

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

Кафедра теоретических основ радиотехники

ОТЧЕТ

**по лабораторной работе №1
по дисциплине «Информатика»**

**ТЕМА: «Знакомство с элементами языка C++. Операторы консольного
ввода/вывода. Пустой оператор. Оператор-выражение. Составной
оператор. Арифметические операции»**

Студент гр. 2112

Детковский Д.В.

Преподаватель

Поздеев А.А.

Санкт-Петербург

2022

Цель работы.

Знакомство с интерфейсом среды программирования, составом проекта, этапами создания приложения. Освоение принципа создания консольного приложения; знакомство с операторами и основными математическими операциями; линейный алгоритм.

Краткие теоретические сведения.

<code>setlocale(LC_ALL, "Russian");</code>	Для организации ввода\вывода на русском языке
<code><тип данных> a;</code>	Объявление переменных; исходные данные
<code>cout << <составной оператор>;</code>	Вывод данных на консоль
<code>cin >> <составной оператор>;</code>	Ввод данных из консоли
<code>(условие) ? x : y</code>	Если условие истинно, то возвращается значение x, иначе значение y

Упражнение 1. Создание простейшего консольного приложения.

1. Введите код, представленный ниже (следите за тем, чтобы автоматически сгенерированные строки не дублировались!):

```
2. //подключение библиотек
#include <iostream>
#include<locale.h>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian"); //для организации ввода/вывода
на русском языке
//объявление переменных
    float a b; // исходные данные
//ввод исходных данных
    cout << Введите данные для выполнения арифметических
действий\n";
//"\n" - символ новой строки
    cout < "a = ";
    cin >> a;
    cout << "b = ";
    cin > b
//расчет и вывод результата
    t = a + b;
    cout "a + b = << t << endl;
    t = a - b;
    cout << "a - b = ";
    cout << t;
    cout "\n";
    t = a * b;
    cout << a * b = << t << endl;
    t = a / b;
    cout << "a:b = " << t << endl;
    cout << Нажмите 0 для выхода;
    cin >> a
    return 0;
}
```

Запустите отладку проекта. Если ошибок нет, программа запустится на выполнение. При наличии ошибок в наборе – устранить их и вновь выполнить п. 4.

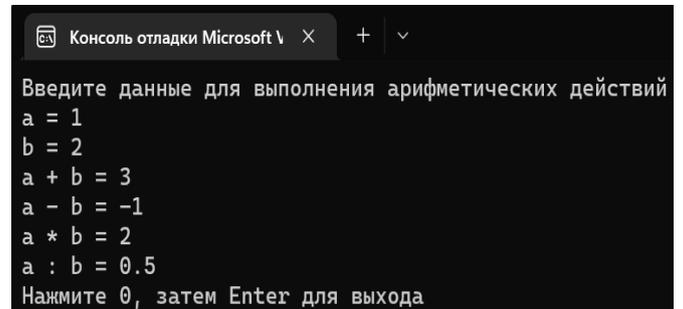
3. Пронаблюдайте выполнение программы, вводя различные исходные данные. Убедитесь в правильности полученных результатов.

4. В блоке расчета и вывода результата уберите все символы «<<< endl», запустите программу, просмотрите результат, сделайте вывод о цели этих символов. Верните строки блока вывода результата в исходное состояние.

5. В блоке ввода исходных данных в строке cout << "Введите данные для арифметических действий\n"; уберите символы <<\n>, запустите программу, просмотрите результат, сделайте вывод о цели этих символов. Верните строку cout << "Введите данные для арифметических действий\n".
6. Сделайте выводы о назначении <<endl> и <<\n>.

```
#include <iostream>
#include <locale.h>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    float a, b, t;
    cout << "Введите данные для
    выполнения арифметических действий\n";
    cout << "a = ";
    cin >> a;
    cout << "b = ";
    cin >> b;
    t = a + b;
    cout << "a + b = " << t << endl;
    t = a - b;
    cout << "a - b = " << t << endl;
    t = a * b;
    cout << "a * b = " << t << endl;
    t = a / b;
    cout << "a : b = " << t << endl;
    cout << "Нажмите 0, затем Enter для
    выхода\n";
    cin >> a;
    return 0;
}
```

Решение



```
Консоль отладки Microsoft \ x + v
Введите данные для выполнения арифметических действий
a = 1
b = 2
a + b = 3
a - b = -1
a * b = 2
a : b = 0.5
Нажмите 0, затем Enter для выхода
```

Вывод: <<endl> и <<\n> означают, что текст, который будет выводиться дальше, начнется на следующей строке. Но у них есть отличия. <<endl> используется как отдельная команда в программе. <<\n> используется в текстовой строке.

Упражнение 2. Число сотен, десятков, единиц.

1. Создайте новый консольный проект.
2. Напишите программу, определяющую число сотен, десятков и единиц в введенном пользователем трехзначном числе. В программе должны присутствовать комментарии поясняющие ее строки.

Пример:

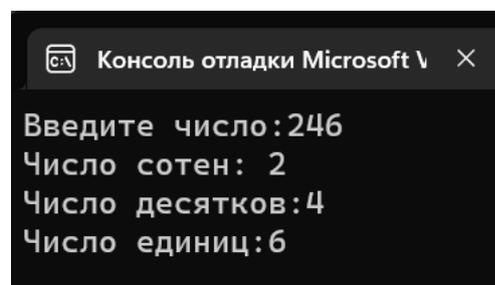
Входные данные	Выходные данные
427	Число сотен: 4 Число десятков: 2 Число единиц: 7

3. Запустите программу на исполнение, проверьте правильность работы.

```
// Подключение библиотек
#include <iostream>
#include <locale.h>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian"); // Для организации
    ввода\вывода на русском языке
    int s, d, e, ch; // Объявление переменных;
    исходные данные
    cout << "Введите число:";
    cin >> ch; // Ввод исходных данных
    s = ch / 100; // Получение числа сотен
    ch = ch % 100; // Избавляемся от сотен
    d = ch / 10; // Получение числа десятков
    e = ch % 10; // Получение числа единиц
    //Вывод данных
    cout << "Число сотен: " << s << endl;
    cout << "Число десятков:" << d << endl;
    cout << "Число единиц:" << e << endl;
    return 0;
}
```

Решение



Упражнение 3. Часы.

1. Создайте новый консольный проект.
2. Напишите программу, выводящую на экран время в формате h:mm:ss (от 0:00:00 до 23:59:59). В программе должны присутствовать комментарии поясняющие ее строки.

Пользователь вводит N секунд.

Пример:

Входные данные	Выходные данные
3602	1:00:02
129700	12:01:40

3. Запустите программу на исполнение, проверьте правильность работы.

Решение

```
// Подключение библиотек
#include <iostream>
#include <locale.h>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian"); // Для организации ввода\вывода на русском языке
    int h, m, s; // Объявление переменных; исходные данные
    cout << "Введите кол-во секунд:";
    cin >> s; // Ввод исходных данных
    h = s / 3600; // Получение числа часов
    s = s % 3600; // Избавляемся от часов
    m = s / 60; // Получение числа минут
    s = s % 60; // Получение числа секунд
    //Вывод данных условие: если минут (m) меньше 10, то выводится 0, затем m
    cout << "Время: " << h << ":" << (m < 10 ? "0" : "") << m << ":" << (s < 10 ? "0" : "") << s << endl;

    return 0;
}
```

Введите кол-во секунд:65789
Время: 18:16:29

Упражнение 4. Вычисление по формуле.

1. Создайте новый консольный проект.
2. Напишите программу, вычисляющую значение y . Значения d , x вводит пользователь.

$$y = \frac{d(|\ln x - 1|)^{(\sin x)^3}}{\sqrt{x}}$$

На экран консольного приложения должен осуществляться вывод y , его целой и дробной части. В программе должны присутствовать комментарии поясняющие ее строки.

Пример:

Входные данные	Выходные данные
6	Введите x
5	Введите d
	2.05166546

3. Запустите программу на исполнение, проверьте правильность работы.

Решение

```
// Подключение библиотек
#include <iostream>
#include <locale.h>
#include <math.h>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian"); // Для организации ввода\вывода на русском языке
    float y, d, x; // Объявление переменных; исходные данные
    cout << "Введите x:";
    cin >> x; // Ввод исходных данных
    cout << "Введите d:";
    cin >> d; // Ввод исходных данных
    // Присвоение переменной уравнение
    y = (d * pow((abs(log(x) - 1)), pow(sin(x), 3))) / (sqrt(x));
    //Вывод данных
    cout << "Ответ: " << y << endl;

    return 0;
}
```

```
Введите x:6
Введите d:5
Ответ: 2.05167
```

Контрольные вопросы.

1. Объясните термин «консольное приложение».

Консольное приложение — это программа, которая для взаимодействия с пользователем использует консоль - клавиатуру и монитор, работающий в режиме отображения символьной информации (буквы, цифры и специальные знаки).

2. Что является начальной точкой выполнения программы в консольном приложении?

Начальной точкой выполнения программы является функция `main`.

3. Поясните назначение функции `main`.

Функция `main` управляет выполнением программы, вызывая другие ее функции.

4. Для какой цели служит инструкция `return` в теле функции `main`?

Оператор `return` завершает выполнение программы в конце функции `main`, хотя по разным причинам оно может завершаться и в других местах программы.

5. Какие действия необходимо выполнить, чтобы использовать функцию, описанную в другом модуле?

Для подключения модуля используем директива `#include`.

6. Поясните цель директивы компилятора `#include`? Каков ее синтаксис? В каком месте программного кода она может находиться?

Если функция описана в другом модуле, то, для ее использования модуль должен быть подключен к текущему модулю с помощью директива `#include`. Формат директивы: `#include <ИмяФайла>`

7. Поясните цель и формат команды `cin`.

`cin` – канал связи для ввода с клавиатуры. Формат команды: `cin >> ИмяПеременной;`

8. Поясните цель и формат команды `cout`.

`cout` - канал связи для вывода на экран. Формат команды: `cout << «выводимая информация»;` или `cout << ИмяПеременной;`

9. Объясните термин «компиляция».

Компиляция – преобразование операндов «вверх», к типу большего операнда.

10. Какие части проекта подвергаются компиляции?

Компиляции подвергаются только арифметические типы данных.

11. Какие типы данных существуют, поясните их различие.

Типы данных для работы с целыми и действительными числами - для работы с целыми числами (табл. 1.1):

Таблица 1.1 Целые типы

Имя	Размер	Диапазон
<code>char</code>	1 байт	от -128 до 127
<code>short int</code>	2 байта	от -32 768 до 32 767
<code>int</code>	4 байта*	от -2 147 483 648 до 2 147 483 647

Имя	Размер	Диапазон
long int	8 байт*	от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807

- для работы с беззнаковыми числами добавляется приставка **unsigned** (табл. 1.2)

Таблица 1.2 Приставка unsigned

Имя	Размер	Диапазон
unsigned char	1 байт	от 0 до 255
unsigned short int	2 байта	0 до 65 535
unsigned int	4 байта*	от 0 до 4 294 967 295
unsigned long int	8 байт*	от 0 до 18 446 744 073 709 551 615

- для работы с действительными числами (табл. 1.3):

Таблица 1.3 Вещественные типы

Имя	Размер	Диапазон
float	4 байта	от $1.175494351 \cdot 10^{-38}$ до $3.402823466 \cdot 10^{+38}$
double	8 байт	от $2.2250738585072014 \cdot 10^{-308}$ до $1.7976931348623158 \cdot 10^{+308}$
long double	10 байт*	от $3.4 \cdot 10^{-4932}$ до $1.1 \cdot 10^{4932}$

*) зависит от типа компилятора

12. Как происходит объявление переменных?

Сначала указывается тип данных для этой переменной, а затем название этой переменной. Пример объявления переменных: `int a;` // объявление переменной а целого типа

13. Поясните назначения оператора присваивания?

При помощи оператора присваивания переменной могут присваиваться константы и выражения, значения переменных любого типа. Оператор присваивания помещает значение выражения в место, отведённое переменной.

Вывод.

Познакомились с интерфейсом среды программирования, составом проекта, этапами создания приложения. Освоили принципы создания консольного приложения; знакомство с операторами и основными математическими операциями; линейный алгоритм.