АиСД: лекция 1

Denis Bakin

Контакты

- Бакин Денис Филиппович
- tg: @dfbakin
- dfbakin_1@edu.hse.ru
- informatics.msk.ru -> АиСД(25-26) – Лицей ВШЭ

C++ кратко

- Статическая типизация
- Производительность и контроль памяти
- Совместимость с C
- Компилируемый

Первая программа

```cpp
#include <iostream>

int main() {
    std::cout << "Hello, world!\n";
}
```

Ключевые элементы:

- #include
- Точка входа: int main()
- Поток вывода: std::cout
- Перенос строки: '\n'

Переменные и типы

Переменная

- Имя + тип + (значение)
- Память выделяется на этапе компиляции (для статически известных объектов)
- Инициализацию не пропускать

```
int x = 0;
double pi = 3.14159;
bool ok = true;
char letter = 'A';

std::cout << "PI equals to " << pi << std::endl;
```

Переменные и типы

Создание переменной

```
int main() {
    int a, b, c; // объявление без инициализации
    a = 10; // оператор присваивания

    // вариант с одновременной инициализацией является предпочтительным
    int new_num_1 = 10; // создаем переменную со значением 10
    int new_num_2{10}; // альтернатива
}
```

Переменные и типы

Типы (основные)

```
// #include <string> должен быть написан выше для корректной работы с std::string


char c = '1';                  // символ
std::string s = "text";        // строка, не является встроенным типом, состоит из символов (c


bool b = true;                 // boolean, булевая, логическая переменная, принимает значения t

// целые числа
int i = 42;                    // integer, целое число (как правило, 4 байта)
short int si = 17;             // короткое целое (занимает 2 байта)
long li = 12321321312;         // длинное целое (как правило, 8 байт)

// числа с плавающей точкой
float f = 2.71828;             // дробное число с плавающей запятой (4 байта)
double d = 3.141592;           // дробное число двойной точности (8 байт)
long double ld = 1e15;         // длинное дробное (как правило, 16 байт)
```

Ввод-вывод

Потоки C++

```
#include <iostream>
#include <string>

int main() {
    std::string name;  // объявляем переменную name
    std::cout << "What is your name?\n";
    std::cin >> name;  // считываем её значение с клавиатуры
    std::cout << "Hello, " << name << "!\n";

    // объявляем нужные нам переменные
    // не инициализируем, потому что сразу будем в них читать значения
    int age, height;
    std::cout << "What is your age?\n";
    std::cin >> age;
    std::cout << "How tall are you?" << std::endl;
    std::cin >> height;

    std::cout << "Hm, it seems " << name << " is " <<
            height << "cm tall at " << age << " years old!\n";
}
```

Арифметика и приведения

Основные операции

- +, -, *, /, %
- Целочисленное деление: 7 / 3 = 2
- Остаток: 7 % 3 = 1

```
int a = 7, b = 3;
int q = a / b;    // 2
int r = a % b;    // 1
```

Арифметика и приведения

Приведения

```
// целочисленное и точное деление
int c = 6, d = 4;
c / d; // 1
static_cast<float>(c) / d; // 1.5
1. * c / d; // 1.5
```

1. `static_cast<T>(object)` приводит объект к типу `T` , если это возможно
2. `1. * c / d` — сначала выполнится умножение (`float * int = float`), затем точно деление, так как в нем участвует `float` : `float / int = float`

Арифметика и приведения

Инкремент / декремент

```
#include <iostream>

int main() {
    int a = 2;
    ++a;
    std::cout << a << '\n'; // 3
}
```

Арифметика и приведения

Сокращённые формы

```
int x = 10;
x += 5;
x -= 2;
x *= 3;
x /= 4;
x %= 3;
```

Задача: гипотенуза

Задача:
Даны два целых числа a и b. Найдите гипотенузу прямоугольного треугольника с катетами a и b.

Входные данные:
В двух строках вводятся два целых положительных числа, не превышающих 1000.

Выходные данные:
Выведите длину гипотенузы.

Задача: гипотенуза

Решение

Формула для гипотенузы:

$$c = \sqrt{a^2 + b^2}$$

Задача: гипотенуза

Пример кода на C++

```cpp
#include <iostream>
#include <cmath>

int main() {
    int a, b;
    std::cin >> a >> b;
    float hypot = std::sqrt(a * a + b * b);
    std::cout << hypot << '\n';
}
```

Уведомление

Примечание:
Для вычисления квадратного корня используется функция `std::sqrt` из библиотеки `<cmath>`.

Задача: гипотенуза

Точность вывода

```
#include <iomanip>
// ...
std::cout << std::fixed << std::setprecision(6) << c << '\n';
```

- `std::fixed` — фиксированный формат (без экспоненты): 123.456000 вместо 1.23456e+02
- `std::setprecision(6)` — 6 знаков после запятой

Задача о разрядах числа

Позиционные системы счисления

Любое число N в системе счисления с основанием d записывается как:

$$N = b_m \cdot d^{m-1} + b_{m-1} \cdot d^{m-2} + ... + b_2 \cdot d^1 + \underbrace{b_1 \cdot d^0}_{=b_1}$$

где b_i — цифры числа, $0 \leq b_i < d$.

Задача о разрядах числа

Получение цифры в разряде

Чтобы получить цифру в разряде k (отсчитывая с конца, $k = 0$ — единицы), используем:

$$\text{digit}_k = \left(\left\lfloor \frac{N}{d^k} \right\rfloor \right) \bmod d$$

Проще говоря, “убираем” деление разряды числа, а затем берем последнюю цифру – стоящую в новом разряде единиц

Задача о разрядах числа

Пример: десятичная система

Пусть $N = 12345$.

- Единицы: $12345 \bmod 10 = 5$
- Десятки: $\left\lfloor \frac{12345}{10} \right\rfloor \bmod 10 = 1234 \bmod 10 = 4$
- Сотни: $\left\lfloor \frac{12345}{100} \right\rfloor \bmod 10 = 123 \bmod 10 = 3$

Задача о разрядах числа

Пример кода

```cpp
#include <iostream>
#include <cmath>


int main() {
    int num = 12345;
    std::cout << num % 10 << '\n';        // 5
    std::cout << num / 10 % 10 << '\n';   // 4
    std::cout << num / 100 % 10 << '\n'; // 3
}
```

Ветвление

Операторы сравнения

Оператор сравнения — некоторый логический оператор, который возвращает тип `bool` – логическое значение

- `>` (больше)
- `>=` (больше или равно)
- `<` (меньше)
- `<=` (меньше или равно)
- `==` (равно)
- `!=` (не равно)

Ветвление

Синтаксис

```
if (condition1) {
    // случай, когда condition1 истинно
} else if (condition2) {
    // случай, когда condition1 ложно, а condition2 истинно
} else if (condition3) {
    // случай, когда condition1 и condition2 ложны, а condition3 истинно
} else {
    // случай, когда condition1, condition2 и condition3 ложны
}
// исполнение программы продолжится здесь вне зависимости от условия
```

Ветвление

Пример 1

```
#include <iostream>

int main() {
    int a = 5, b = 10;
    if (a < b) {
        std::cout << "a меньше b\n";
    } else {
        std::cout << "a не меньше b\n";
    }
}
```

```cpp
#include <iostream>
int main() {
    int age;
    std::cin >> age;
    if (age > 12) {
        printf("You are %d years old, so, you're too old for this cartoon\n", age);
    } else {
        printf("You are %d years old, so, you're young enough for this cartoon. Enter the nu

        int ticket_id;
        printf("Enter your ticket ID:\n", age);
        std::cin >> ticket_id;
        if (ticket_id % 6 != 0 && (ticket_id % 10) % 2 == 0) {
            printf("Enjoy the movie, customer!\n");
        } else if (ticket_id >= 0 && ticket_id < 10) {
            printf("Hi! Have a nice day at work, employee\n");
        } else if (ticket_id == 777) {
            printf("Glad to see you again, the CEO!\n");
        } else { // неверный номер билета, так как ни одно из условий не было выполнено
            printf("Invalid ticket ID! Call staff for support\n");
        }
    }
}
```

Ветвление

Сравнение вещественных чисел

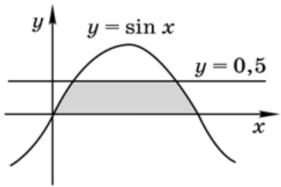
```cpp
#include <iostream>
#include <algorithm>

int main() {
    float number;
    std::cin >> number;

    // if (number == 3.0) {
        // INCORRECT
    // }

    if (std::abs(number - 3.0) < 0.000001) { // проверяем на равенство числу 3.0
        std::cout << "you have entered 3.0\n";
    }

    printf("Program finished\n");

}
```

Задача о принадлежности точки некоторой области

- область над графиком $= \{(x, y) : \quad y \geq f(x)\}$
- область под графиком $= \{(x, y) : \quad y \leq f(x)\}$
- любую область мы можем задать как объединение и пересечение некоторых областей

Задача №112166. Точка - 2

Напишите программу, которая определяет, попала ли точка с заданными координатами в заштрихованную область.



Входные данные

Входная строка содержит два вещественных числа – координаты точки на плоскости (сначала x -координата, затем – y -координата).

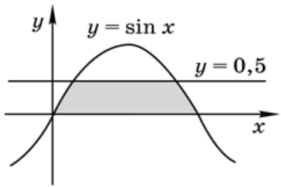
Выходные данные

Программа должна вывести слово 'YES', если точка попала в заштрихованную область, и слово 'NO', если не попала.

Figure 1: Задача 112166 на использование условий

Задача №112166. Точка - 2

Напишите программу, которая определяет, попала ли точка с заданными координатами в заштрихованную область.



Входные данные

Входная строка содержит два вещественных числа – координаты точки на плоскости (сначала x -координата, затем – y -координата).

Выходные данные

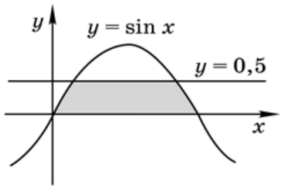
Программа должна вывести слово 'YES', если точка попала в заштрихованную область, и слово 'NO', если не попала.

Figure 2: Задача 112166 на использование условий

- ниже $y = \sin x$

Задача №112166. Точка - 2

Напишите программу, которая определяет, попала ли точка с заданными координатами в заштрихованную область.



Входные данные

Входная строка содержит два вещественных числа – координаты точки на плоскости (сначала x -координата, затем – y -координата).

Выходные данные

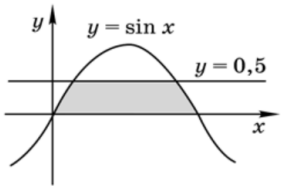
Программа должна вывести слово 'YES', если точка попала в заштрихованную область, и слово 'NO', если не попала.

Figure 2: Задача 112166 на использование условий

- ниже $y = \sin x$
- ниже $y = 0.5$

Задача №112166. Точка - 2

Напишите программу, которая определяет, попала ли точка с заданными координатами в заштрихованную область.



Входные данные

Входная строка содержит два вещественных числа – координаты точки на плоскости (сначала x -координата, затем – y -координата).

Выходные данные

Программа должна вывести слово 'YES', если точка попала в заштрихованную область, и слово 'NO', если не попала.

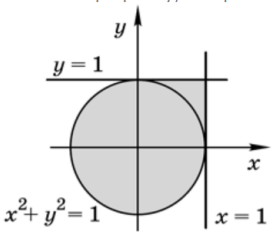
Figure 2: Задача 112166 на использование условий

- ниже $y = \sin x$
- ниже $y = 0.5$
- выше $y = 0$

```cpp
#include <iostream>
#include <cmath>

int main() {
    float x, y;
    std::cin >> x >> y;
    if (y <= std::sin(x) && y >= 0 && y <= 0.5) {
        printf("YES\n");
    } else {
        printf("NO\n");
    }

}
```

Задача №112173. Точка - 9

Напишите программу, которая определяет, попала ли точка с заданными координатами в заштрихованную область.



Входные данные

Входная строка содержит два вещественных числа – координаты точки на плоскости (сначала x -координата, затем – y -координата)

Выходные данные

Программа должна вывести слово 'YES', если точка попала в заштрихованную область, и слово 'NO', если не попала.

Figure 3: Решение задачи 112166 на использование условий

Цикл for

```cpp
#include <iostream>

int main() {
    int n;
    std::cin >> n;

    for (int i = 1; i < n; ++i) {
        printf("%d\n", i);
    }
}
```

Задача: сумма чисел

$$1 + (1 + 2) + (1 + 2 + 3) + \cdots + (1 + 2 + \cdots + n) = ?$$

Задача: сумма чисел

$1 + (1 + 2) + (1 + 2 + 3) + \cdots + (1 + 2 + \cdots + n) = ?$

```cpp
#include <iostream>
// #include <cstdint>

int main() {
    unsigned long long int n; // uint64_t n;
    std::cin >> n;
    unsigned long long int sum = 0;
    unsigned long long int last_add = 1;

    for (unsigned long long int i = 2; i < n + 2; ++i) {
        sum += last_add;
        last_add += i;
    }
    std::cout << sum << '\n';
}
```

Задача: сумма чисел

$1 + (1 + 2) + (1 + 2 + 3) + \cdots + (1 + 2 + \cdots + n) = ?$

```cpp
#include <iostream>
// #include <cstdint>

int main() {
    uint64_t n;
    std::cin >> n;
    uint64_t sum = 0;
    uint64_t last_add = 1;

    for (uint64_t i = 2; i < n + 2; ++i) {
        sum += last_add;
        last_add += i;
    }
    std::cout << sum << '\n';
}
```

Цикл while (с предусловием)

- Используется, когда заранее неизвестно количество итераций.
- Проверяет условие перед каждой итерацией.
- Можно рассматривать как цикл `for` без инициализации и шага.

Применение: разбор числа по цифрам

- Для получения i-й цифры с конца:
  $\text{digit}_i = N/d^{\,i} \,\%\, d$
- При делении нацело на d цикл продолжается, пока число не станет 0.

Пример: вывод всех цифр числа

```cpp
#include <iostream>

int main() {
    int num;
    std::cin >> num;
    while (num) {
        std::cout << num % 10 << '\n';
        num /= 10;
    }
}
```

Совет

В каком порядке она это сделает? Как нужно модифицировать программу, чтобы она выводила цифры в системе счисления d? Как в таком случае поступить с остатками, которые больше 9?

Типовые приемы

Счетчик

- Подсчет элементов, удовлетворяющих условию

```
#include <iostream>

int main() {
    int new_num;
    std::cin >> new_num;

    int cnt = 0;
    while (new_num) {
        if (new_num % 2 == 0) {
           ++cnt;
        }
        std::cin >> new_num;
    }
    std::cout << cnt << std::endl;
}
```

Типовые приемы

Поиск минимума и максимума

```
#include <iostream>
#include <algorithm>

int main() {
    int new_num;
    std::cin >> new_num;
    int max = new_num, min = new_num;

    while (new_num) {
        max = std::max(max, new_num);
        min = std::min(min, new_num);
        std::cin >> new_num;
    }

    printf("%d %d\n", min, max);
}
```