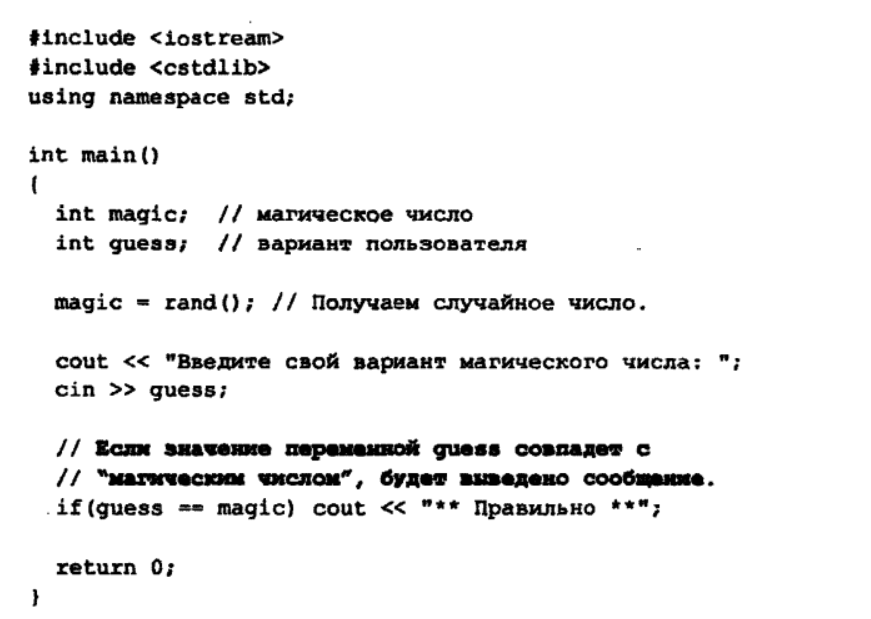
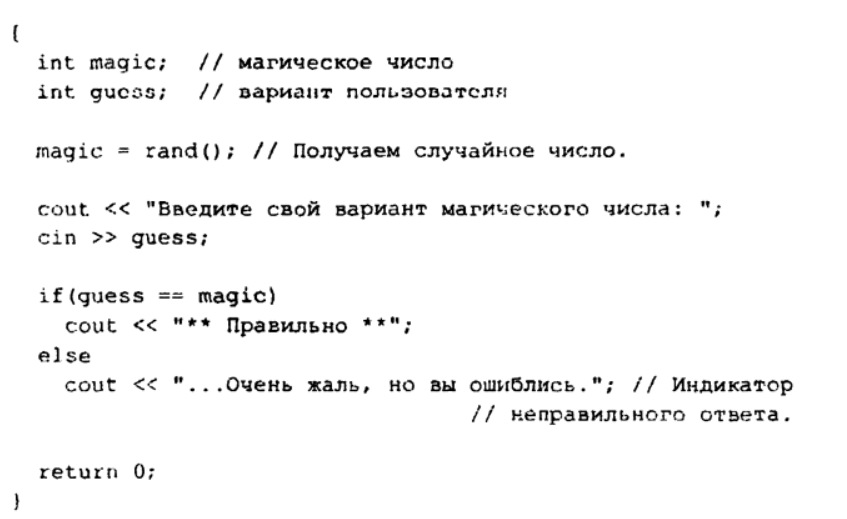
| Таблица 1 — Типы данных С++ | | |
| --- | --- | --- |
| **Тип** | **байт** | **Диапазон принимаемых значений** |
| целочисленный (логический) тип данных | | |
| bool | 1 | 0   /   255 |
| целочисленный (символьный) тип данных | | |
| char | 1 | 0   /   255 |
| целочисленные типы данных | | |
| short int | 2 | -32 768    /    32 767 |
| unsigned short int | 2 | 0  /  65 535 |
| int | 4 | -2 147 483 648   /   2 147 483 647 |
| unsigned int | 4 | 0     /     4 294 967 295 |
| long int | 4 | -2 147 483 648    /    2 147 483 647 |
| unsigned long int | 4 | 0     /     4 294 967 295 |
| типы данных с плавающей точкой | | |
| float | 4 | -2 147 483 648.0  / 2 147 483 647.0 |
| long float | 8 | -9 223 372 036 854 775 808 .0   /   9 223 372 036 854 775 807.0 |
| double | 8 | -9 223 372 036 854 775 808 .0   /   9 223 372 036 854 775 807.0 |

| Таблица 1 — Типы данных С++ | | |
| --- | --- | --- |
| **Тип** | **байт** | **Диапазон принимаемых значений** |
| целочисленный (логический) тип данных | | |
| bool | 1 | 0   /   255 |
| целочисленный (символьный) тип данных | | |
| char | 1 | 0   /   255 |
| целочисленные типы данных | | |
| short int | 2 | -32 768    /    32 767 |
| unsigned short int | 2 | 0  /  65 535 |
| int | 4 | -2 147 483 648   /   2 147 483 647 |
| unsigned int | 4 | 0     /     4 294 967 295 |
| long int | 4 | -2 147 483 648    /    2 147 483 647 |
| unsigned long int | 4 | 0     /     4 294 967 295 |
| типы данных с плавающей точкой | | |
| float | 4 | -2 147 483 648.0  / 2 147 483 647.0 |
| long float | 8 | -9 223 372 036 854 775 808 .0   /   9 223 372 036 854 775 807.0 |
| double | 8 | -9 223 372 036 854 775 808 .0   /   9 223 372 036 854 775 807.0 |

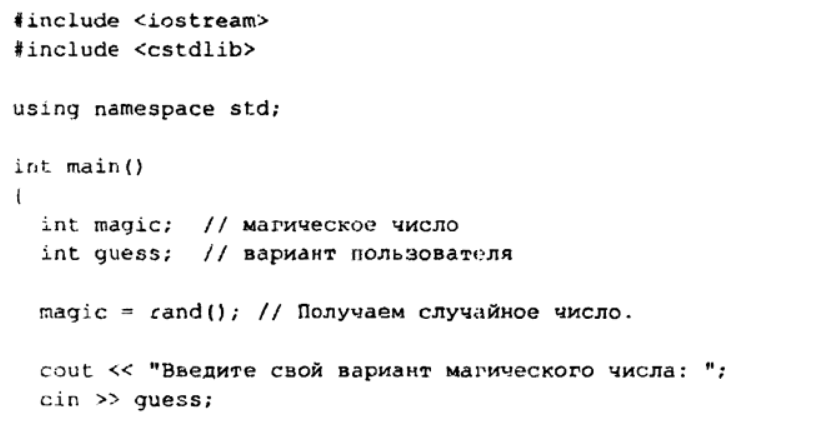
1. «Угадай магическое число»

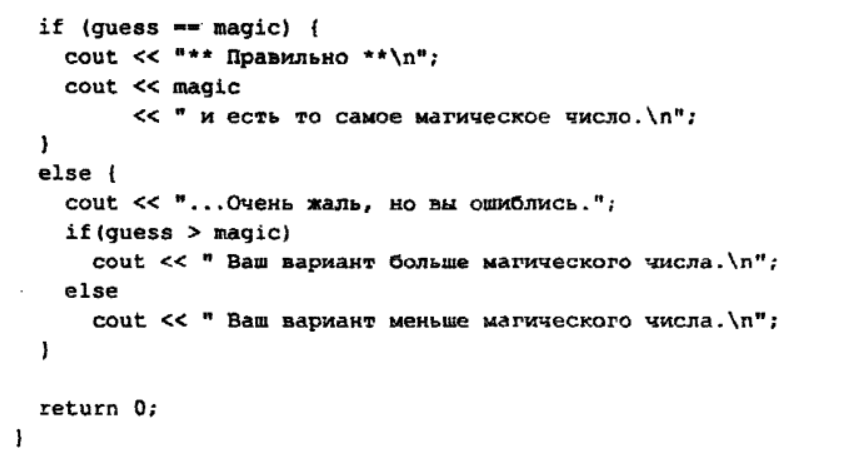


1. Усовершенствуем «угадай магическое число»



1. 2 Усовершенствование «угадай магическое число»





1. Пример использования switch

int chisl;

cout<<'введите цифру от 1 до 5\n';

cin>>chisl;

switch (chisl)

{

   case 1: // не совпадает - пропускается

       cout << 1 << '\n';

       break;

   case 2:

       cout << 2 << '\n'; // выполнение начинается здесь

       break; // оператор break завершает выполнение switch

   case 3:

       cout << 3 << '\n';

       break;

   case 4:

       cout << 4 << '\n';

       break;

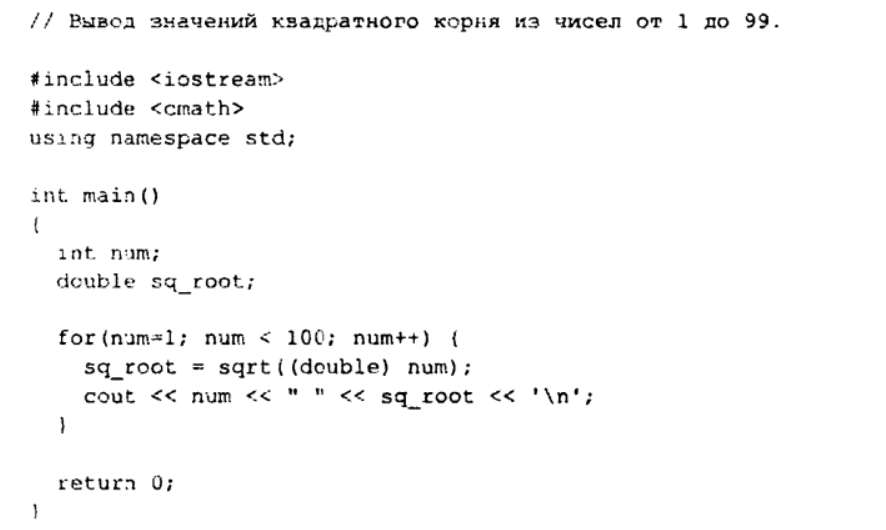
   default:

       cout << 5 << '\n';

       break;

}

1. Цикл for



1. Цикл while

**#include <iostream>**

**using** **namespace** std;

**int** **main**() {

**int** i = 0; *// инициализируем счетчик цикла.*

**int** sum = 0; *// инициализируем счетчик суммы.*

**while** (i < 1000) {

i++; sum += i;

}

cout << "Сумма чисел от 1 до 1000 = " << sum << endl;

**return** 0;

}

1. Цикл do-while

**#include <iostream>**

**using** **namespace** std;

**int** **main** () {

**int** i = 0; *// инициализируем счетчик цикла.*

**int** sum = 0; *// инициализируем счетчик суммы.*

**do** {*// выполняем цикл.*

i++; sum += i;

} **while** (i < 1000); *// пока выполняется условие.*

cout << "Сумма чисел от 1 до 1000 = " << sum << endl;

**return** 0;

}

**Задачи**

**1. Составьте программу, выводящую на экран квадраты чисел от 10 до 20 включительно.**

**2. Даны натуральные числа от 35 до 87. Вывести на консоль те из них, которые при делении на 7 дают остаток 1, 2 или 5.**

**3. Найдите сумму**1+2+3+…+n**, где число**n**вводится пользователем с клавиатуры.**

**4. Найдите произведение цифр трехзначного числа.**

**5. Найдите количество четных цифр данного натурального числа.**

**6. Найдите наибольшую цифру данного натурального числа.**

**7. Найдите все четырехзначные числа, сумма цифр каждого из которых равна 15.**

**8. Написать игру. Игра будет следующей — компьютер загадывает число в диапазоне, заданном пользователем. Пользователь пытается угадать число, при этом компьютер сообщает больше загаданное число или меньше. Кроме того, программа выводит минимальное количество шагов, за которое можно было отгадать. И конечно же предлагает сыграть еще.**