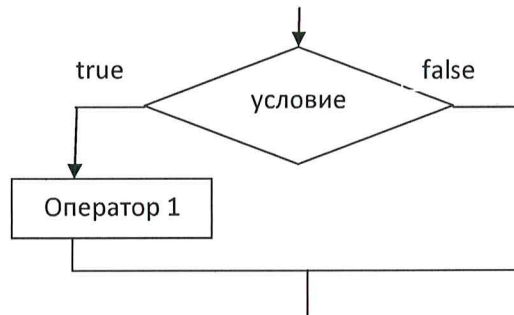
```
if (условие)
```

```
    оператор1;
```

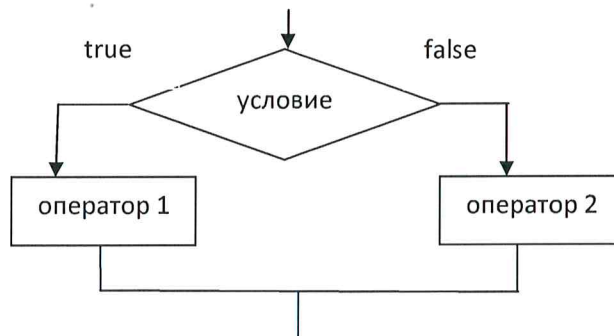
```
[else
```

```
    оператор2;]
```

- Неполная развилка



- Полная развилка



*Пример:*

```
a) if (a>b)
    cout<<a;
else
    cout<<b;
```

### 5) Оператор-переключатель switch

```
switch (выражение)
```

```
{
```

```
    case константа1: {оператор1;break;}
```

```
    case константа2: {оператор2;break;}
```

```
    .....
```

```
    case константаN: {операторN;break;}
```

```
    [ default: оператор по умолчанию;]
```

```
}
```

*Пример:*

```
a) switch(i)
{
    case 0:c0++;
    case 1:c1++;
}
```