

Занятие 3

1. Оператор множественного выбора switch

Оператор множественного выбора имеет вид

```
switch (/*переменная или выражение*/)
{
    case /*константное выражение 2/*:
        /* операторы */;
        break;
}
{
    case /*константное выражение 2/*:
        /* операторы */;
        break;
}
default:
{
    /* операторы */;
}
}
```

Пример. Оператор выбора

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

int main()
{
    setlocale(0, "");
    srand(time(NULL));
    int a = rand() % 3 + 1;
    int b = rand() % 2;
    switch (a+b)
    {
        case 1:
        {
            cout << 1 << endl;
            break;
        }
        case 2:
        {
            cout << 2 << endl;
            break;
        }

        default:
            cout << 10*a << endl;
    }
    return 0;
}
```

Решите задачи:

- По оценке (2-5) выведите ее словесное описание.
- По номеру координатной четверти выведите знаки координат точки.

2. Числовые типы данных

- **short**: представляет целое число в диапазоне от -32767 до 32767 . Занимает в памяти 2 байта (16 бит). Данный тип также имеет синонимы `short int`, `signed short int`, `signed short`.
- **unsigned short**: представляет целое число в диапазоне от 0 до 65535. Занимает в памяти 2 байта (16 бит). Данный тип также имеет синоним `unsigned short int`.
- **int**: представляет целое число. В зависимости от архитектуры процессора может занимать 2 байта (16 бит) или 4 байта (32 бита). Диапазон предельных значений соответственно также может варьироваться от -32767 до 32767 (при 2 байтах) или от $-2\,147\,483\,647$ до $2\,147\,483\,647$ (при 4 байтах). Но в любом случае размер должен быть больше или равен размеру типа `short` и меньше или равен размеру типа `long`. Данный тип имеет синонимы `signed int` и `signed`.
- `unsigned int`: представляет положительное целое число. В зависимости от архитектуры процессора может занимать 2 байта (16 бит) или 4 байта (32 бита), и из-за этого диапазон предельных значений может меняться: от 0 до 65535 (для 2 байт), либо от 0 до $4\,294\,967\,295$ (для 4 байт). В качестве синонима этого типа может использоваться `unsigned`
- `long`: представляет целое число в диапазоне от $-2\,147\,483\,647$ до $2\,147\,483\,647$. Занимает в памяти 4 байта (32 бита). У данного типа также есть синонимы `long int`, `signed long int` и `signed long`
- `unsigned long`: представляет целое число в диапазоне от 0 до $4\,294\,967\,295$. Занимает в памяти 4 байта (32 бита). Имеет синоним `unsigned long int`.
- `long long`: представляет целое число в диапазоне от $-9\,223\,372\,036\,854\,775\,807$ до $+9\,223\,372\,036\,854\,775\,807$. Занимает в памяти, как правило, 8 байт (64 бита). Имеет синонимы `long long int`, `signed long long int` и `signed long long`.
- `unsigned long long`: представляет целое число в диапазоне от 0 до $18\,446\,744\,073\,709\,551\,615$. Занимает в памяти, как правило, 8 байт (64 бита). Имеет синоним `unsigned long long int`.
- `float`: представляет вещественное число ординарной точности с плавающей точкой в диапазоне $\pm 3.4E-38$ до $3.4E+38$. В памяти занимает 4 байта (32 бита)
- `double`: представляет вещественное число двойной точности с плавающей точкой в диапазоне $\pm 1.7E-308$ до $1.7E+308$. В памяти занимает 8 байт (64 бита)
- `long double`: представляет вещественное число двойной точности с плавающей точкой не менее 8 байт (64 бит). В зависимости от размера занимаемой памяти может отличаться диапазон допустимых значений.
- `void`: тип без значения

2

Команда `sizeof(тип)` может быть использована для определения количества байт, которые отводятся для хранения типа данных:

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

int main()
{
    cout << "int = " << sizeof(int) << endl;
    cout << "short int = " << sizeof(short int) << endl;
    cout << "long int = " << sizeof(long int) << endl;
    cout << "long long int = " << sizeof(long long int) << endl;
    cout << "unsigned long int = " << sizeof(unsigned long int) << endl;
    return 0;
}
```

Задание. Выведите на экран количество бит, которые отводятся для хранения типа `double`.

3. Форматированный вывод данных

В библиотеке `iomanip` существует несколько функций, которые позволяют задавать форматированный вывод на экран: `setfill` – заполняет пустое пространство символом, `setw` – задает ширину поля для вывода, `setprecision` – устанавливает количество знаков после десятичной запятой, `fixed` – вывод чисел с плавающей точкой в фиксированной форме (по умолчанию), `right` – выравнивание по правому краю.

```
#include <iostream>
// библиотека манипулирования вводом/выводом
#include <iomanip>
using namespace std;
int main()
{
    double a = 7.1234;
    cout << setfill('~')
         << setw(10)
         << fixed
         << setprecision(5)
         << a << endl;
    return 0;
}
```

Задание. Выведите на печать таблицу значений $\sin(x)$ и аргумента для значений x от 1 до 7.

4. Преобразование типов данных

В C++ существует явное и неявное преобразование типов данных. Неявное преобразование происходит во время компиляции программы, например, когда целое делим на целое, то дробная часть отбрасывается, и в ответе получаем целое.

Явное преобразование можно производить двумя способами:

```
(double)7
static_cast<double>(7)
```

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

int main()
{
    // Неявное преобразование типов
    cout << 7/2 << endl;

    //cout << 7.0 / 2 << endl;
    //cout << 7 / 2.0 << endl;

    // Явное преобразование типов. 2 способа
    cout << (double)7 / 2 << endl;
    cout << 7 / (double)2 << endl;

    cout << static_cast<double>(7) / 2 << endl;
    cout << 7 / static_cast<double>(2) << endl;
    return 0;
}
```

Задание. В цикле генерируйте вещественные случайные числа от 10 до 20 с тремя знаками после запятой. Выведите на экран таблицу чисел – вещественных и этих же чисел, но преобразованных к целому типу.

5. Время выполнения программы

Для определения времени исполнения программы служит функция **clock()** библиотеки **ctime**.

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

int main()
{
    unsigned int start = clock();
    for (double i = 1; i < 1.0E+4; ++i) { cout << i; }
    unsigned int end = clock();
    cout << "time = " << (end-start) / 1000.0 << endl;
    return 0;
}
```

Выполнить следующие расчеты:

- Определите время генерации *большого количества* случайных чисел в диапазоне от 1000 до 1001.
- Определите - как отличается операция `i++` от `++i` по временной характеристике.
- Определите - как отличается операция умножения на 2 от сдвига вправо для большого количества случайных чисел. Напишите код для генерации случайных чисел такой, чтобы для первой и второй операции числа были одинаковыми.

Задания для закрепления пройденного материала

- Случайным образом задавайте целое число в диапазоне от 1 до 6 (количество точек на грани игрального кубика). В зависимости от выпавшего числа выведите его словесное описание.
- Случайным образом задавайте целое число от 0 до 1 (орел или решка при бросании монеты). Выполните опыт 100 раз. Посчитайте количество выпавших орлов и решек. Считайте, что 100 раз бросания монеты – это один опыт. Повторите опыт 1000 раз (с помощью внешнего цикла). Посчитайте и выведите на экран процент выпадения орла.
- Повторите задание 2 для игрального кубика. Посчитайте и выведите на экран процент выпадения числа 5.
- Выведите таблицу основных числовых типов данных и количество бит, которые они занимают.
- Сформируйте таблицу умножения целых чисел от 1 до 20.
- Определите, как отличается время исполнения команды **cout** для целого и вещественного чисел. Проверку выполняйте для большого количества операций.
- Определите, как отличается время возведения во вторую степень целых чисел от умножения числа на тоже число. Проверку выполняйте для большого количества операций.
- Определите, как отличается время сложения десяти целых одинаковых чисел от умножения на 10. Проверку выполняйте для большого количества операций.

Контрольные вопросы

1. Что произойдет, если в операторе switch не использовать оператор break?
2. Какую библиотеку нужно подключить, чтобы можно было использовать функцию setw?
3. Каким образом можно задать количество позиций после десятичной точки для вывода вещественных чисел на экран?
4. Какая функция используется для определения времени выполнения программы?
5. Исправьте ошибку в следующей строке: `cout << <double>7 / 2 endl;`
6. Что будет выведено на экран после выполнения команды: `cout << 10/3 << endl;`
7. Какие способы преобразования типов данных существуют в C++?
8. Что произойдет, если в переменную типа `unsigned int` записать отрицательное число и вывести эту переменную на экран?
9. Проанализируйте код, приведенный ниже, и найдите ошибку:

```
int s;  
for (double i = 1.0; i <= 10; i++) { s += i; }  
cout << s << endl;
```

10. Проанализируйте код, приведенный ниже, и найдите ошибку:

```
unsigned int n1=clock();  
for (double i = 1.0; i <= 1.0E-12; i++) { }  
unsigned int n2 = clock();  
cout << n1-n2 << endl;
```