

1. Введение в C++

C++ - это язык программирования, который был разработан Бьерном Страуструпом в 1979 году на основе более раннего языка C с добавлением объектно-ориентированного программирования.

C++ является компилируемым языком. Программа на C++ после написания кода компилируется в машинный код и запускается как самостоятельное приложение.

Основные преимущества C++:

- Высокая производительность и близость к железу
- Поддержка объектно-ориентированного и структурного программирования
- Переносимость кода между разными платформами
- Обширные библиотеки готовых функций

C++ широко используется для системного программирования, разработки драйверов, компьютерных игр, GUI-приложений и многого другого.

2. Структура программы на C++

Любая программа на C++ имеет следующую базовую структуру:

- Директивы препроцессора (например, `#include` для подключения библиотек)
- Функция `main()` - точка входа в программу
- Объявление переменных и констант
- Тело программы в фигурных скобках после `main()`
- Комментарии для пояснения кода

В функции `main()` находится основной код программы. После компиляции именно эта функция начинает выполняться.

Комментарии в C++ начинаются с `//` для однострочных комментариев или заключаются между `/*` и `*/` для многострочных.

3. Ввод и вывод данных

Для работы с вводом-выводом в C++ подключается заголовочный файл `iostream`:

```
#include <iostream>
```

Далее создаются объекты `cin` и `cout` из стандартного пространства имён `std`:

```
using namespace std;
```

***cout** – используется для вывода данных на экран
cin - для ввода данных с клавиатуры.*

Вывод строки текста:

```
cout << "Hello World!";
```

Вывод значения переменной:

```
int a = 10;  
cout << a;
```

Вывод с переходом на новую строку:

```
cout << "Hello World!" << endl;
```

Ввод данных в переменную:

```
int age;  
cin >> age;
```

Таким образом организуется взаимодействие программы с пользователем через консоль.

4. Форматирование вывода

Для управления форматом выводимых данных в C++ используются специальные манипуляторы, которые применяются к потоку вывода cout:

- setw(n) - устанавливает минимальную ширину поля в n символов
- setprecision(n) - устанавливает точность для вывода вещественных чисел
- setfill(ch) - устанавливает символ заполнения пустого пространства
- hex - выводит число в шестнадцатеричном формате
- oct - выводит число в восьмеричном формате

Пример форматирования:

```
int x = 15;  
cout << setfill('#') << setw(8) << x << endl;
```

// выведет "###15###"

5. Ввод разных типов данных

Для ввода данных с клавиатуры используется объект `cin`. Он может считывать значения в переменные разных типов:

- Целые числа: `int`, `long`, `short`
- Вещественные числа: `float`, `double`
- Символьные и строковые данные: `char`, `string`
- Логические значения: `bool`

Пример ввода разных типов:

```
int num;  
cin >> num;
```

```
float f;  
cin >> f;
```

```
string s;  
getline(cin, s);
```

```
bool flag;  
cin >> boolalpha >> flag;
```

6. Обработка ошибок

При вводе-выводе могут возникать ошибки, поэтому важно проверять корректность:

- Проверка `cin.fail()` - была ли ошибка при вводе
- Использование `try/catch` для перехвата исключений
- Проверка `cin.good()` - ввод был успешен
- Метод `clear()` для сброса флагов ошибок в потоке

Пример:

```
int num;
cin >> num;

if(cin.fail()) {
    // обработка ошибочного ввода
    cin.clear(); // сброс флага ошибки
    cin >> num; // повторная попытка ввода
}
```

7. Пример программы

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

int main() {

    int age; // объявление переменной для ввода

    cout << "Введите ваш возраст: "; // вывод приглашения

    cin >> age; // ввод значения в переменную age

    // проверка на ошибку при вводе
    if(cin.fail()) {
        cout << "Ошибка при вводе данных. Попробуйте еще раз." << endl;
        cin.clear(); // сброс флага ошибки
        cin >> age; // повторный ввод
    }

    // вывод полученного значения
    cout << "Ваш возраст: " << age << " лет" << endl;

    return 0;
}
```

Пояснение кода:

1. Подключаем заголовочный файл `iostream`, который содержит функции ввода-вывода.
2. Используем пространство имен `std`, в котором находятся `cin` и `cout`.
3. Объявляем переменную `age` для хранения вводимого возраста.
4. Выводим приглашение для ввода с помощью `cout`.
5. Считываем введенное значение в переменную `age` через `cin`.
6. Проверяем, была ли ошибка при вводе с помощью метода `fail()`.
7. Если была ошибка, выводим сообщение и сбрасываем флаг ошибки методом `clear()`.
8. Делаем повторную попытку ввода в переменную `age`.
9. Выводим полученное значение `age` на экран.
10. Завершаем программу с кодом `0` (успешное завершение).

Таким образом, эта программа демонстрирует основной принцип взаимодействия с пользователем через консоль с использованием `cin` и `cout` в C++.

8. Номера задач по теме

- Задача №2936. Гипотенуза
- Задача №2937. Следующее и предыдущее
- Задача №2938. Дележ яблок - 1
- Задача №2939. Дележ яблок - 2
- Задача №2940. МКАД
- Задача №2941. Последняя цифра
- Задача №2942. Число десятков двузначного числа
- Задача №2943. Число десятков
- Задача №2944. Сумма цифр
- Задача №2945. Следующее четное
- Задача №2947. Электронные часы - 1
- Задача №2948. Электронные часы - 2
- Задача №2949. Обмен значений
- Задача №2950. Конец уроков
- Задача №2951. Стоимость покупки
- Задача №2952. Разность времен
- Задача №2953. Автопробег
- Задача №2954. Дележ яблок - 3
- Задача №2955. Улитка
- Задача №2956. Симметричное число
- Задача №2957. Проверьте делимость
- Задача №2958. Максимум

Сайт с задачами: <https://informatics.msk.ru>