

C++

Урок 2

A hand-drawn blue oval frame with a slightly irregular, sketchy border, centered on the page. The word "Повторение" is written inside this frame in a bold, black, sans-serif font.

Повторение

Из чего состоит программа?

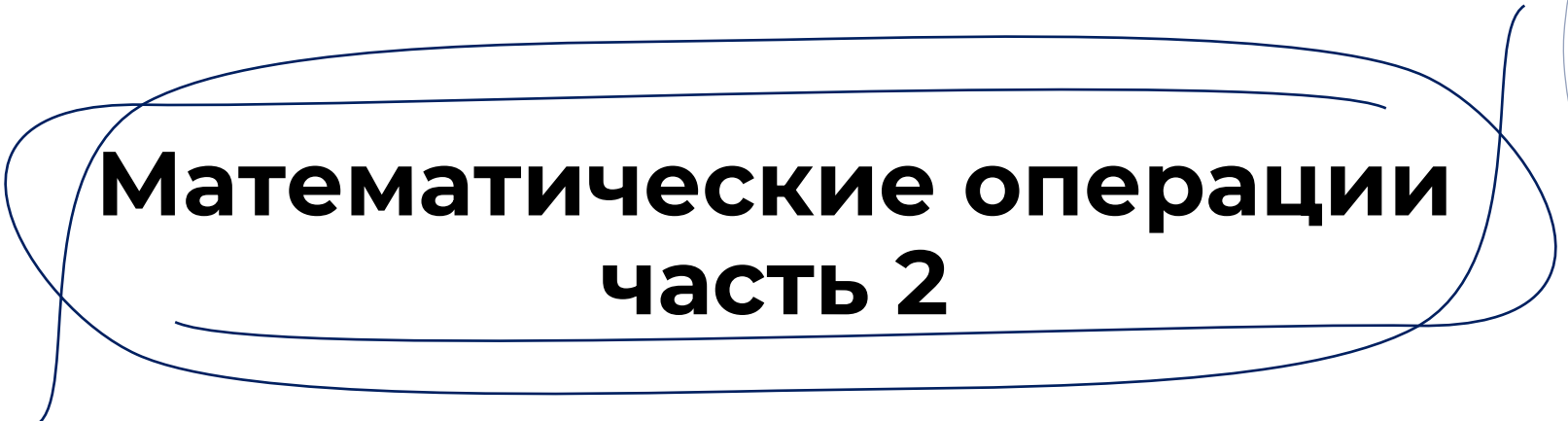
**С помощью какой библиотеки
реализуется ввод и вывод?**



Что такое переменная?

A hand-drawn blue oval frame with a double-line border, centered on the page. The text "Разбор ДЗ" is written inside this frame in a bold, black, sans-serif font.

Разбор ДЗ

A hand-drawn blue oval frame surrounds the text. The frame is composed of two overlapping loops, with the top loop being slightly larger and more defined than the bottom one. The lines are thin and have a slightly irregular, hand-drawn quality.

Математические операции

часть 2

Особенности операций

Сокращенное присваивание

```
int q = 5;  
var = var + q;  
var+=q; // сокращенное сложение  
  
var = var - q;  
var-=q; // сокращенное вычитание  
var = var / q;  
var/=q; // сокращенное деление  
  
var = var * q;  
var*=q; // сокращенное умножение
```

Исключаем дублирование кода

Особенности операций

Сокращенное присваивание

```
int sum, a, b, c, d;  
sum = 0;  
a = 5;  
sum = sum + a; // 5  
b = 4;  
sum = sum + b; // 4 + 5 = 9  
c = 8;  
  
sum += c; // 9 + 8 = 17  
d = 8;  
sum += d; // 17 + 8 = 25
```

Исключаем дублирование кода

**Кто знает библиотеку, которая расширяет
возможности использования математики?**

Библиотека cmath

```
#include <iostream>
// библиотека ввода вывода
#include <cmath>
// использование математических операций
#include <iomanip>
// для фиксированного вывода
```

#include <cmath> // математические операции

+

#include <iomanip> // для ограничения кол-ва знаков при выводе

Библиотека cmath

```
#include <iostream>
// библиотека ввода вывода
#include <cmath>
// использование математических операций
#include <iomanip>
// для фиксированного вывода
```

#include <cmath> // математические операции

+

#include <iomanip> // для ограничения кол-ва знаков при выводе

Для более сложных математических
действий

Библиотека cmath

Округления

round	Округляет число по правилам арифметики, то есть $\text{round}(1.5) == 2$, $\text{round}(-1.5) == -2$
floor	Округляет число вниз ("пол"), при этом $\text{floor}(1.5) == 1$, $\text{floor}(-1.5) == -2$
ceil	Округляет число вверх ("потолок"), при этом $\text{ceil}(1.5) == 2$, $\text{ceil}(-1.5) == -1$
fabs	Модуль (абсолютная величина)

Корни и степени

sqrt	Квадратный корень. Использование: $\text{sqrt}(x)$
cbrt	Кубический корень. Использование: $\text{cbrt}(x)$
pow	Возведение в степень, возвращает a^b . Использование: $\text{pow}(a,b)$

Тригонометрия

sin	Использование: $\text{sin}(x)$
cos	Использование: $\text{cos}(x)$
tan	Использование: $\text{tan}(x)$

cmath + iomanip

Код программы

```
double a = 34.56;  
a = fabs(a); //модуль числа  
  
double rez = sqrt(a); //квадратный  
double rez2 = cbrt(a); //кубический корень  
double rez3 = pow(a,8); //возведение в степень  
double rez4 = pow(a,(1.0/8)); // КОРНИ  
  
long double y = 1/3.0; //преобразование к вещ типу операнд  
cout <<fixed<<setprecision(6)<<y<<endl;  
//вывод с ограничением на кол-во цифр после запятой
```

Вывод с точностью до 6 знаков после запятой

Задача


Пользователь вводит в консоль число (x);

Задача: Посчитайте значение выражения $(x^2 + \sin(x) - \sqrt{x}) / x^{-2}$ и выведите значение **с точностью до 5 знаков после запятой**.

Задача

Пользователь вводит в консоль число (x);

Задача: Посчитайте значение выражения $(x^2 + \sin(x) - \sqrt{x}) / x^{-2}$ и выведите значение, округлив результат в **большую сторону**.



**Что такое логическое
выражение?**

Логические выражения

Логическое выражение – высказывание, о котором можно сказать **ИСТИННО** оно или **ЛОЖНО**.

Пример:

- Автобус тяжелее человека – истина;
- Йогурт состоит из молока – истина;
- У человека есть жабры – ложь;

Приведите свой пример

Вопрос

Является ли это логическим выражением:

----- Сейчас понедельник?

Вопрос

Является ли это логическим выражением:

----- Сейчас понедельник? **НЕТ**

Логические выражения

Вопросительные предложения **не являются**
логическими высказываниями

Операции отношения

Операции отношения

```
a > b  
a < b;  
a >= b;  
a <= b;  
a == b;  
a != b;
```

Где a, b – целые числа, которые вводит пользователь

Результат:

истина (1) или ложь (0);

Операции отношения

```
a > b  
a < b;  
a >= b;  
a <= b;  
a == b;  
a != b;
```

Где a, b – целые числа, которые вводит пользователь

Важно:

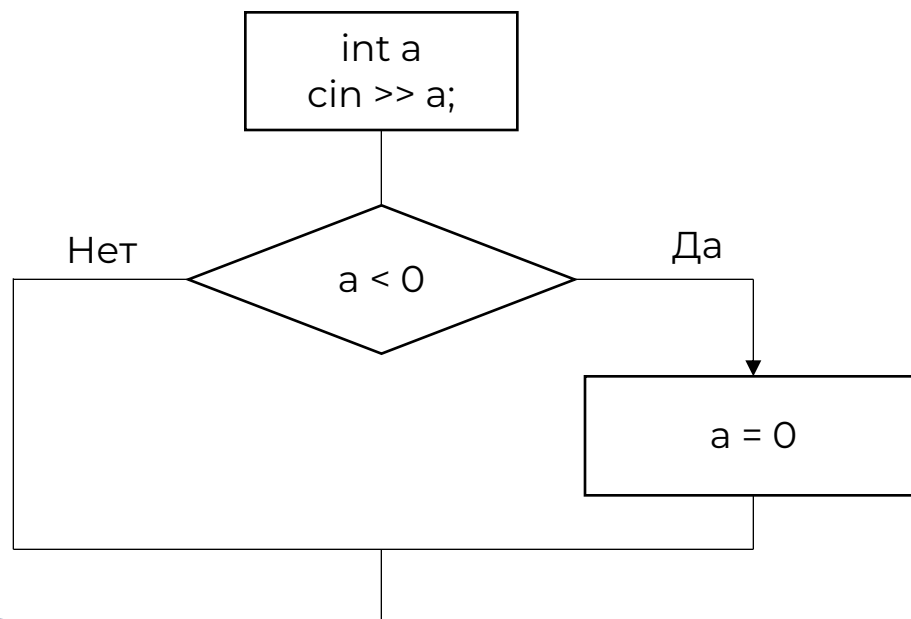
сравнение двух чисел это == (два знака равенства);

Условный оператор if

Условный оператор if

if – оператор, позволяющий реализовывать конструкции:

ЕСЛИ (условие сравнения), **ТО** (операция)



Условный оператор if

Формат:

```
if (условие){  
    действие  
}
```

Условный оператор if

Пример:

```
if (42 > 40){  
    cout <<"Больше"<<endl;  
}
```

Условный оператор if

Пример:

```
int a;  
cin >> a;  
  
if (a == 42){  
    cout <<"Введено 42"<<endl;  
}
```

Условный оператор if

Пример:

```
int a;  
cin >> a;  
  
if (a > 42){  
    cout <<"Введено число больше 42"<<endl;  
}  
  
if (a < 42){  
    cout <<"Введено число меньше 42"<<endl;  
}
```

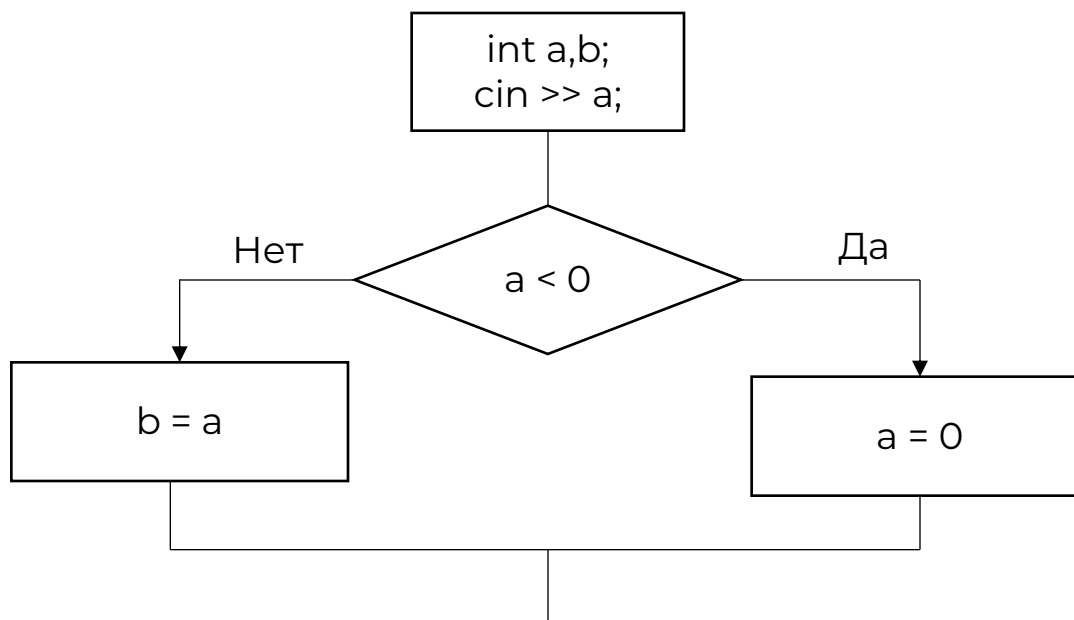
Задача

Пользователь вводит в консоль число;

Задача: проверить, что введенное число кратно 5;

Условный оператор if + else

else – дополняет if ситуацией **иначе**:



Условный оператор if + else

Пример:

```
if (a > 42){  
    cout <<"Введено число больше 42"<<endl;  
}  
else {  
    cout <<"Введено число меньше 42"<<endl;  
}
```

Исключаем дублирование кода

Задача

Пользователь вводит в консоль число;

Задача: проверить, что введенное число четное (вывести “Да”), если нет, то вывести (“Нет”);

Логические операции

Логические операции

Мы не общаемся только простыми предложениями, но обычно объединяем их в составные.

Так же и в программировании.

Простые высказывания объединяются в сложные путем логических операторов

Логические операции

Виды:

and or not

and – оператор конъюнкции (**&&**);

or – оператор дизъюнкции (**||**);

not – оператор инверсии (отрицание) (**!**).

Применение: диапазоны значений

Логические операции

Конъюнкция (*)
Логическое **И**

0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкция (+)
Логическое **ИЛИ**

0	0	0
0	1	1
1	0	1
1	1	1

Инверсия
Логическое **НЕ**

0	1
1	0

Где можно использовать?

Интервалы и полуинтервалы

```
int val;  
cin >> val;  
// [2, 8] диапазон  
if (val >= 2 && val <= 8){  
    cout << "Число принадлежит промежутку";  
}  
else {  
    cout << "Не принадлежит";  
}
```

Задача

Пользователь вводит в консоль число;

Задача: проверить, что введенное число входит в данные промежутки, интервалы и полуинтервалы $(-4; 9) \cup \{18\} \cup (65; 90]$;

Булевы переменные

Булевы переменные

- Принимают 2 значения – **True (1)** или **False (0)**;

Применение: хранение значения логических выражений.

Булевы переменные

Вспомним данный пример:

```
int val;  
cin >> val;  
// [2, 8] диапазон  
if (val >= 2 && val <= 8){  
    cout << "Число принадлежит промежутку";  
}  
else {  
    cout << "Не принадлежит";  
}
```

**Где можно использовать
булеву переменную?**

Булевы переменные

Реализация через булеву переменную:

```
int val;  
cin >> val;  
bool flag = (val >= 2 && val <= 8);  
if (flag) cout << "Число принадлежит промежутку";  
//то же самое что и if (flag != 0)  
else cout << "Не принадлежит";  
}
```

Задача

Пользователь вводит в консоль число;

Задача: проверить, что введенное число входит в данные промежутки, интервалы и полуинтервалы $(-4; 9) \cup \{18\} \cup (65; 90]$; (**через булеву переменную**)

**Что такое тернарный
оператор?**

Тернарный оператор

тернарный оператор

Формат:

(условие) ? если истина : если ложь;

```
int a;  
cin >> a;  
int rez = (a % 2 == 0)? a: a*(-1);
```

Применение: краткая форма записи if else

тернарный оператор

Для трех исходов:

```
(a > 0)? cout <<"Больше":(a == 0)? cout <<"Равно 0": cout <<"Меньше";
```

На выводе: больше, меньше или равно 0