

Практические занятия с применением C++

Примеры и задания могут выполняться в Online-компиляторе, поддерживающий язык программирования C++:

<https://repl.it/languages/cpp>

<http://cpp.sh/>

<http://www.pythontutor.com/visualize.html#mode=edit>

http://rextester.com/l/cpp_online_compiler_visual

https://www.onlinegdb.com/online_c++_compiler

<https://cintit.ru/compiler/>

<https://www.jdoodle.com/online-compiler-c++/>

Занятие 1. Ввод, вычисления и вывод данных

Цель: познакомиться с основными типами данных и операциями, выполняемыми над операндами различных типов.

Задание 1. Первая программа. Ввод и вывод приветствия

- Введите текст программы:

```
// Example program
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string name;
    cout << "What is your name? ";
    getline(cin, name);
    cout << "Hello, " << name << "!\n";
}
```

- Изучите объекты ввода и вывода данных:

cin – стандартный ввод – с клавиатуры,

cout – стандартный вывод на экран монитора.

- Запустите программу на выполнение. Введите имя, состоящее из нескольких слов, проверьте работу функции вывода строки `getline()`
- Замените строку `getline (cin, name);` на

`cin >> name;`

- Снова запустите программу, проверьте, что в программу передается только первое слово имени (если оно состоит из нескольких слов)

Задание 2. Вычисления

Программы считывают данные, перерабатывают их и выводят результат.

$$y = a + b$$

y , a и b – операнды:

- a и b входные данные,
- y – выходное данное (результат).

«=» и «+» - операции присваивания и сложения.

- Введите текст программы (комментарии // и /* */ можно не печатать):

```
#include <iostream>
using namespace std;
// Программа сложения двух чисел
int main()
{
    system("chcp 1251")           // для кириллицы
    int x;                        /* объявления переменных*/
    int a, b;
    cout << "Введите a и b:\n";  /* вывод приглашения */
    cin >> a;                     /* ввод с клавиатуры значения a */
    cin >> b;                     /* ввод с клавиатуры значения b */
    x = a + b;                   /* вычисление значения x */
    cout << "x = " << x << endl; /* вывод результата на экран */
    return 0;
}
```

- Запустите программу на выполнение. Введите целочисленные значения, проверьте результат операции сложения.
- Замените оператор сложения на вычитание.
- Запустите программу на выполнение. Введите целочисленные значения, проверьте результат операции вычитания.
- Замените оператор вычитания на деление.
- Запустите программу на выполнение. Введите целочисленные значения. Проверьте, что при делении целых чисел получается также целое число – дробная часть отбрасывается.

Замечание. Если вы будете использовать кириллицу, то не забывайте указывать кодовую страницу `system("chcp 1251")`.

Задание 3. Использование стандартных математических функций

Требуется вычислить площадь равностороннего треугольника по известному периметру.

- Составьте словесное описание алгоритма. Для расчета площади используйте формулу Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где p – полупериметр, a, b, c – стороны треугольника.

- Обратите внимание на необходимость применения стандартной функции вычисления квадратного корня. Опишите типы входных, промежуточных и выходных данных.
- Введите следующую программу:

```
#include <iostream>
#include <math.h> // в этом файле описана функция sqrt()

using namespace std;
int main()
{
    double p;
    cout << "Input p:\n";
    cin >> p;
    // вычисление стороны
    double st = p/3;
    // вычисление площади
    double s = sqrt(p/2*(p/2-st)*(p/2-st)*(p/2-st));
    cout << "s = " << s << endl;
    return 0;
}
```

- Запустите программу на выполнение и проверьте ее работу.

Задание 4. Применение перечислений и структур

Перечисление и структура позволяют определять новые типы, описывающие сущности предметной области.

Требуется описать новую сущность – банковский счет.

- С помощью ключевого слова **enum** создайте (объявите) новый тип – перечисление по имени **AccountType** с набором возможных значений **Checking** и **Deposit** (выделено жирным шрифтом):

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

enum AccountType { Checking, Deposit };

int main()
{
}
```

- После объявления перечисления с помощью ключевого слова **struct** создайте структуру, представляющую банковский счет по имени `BankAccount` и описывающую его свойства (поля структуры): название, период, ставку и тип счета (обратите внимание на типы переменных-полей структуры):

```
struct BankAccount
{
    string name;    // строковый тип
    int period;     // целый тип
    double price;   // вещественный тип
    AccountType acType; // тип ранее созданного перечисления
};
```

- В функции `main()` объявите переменную типа `BankAccount` по имени `goldAccount` и присвойте полям этой структурной переменной произвольные значения, соответствующие их типам:

```
int main()
{
    BankAccount goldAccount;
    goldAccount.name = "Gold";
    goldAccount.period = 8;
    goldAccount.price = 6.5;
    goldAccount.acType = Checking;

    return 0;
}
```

- Далее в этой же функции `main()` (перед инструкцией `return 0`) объявите еще одну переменную типа `BankAccount` по имени `MyAc` и тоже присвойте полям этой структурной переменной произвольные значения, соответствующие их типам, но с использованием другого варианта синтаксиса:

```
BankAccount MyAc = {"Basic", 10, 1.6, Deposit};
```

- Добавьте код функции `main()` для вывода названия счетов на экран:

```
cout << "Hello, BankAccount: ";
cout << MyAc.name << "; Period: " << MyAc.period << endl;
cout << goldAccount.name << "; Period: " << goldAccount.period << endl;
return 0;
}
```

- Запустите программу на выполнение, изучите информацию, выведенную на экран.