

Методист курса: разработчик АО НПО "РусБиТех". Гудзенко Артем Валерьевич

Создание архива курса

- Весь код после занятия будет хранится на гите по этой ссыль https://github.com/Danykrane/EU_code_c
- При сдачи дз (можно мне в тг) необходимо указывать:
- Номер дз,
- Свое имя!
- Пример: "Высылаю дз1 Дима"

Структура курса по С++

Занятия: 64 часа – 32 занятия (3 модуля)

Домашние работы: 32 задания (12 + 12 + 8) на 3 модуля

1 модуль: Основы языка С++

2 модуль: ООП + шаблоны проектирования

3 модуль: Промышленная разработка.

Совместная Практика

Три/пять совместных заданий (можно выполнять и одному):

- На 5 занятии (квадратные матрицы)
- На 8 занятии (указатели и ссылки)
- На 14 занятии (Классы ООП)
- На 22 занятии (разработка проекта с определенным шаблоном проектирования)
- На 30 занятии (разработка фильтров для обработки изображения)

Из чего состоит программа на языке C++?

Программа на С++

Структура программы

```
#include <iostream> // библиотека ввода вывода
using namespace std;
int main() {
cout << "My name is - Artyom\nGlad to see you!\n";</pre>
// "\n" - символ переноса строки
cout << "My name is - Artyom"<<endl;</pre>
cout <<"Glad to see you!";</pre>
```

Структура программы

```
#include <iostream> // библиотека ввода вывода
```

- Подключение библиотек

```
using namespace std;
```

- Пространство имен

```
int main() {
// основной код тут
}
```

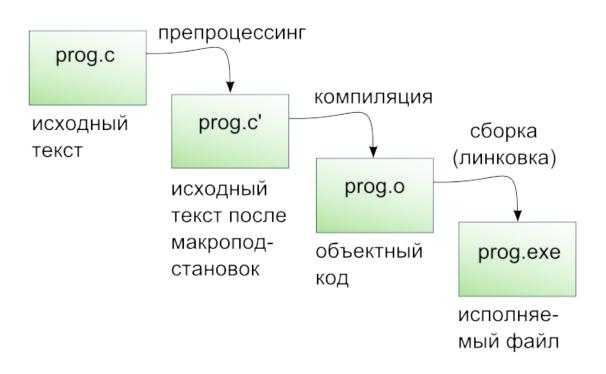
- Основная функция

Как проходит компиляция?

Компиляция

Компиляция программы

Этапы после того, как нажмем на *run code*



Вывод

#include <iostream> // библиотека ввода вывода

- Библиотека для операций ввода и вывода в консоль

I – input (ввод) o – output (вывод) Stream (поток)

iostream

Вывод

```
#include <iostream> // библиотека ввода вывода
using namespace std;
int main() {
cout << "My name is - Artyom"<<endl;
cout <<"Glad to see you!";
cout << "My name is - Artyom\nGlad to see you!\n";

// "\n" - символ переноса строки
}
```

с подключенным пространством имен std

```
#include <iostream> // библиотека ввода вывода
int main() {
  std::cout << "My name is - Artyom"<<std::endl;
  std::cout <<"Glad to see you!";

  std::cout << "My name is - Artyom\nGlad to see you!\n";

// "\n" - символ переноса строки
}</pre>
```

без подключения пространства имен std

Ввод

```
#include <iostream> // библиотека ввода вывода

using namespace std;

int main() {

int age; //создани переменной для хранения возраста
cin >> age; // ввод в переменую age значения с клавиатуры
}
```

Ввод + Вывод

```
#include <iostream> // библиотека ввода вывода

using namespace std;

int main() {

int age; //создани переменной для хранения возраста

cout<<"Сколько вам лет?"<<endl;

cin >> age; // ввод в переменую age значения с клавиатуры

cout << "Ваш возраст: "<<age<<endl;
}
```

Ввод + Вывод

Сколько вам лет?

22

Ваш <u>возраст</u>: 22

artemgudzenko@MacBook-Air-Artem 1_less %

Задача

Задача: выведите 10 раз строку *****. После вывода строки перенос на новую.

Что такое переменная?

Приведите пример из жизни

Переменная

Переменная

Переменная - это именованная область памяти, к которой мы имеем доступ из программы;

Типы данных



• Символы

• Целые числа

• Строки

• Неопределенный тип

• Вещественные числа

Для чего нужны разные типы данных?

Типы данных

1. Целые числа

Имя типа	Подтипы	Размер,	Интервал
ALIMI KIMIN	ПОДТИПЫ	· ·	•
		байт	значений
char или _int[8]	[signed] char] 1	-128127
	unsigned char		0255
short или _int[16]	[signed] short	2	-3276832767
	unsigned short		065535
int или long или	[signed] int	4	-2 ³¹ 2 ³¹ -1
_int[32]	unsigned int		0 2 ³² -1
	[signed] long		
	unsigned long		
long long или	[signed] long long	8	-2 ⁶³ 2 ⁶³ -1
_int[64]	unsigned long long		0 2 ⁶⁴ -1
bool		1	false (0), true(1)

Типы данных

2. Вещественные типы

Тип	Размер,	Значащих	Минимальное	Максимальное
	байт	цифр	положительное	положительное
			число	число
float	4	6	1.175494351e-38	3.402823466e38
double (long double)	8	15	2.2250738585072014 e-308	1.797693134862318 e308

3. Неопределенный («пустой») тип void

Для:

- нетипизированных указателей;
- функций.

Объявление переменных

Объявление переменных и констант

Формат:

Изменчивость Тип Имя = Значение;

Изменчивость – описатель возможности изменения значений:

- **const** константа неизменяемое значение,
- без указания обычная переменная

Тип – тип данных: int, char, float, double и т.д.;

Имя – само имя переменной или константы;

Значение – начальное значение переменной или значение константы.

Объявление переменных и констант

Примеры:

```
#include <iostream> // библиотека ввода вывода

using namespace std;

int main() {

int a, b; // объявили две целые переменные

float pi=3.14,k(5.45); // объявили + сразу инициализировали значения

const unsigned char letter='a'; // константа — код буквы «а»

int a=7; // объявили + инициализировали переменную 15

const int a(1);

}
```

Различай

Объявление переменной - выделение для нее памяти;

Инициализация – передача переменной какого-то значения (присвоили переменной определенное значение).

Задача

Задача: пользователь вводит с клавиатуры свой вес, рост, величина бицепса.

Выведите результат в следующем формате:

Данные для врача получены!

Вес: 86.8 кг;

Рост: 185 см;

Размер бицепса: 35 см.

Какие математические операции вы знаете?

Математические операции

Операции

• Арифметические:

```
+ – сложение;
- – вычитание;
* – умножение;
/ – деление.
```

<u>Результат – вещественное</u>, если хотя бы одно из чисел вещественное.

<u>Результат – целое</u>, если делимое и делитель целые.

% - остаток от деления целых чисел.

- Логические: ! (не), && (и), || (или).
- Логические поразрядные:
 - (нe), & (и), | (или), ^ (исключающее или).
- Отношения:



Задача: пользователь вводит с клавиатуры свой рост, однако в стране лилипутии он становится гигантом, поэтому лилипуты сделали новую шкалу измерения: 1 см человеческий = 3.86 лилипутский.

Выведите рост пользователя по лилипутским меркам.

Задача

Задача: пользователь вводит с клавиатуры свой год рождения, посчитайте сколько ему лет сейчас и выведите результат на экран. (занесите результат в новую переменную и выведите ее)

Особенности операций

Целочисленное деление на 10ⁿ

```
#include <iostream>
using namespace std;
int main() {
int number = 123456;
cout << number /10 <<endl; //12345</pre>
cout << number /100 <<endl; //1234</pre>
cout << number /1000 <<endl; //123</pre>
cout << number /10000 <<endl; //12</pre>
cout << number /100000 <<endl; //1</pre>
cout << number /1000000 <<endl; //0</pre>
```

Отбрасываем от числа столько цифр, сколько степень у 10 (количество нулей числа)

Особенности операций

Остаток от деления на 10ⁿ

```
#include <iostream>
using namespace std;
int main() {
int number = 123456;
cout << number % 10 <<endl; //6</pre>
cout << number % 100 <<endl; //56</pre>
cout << number % 1000 <<endl; //456</pre>
cout << number % 10000 <<endl; //3456</pre>
cout << number % 100000 <<endl; //23456</pre>
cout << number % 1000000 <<endl; //123456</pre>
```

Берем от числа столько цифр, сколько степень у 10 (количество нулей числа)

Задача

На вход подаются числа: 45 123, 1 223, 756 678, 567

Задача: вывести у каждого числа первую цифру и вторую справ (с конца) цифру. Каждая цифра на новой строке

Задача

На вход подаются числа: 45 123, 567

Задача: вывести у каждого числа все цифры. Каждая цифра на новой строке

Особенности операций

Остаток от деления на 2

```
#include <iostream>
using namespace std;
int main() {
  int number1 = 6;
  int number2 = 7;
  cout << number1 % 2 <<endl; //0
  cout << number2 % 2 <<endl; //1
}</pre>
```

Если при остатке от деления на 2 получается 1 – число не четное;

Если при остатке от деления на 2 получается 0 – число четное.

Задача

На вход подаются числа: 45 123, 567

Задача: доказать, что оба числа нечетные

Вопросы?

Повтороение

Из чего состоит программа?

С помощью какой библиотеки реализуется ввод и вывод?

Что такое переменная?

Разбор ДЗ

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 оперций
#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 оперций
#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 ecть round(1.5) == 2, round(-1.5) == -2	
floor	Округляет число вниз ("пол"), при этом floor(1.5) == 1, floor(-1.5) == -2	
ceil	Округляет число вверх ("потолок"), при этом ceil(1.5) == 2, ceil(-1.5) == -1	
fabs	Модуль (абсолютная величина)	

Корни и степени

sqrt	Квадратный корень. Использование: sqrt(x)	
cbrt	Кубический корень. Использование: cbrt(x)	
pow	Возведение в степень, возвращает ab. Использование: pow(a,b)	

Тригонометрия

sin	Использование: sin(x)
cos	Использование: cos (x)
tan	Использование: 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; //преобразоввание к вещ типу операнд
cout <<fired<<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 (х);

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 (х);

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? **HET**

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 \ll b;
a == b;
a != b;
```

Где a, b – целые числа, которые вводит пользователь

Важно:

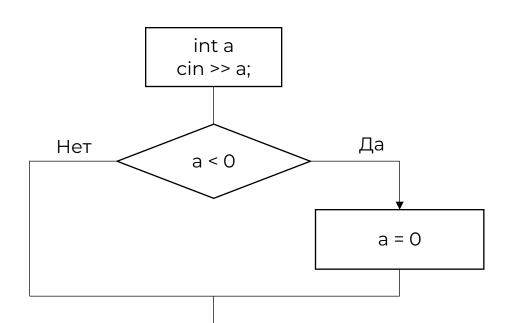
сравнение двух чисел это == (два знака равенства);

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;
}</pre>
```

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;
}</pre>
```

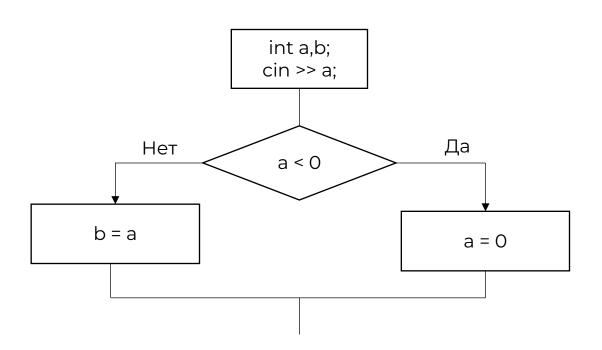
Задача

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;
}</pre>
```

Исключаем дублирование кода

Задача

Пользователь вводит в консоль число;

Задача: проверить, что введенное число четное (вывести "Да"), если нет, то вывести ("Het");

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

Так же и в программировании.

Простые высказывания объединяются в сложные путем <u>логических операторов</u>

Виды:

and or not

```
    and – оператор конъюнкции ( && );
    or – оператор дизъюнкции ( || );
    not – оператор инверсии (отрицание) (!).
```

Применение: диапазоны значений

Конъюнкция (*) Логическое **И**

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

Дизъюнкция (+) Логическое **или**

Ο	0	0
0	1	1
1	0	1
1	1	1

Инверсия Логическое **НЕ**

Ο	1
1	0

Интервалы и полуинтервалы

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

Задача

Пользователь вводит в консоль число;

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

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

Применение: хранение значения логических выражений.

Вспомним данный пример:

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

Где можно использовать булевую переменную?

Реализация через булевую 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";
}</pre>
```

Задача

Пользователь вводит в консоль число;

Задача: проверить, что введенное число входит в данные промежутки, интервалы и полуинтервалы (-4; 9) ∪{18} ∪(65; 90]; (**через булевую** перемеменную)

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