



9-11 классы

Программирование на C++

Презентация занятия

Типы данных, строки.
Оператор switch, цикл for.

3 занятие



Минцифры
России



20.35
УНИВЕРСИТЕТ

Программирование
на C++

Теоретическая часть

Типы данных, строки.
Оператор switch, цикл for.

3 занятие



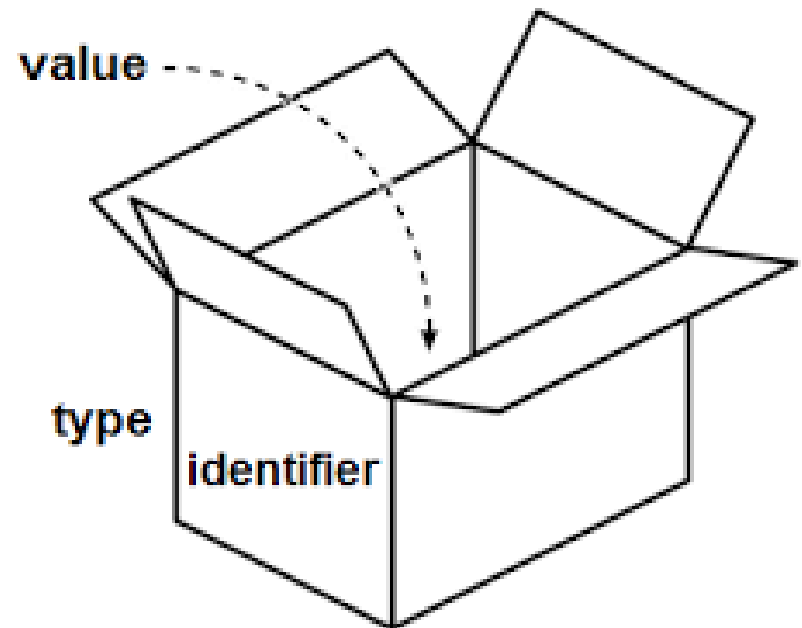
2020

Тема: Типы данных, строки. Оператор switch, цикл for.

Типы данных

Операционная система распределяет память и выбирает, что будет сохранено в выделенную память, основываясь на типе данных переменной.

Тип данных определяет правильное использование идентификатора (имени), какой тип информации может быть сохранен, и какие типы операций могут быть выполнены.



Тема: Типы данных, строки. Оператор switch, цикл for.

Выражения

В приведенных ниже примерах показано правильное (legal) и неправильное (illegal) использование выражений языка C++.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      55 + 15;    // legal C++ expression
8                  //Both operands of the + operator are integers
9
10     55 + "John";// illegal
11                  // The + operator is not defined for integer and string
12
13     return (0);
14 }
```

Тема: Типы данных, строки. Оператор switch, цикл for.

Числовые типы данных

Числовые типы данных включают:

Целые числа, такие как -7, 42.

Числа с плавающей точкой (десятичные дроби), такие как 3.14, -42.67.

Строки и Символы

Строки состоят из чисел, букв, или символов. Строковые литералы размещаются в двойных кавычках; примеры "Hello", "My name is David".

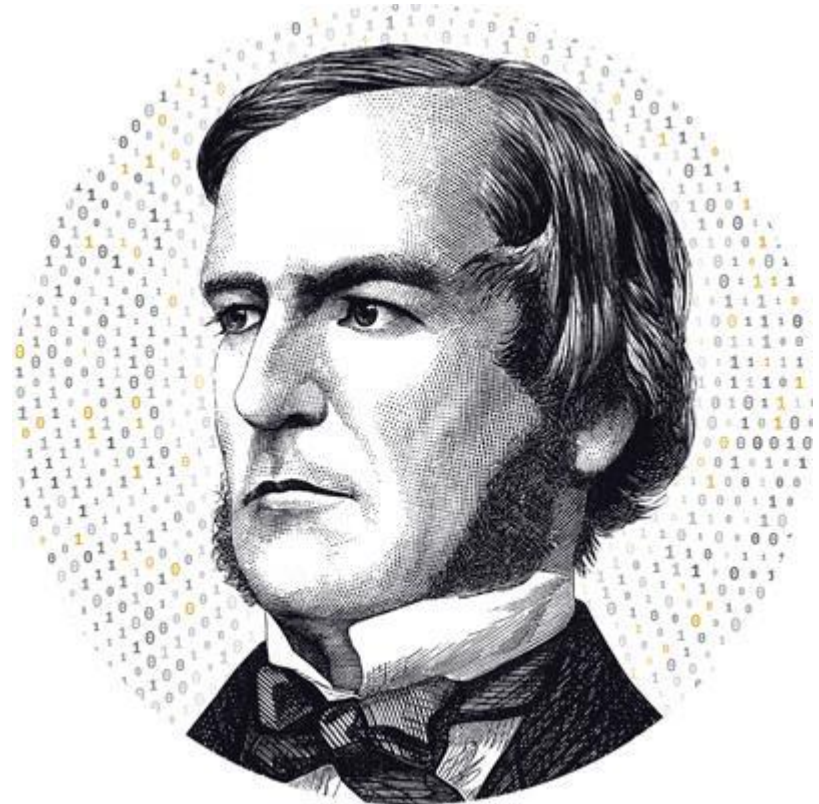
Символы это одиночные буквы или символы, они должны быть закрыты между одинарными кавычками, например 'a', 'b', и т.д.

Тема: Типы данных, строки. Оператор switch, цикл for.

Булевый тип

Булевый тип возвращает только два возможных значения: true(истина) -> (1) и false(ложь) -> (0).

Условные выражения являются примером Булевого типа данных.



George Boole

Тема: Типы данных, строки. Оператор switch, цикл for.

Целочисленный тип

Целочисленный тип хранит не дробные числа, которые могут быть положительными или отрицательными. Например, к целочисленному типу может относиться 42, -42, и подобные числа.

Размер целочисленного типа варьируется согласно с архитектурой системы, на которой запускается программа, однако 4 байта это минимальный размер в большинстве современных систем.

Используйте ключевое слово `int` для определения целочисленного типа.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      int a = 42;
9      int b;
10
11     b = a - 42; // b == 0;
12     cout << a << endl;
13     cout << b << endl;
14
15     return (0);
16 }
```

Тема: Типы данных, строки. Оператор switch, цикл for.

Целочисленный тип

Некоторые из базовых типов, включая целочисленный, могут быть модифицированы используя один или несколько следующих спецификаторов:
signed: Знаковый целый тип может хранить и отрицательные и положительные значения.

unsigned: Беззнаковые целые могут хранить только положительные значения.

short: Половина стандартного размера.

long: Удвоенный стандартный размер.

Целочисленный тип данных резервирует 4-8 байт в зависимости от операционной системы.

```
int      a = 42;      // int == signed int
short    b = 123;
long     c = 12345;
unsigned int d = 100000;
```


Тема: Типы данных, строки. Оператор switch, цикл for.

Числа с плавающей точкой

Переменные типа с плавающей точкой могут хранить числа, такие как 420.0, - 3.33, или 0.03325. Слова «плавающая точка» ссылаются на тот факт, что некоторое количество цифр может быть до и после десятичной точки. Вы можете сказать, что десятичная точка имеет способность "плавать".

Существует три различных типа данных с плавающей точкой: float, double, и long double.

```
float    a = 1.5;  
long     b = 100.001;  
double   c = 200.1;
```

В большинстве современных архитектур, float равен 4 байтам, double равен 8, и long double может быть равен типу double (8 байт), или 16 байт.

Числа с плавающей точкой всегда являются знаковыми, что значит, что они могут хранить и положительные и отрицательные значения.



Тема: Типы данных, строки. Оператор switch, цикл for.

Символьный тип

Переменная типа char хранит целое число размером 1 байт. Однако, вместо интерпретирования значения типа char как целого, значение переменной типа char интерпретируется как символ набора ASCII.

American Standard Code for Information Interchange (ASCII) это

таблица, которая используется для представления текста на компьютерах.

| Char | ASCII | Decimal | Bits | Char | ASCII | Decimal | Bits | Char | ASCII | Decimal | Bits |
|------|-------|---------|--------|------|-------|---------|--------|------|-------|---------|--------|
| 0 | 48 | 0 | 000000 | F | 70 | 22 | 010110 | d | 100 | 44 | 101100 |
| 1 | 49 | 1 | 000001 | G | 71 | 23 | 010111 | e | 101 | 45 | 101101 |
| 2 | 50 | 2 | 000010 | H | 72 | 24 | 011000 | f | 102 | 46 | 101110 |
| 3 | 51 | 3 | 000011 | I | 73 | 25 | 011001 | g | 103 | 47 | 101111 |
| 4 | 52 | 4 | 000100 | J | 74 | 26 | 011010 | h | 104 | 48 | 110000 |
| 5 | 53 | 5 | 000101 | K | 75 | 27 | 011011 | i | 105 | 49 | 110001 |
| 6 | 54 | 6 | 000110 | L | 76 | 28 | 011100 | j | 106 | 50 | 110010 |
| 7 | 55 | 7 | 000111 | M | 77 | 29 | 011101 | k | 107 | 51 | 110011 |
| 8 | 56 | 8 | 001000 | N | 78 | 30 | 011110 | l | 108 | 52 | 110100 |
| 9 | 57 | 9 | 001001 | O | 79 | 31 | 011111 | m | 109 | 53 | 110101 |
| : | 58 | 10 | 001010 | P | 80 | 32 | 100000 | n | 110 | 54 | 110110 |
| ; | 59 | 11 | 001011 | Q | 81 | 33 | 100001 | o | 111 | 55 | 110111 |
| < | 60 | 12 | 001100 | R | 82 | 34 | 100010 | p | 112 | 56 | 111000 |
| = | 61 | 13 | 001101 | S | 83 | 35 | 100011 | q | 113 | 57 | 111001 |
| > | 62 | 14 | 001110 | T | 84 | 36 | 100100 | r | 114 | 58 | 111010 |
| ? | 63 | 15 | 001111 | U | 85 | 37 | 100101 | s | 115 | 59 | 111011 |
| @ | 64 | 16 | 010000 | V | 86 | 38 | 100110 | t | 116 | 60 | 111100 |
| A | 65 | 17 | 010001 | W | 87 | 39 | 100111 | u | 117 | 61 | 111101 |
| B | 66 | 18 | 010010 | ' | 96 | 40 | 101000 | v | 118 | 62 | 111110 |
| C | 67 | 19 | 010011 | a | 97 | 41 | 101001 | w | 119 | 63 | 111111 |
| D | 68 | 20 | 010100 | b | 98 | 42 | 101010 | | | | |
| E | 69 | 21 | 010101 | c | 99 | 43 | 101011 | | | | |





Тема: Типы данных, строки. Оператор switch, цикл for.

Символьный тип

Символы представляются закрытыми между одиночными кавычками (например 'a', 'b', и т.д.).

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      char character = 'c';
8      cout << character << endl;
9
10     return (0);
11 }
```

'=' '&' '@'

'^' '&' '@'

'+' '#' '('

'%' '*'



Тема: Типы данных, строки. Оператор switch, цикл for.

Строки

Строка -это упорядоченная последовательность символов, заключенная в двойные кавычки.

Строки являются частью Стандартной Библиотеки.

Вам необходимо включить библиотеку `<string>` чтобы использовать тип данных `string`. Либо вы можете использовать библиотеку, которая включает библиотеку с типом данных `string`.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      string str = "I am learning C++";
8      cout << str << endl;
9      cout << str[0] << endl;
10     cout << str[1] << endl;
11     cout << str[2] << endl;
12     cout << str[3] << endl;
13
14     return 0;
15 }
```

Библиотека `<string>` включена в библиотеку `<iostream>`, так что вам не нужно отдельно включать `<string>`, если вы уже включили `<iostream>`.





Тема: Типы данных, строки. Оператор switch, цикл for.

Булевый тип

Переменные булевого типа имеют только два возможных значения: true(истина) (1) и false(ложь) (0).

Для объявления булевой переменной используйте ключевое слово **bool**.

Если Булево значение присвоено целому числу, то истина становится 1, а ложь становится 0.

Если целое значение присвоено Булевому, то 0 становится false (ложью) и любое значение не равное нулю становится true (истинным).

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      bool a = 1;
8      bool b = false;
9      bool c = 123;
10     cout << a << endl;
11     cout << b << endl;
12     cout << c << endl;
13
14     return 0;
15 }
```



Тема: Типы данных, строки. Оператор switch, цикл for.

Правила наименования переменных

Используйте следующие правила при наименовании переменных:

- Все переменные должны начинаться с буквы алфавита или с нижнего подчеркивания (_).
- После первой буквы имя переменной может содержать дополнительные буквы, а также и числа. Пробелы или специальные символы использовать в имени переменных нельзя.

Чувствительность к регистру

C++ чувствителен к регистру, это означает, что идентификатор написанный в верхнем регистре не эквивалентен такому же другому, но в нижнем регистре. Например, **myvariable** это не тоже самое, что и **MYVARIABLE** и не то же самое, что и **MyVariable**.

Это три разные переменные!



Тема: Типы данных, строки. Оператор switch, цикл for.

Выбирайте имена переменных, которые подходят к использованию, например: `firstName`, `lastName`.

Правила наименования переменных

Ключевые слова C++ (зарезервированные слова) не могут быть использованы для наименования переменных.

Например, `int`, `float`, `double`, `cout` не могут быть использованы в качестве имени переменной.

Не существует предела длины имени переменных (в зависимости от оборудования), но старайтесь давать переменным практичные и осмысленные имена.



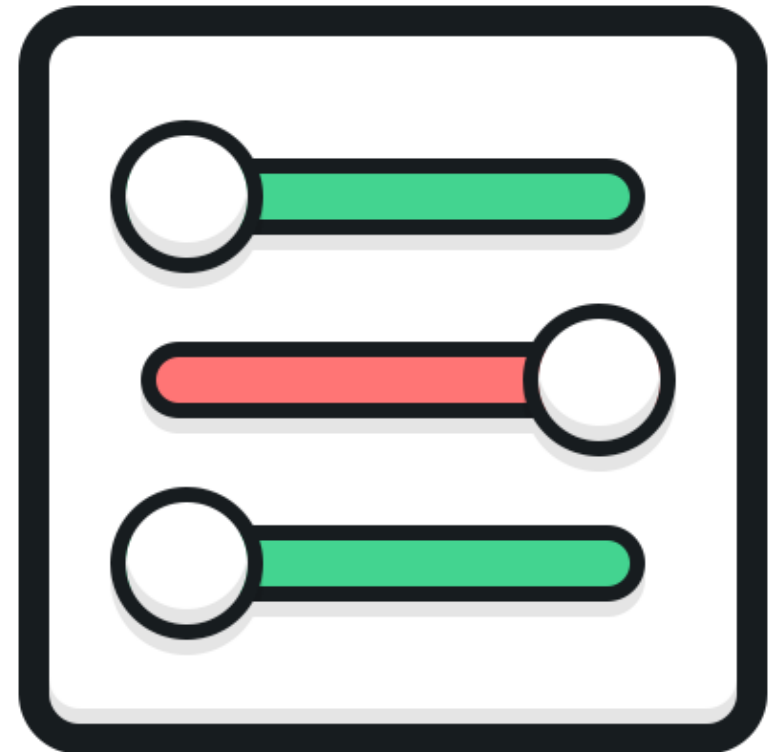
Тема: Типы данных, строки. Оператор switch, цикл for.

Множество условий

Иногда есть необходимость проверить переменную на равенство множеству значений. Это может быть достигнуто с помощью множества операторов if.

Ситуации, в которых необходимо использование нескольких операторов if, возникают часто, в связи с этим было принято решение о создании нового оператора.

Оператор множественного условия ветвления switch (переключатель)



Тема: Типы данных, строки. Оператор switch, цикл for.

Оператор switch

Оператор switch проверяет переменную со списком значений, которые называются cases(случай), для определения которому из них оно равно.

Оператор switch вычисляет выражение для определения, которому из списка случаев оно соответствует. При нахождении соответствия выполняются выражения данного случая.

Оператор switch может содержать любое количество операторов case, за которыми следует значение и двоеточие

```
switch (expression)
{
    case value1:
        statement(s);
        break;
    case value2:
        statement(s);
        break;
    ...
    case valueN:
        statement(s);
        break;
}
```



Тема: Типы данных, строки. Оператор switch, цикл for.

Сравнение if-else и switch

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int age = 42;
7      if (age == 16)
8      {
9          cout << "Too young";
10     }
11     else if (age == 42)
12     {
13         cout << "Adult";
14     }
15     else if (age == 70)
16     {
17         cout << "Senior";
18     }
19     return (0);
20 }
```

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int age = 42;
7      switch (age)
8      {
9          case 16:
10             cout << "Too young";
11             break;
12          case 42:
13             cout << "Adult";
14             break;
15          case 70:
16             cout << "Senior";
17             break;
18     }
19     return (0);
20 }
```



Тема: Типы данных, строки. Оператор switch, цикл for.

Случай по умолчанию (default case)

В операторе switch, может быть использован случай по умолчанию, если ни один из случаев не был реализован.

Код оператора default выполняется, если ни один из случаев не соответствует выражению оператора switch.

Оператор default должен находиться в конце оператора switch.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int age = 3;
7      cout << "Output:" << endl;
8      switch (age)
9      {
10         case 16:
11             cout << "Too young";
12             break;
13         case 42:
14             cout << "Adult";
15             break;
16         case 70:
17             cout << "Senior";
18             break;
19         default:
20             cout << "This is default case" << endl;
21     }
22     return (0);
23 }
```

Output:
This is default case



Тема: Типы данных, строки. Оператор switch, цикл for.

Оператор break

Роль оператора break - закончить оператор switch.
В образцах, в которых переменная соответствует значению в case, выражение, которое следует после оператора case, продолжает выполняться до оператора break. Другими словами, если опустить оператор break будут выполняться все выражения в следующих операторах case, даже те, которые не соответствуют выражению switch.

Оператору default не требуется оператор break, т.к. он является последним оператором case.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int age = 16;
7      cout << "Output:" << endl;
8      switch (age)
9      {
10         case 16:
11             cout << "Too young" << endl;
12         case 42:
13             cout << "Adult" << endl;
14         case 70:
15             cout << "Senior" << endl;
16         default:
17             cout << "This is default case" << endl;
18     }
19     return (0);
20 }
```

Output:
Too young
Adult
Senior
This is default case

Тема: Типы данных, строки. Оператор switch, цикл for.

Цикл for

Цикл for является повторяющимся управляющим выражением, которое позволяет вам эффективно создать цикл, повторяющийся специальное количество раз.

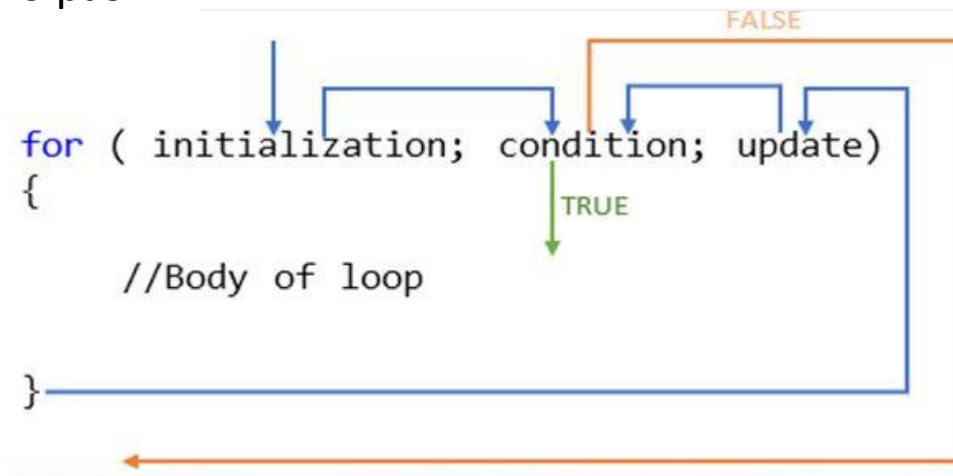
Синтаксис:

Шаг init выполняется первым и больше не повторяется.

Далее, высчитывается условие и, если оно верно, выполняется тело цикла.

Следующим шагом инкрементирующее выражение обновляет переменную, управляющую циклом.

```
for (init; condition; increment)  
{  
    statements;  
}
```



Затем, повторяется тело цикла, останавливаясь только в случае, если условие станет ложным.

Тема: Типы данных, строки. Оператор switch, цикл for.

Цикл for

В описанном ниже примере используется цикл for для вывода чисел от 0 до 9.

В шаге init мы объявили переменную a и присвоили ей число 0. a < 10 является условием. После каждой итерации, выполняется выражение инкремента a++. При увеличении переменной a до 10 условие становится ложным и цикл останавливается.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      for (int a = 0; a < 10; a++)
8      {
9          cout << a << endl;
10     }
11
12     return 0;
13 }
```





Тема: Типы данных, строки. Оператор switch, цикл for.

Цикл for

Есть возможность изменять инкрементирующее выражение.

```
for (int a = 0; a < 50; a += 10)
{
    cout << a << endl;
}
```

```
for (int a = 10; a >= 0; a -= 3)
{
    cout << a << endl;
}
```

Вы также можете использовать декремент.

При использовании цикла for не забывайте про точки с запятыми после выражений init(инициализация) и условие.



Программирование
на C++

Практическая часть

Типы данных, строки.
Оператор switch, цикл for.

3 занятие



Тема: Типы данных, строки. Оператор switch, цикл for.

Задание 1

Найти сумму чисел от 1 до 100.

Тема: Типы данных, строки. Оператор switch, цикл for.

Задание 2

Реализуйте таблицу умножения с помощью цикла for.

Тема: Типы данных, строки. Оператор switch, цикл for.

Задание 3

Напишите программу-калькулятор. Пользователь вводит 2 числа и выбирает арифметический оператор, ваша программа выдаёт результат.

*

Добавьте поясняющие предложения
Например, “Please, enter first number”,

**

Проверка на ноль!