

Лабораторная работа №1. Общие сведения. Основы программирования на C++.

#FIRST. Вывод сообщения на экран.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "У каждой эпохи свой язык\n";
    return 0;
}
```

#COMMENTS. Комментарии в C++.

```
// comments.cpp
// демонстрирует использование комментариев
#include <iostream> // директива препроцессора
using namespace std; // директива using
int main() // функция с именем main
{ // начало тела функции
    cout << "У каждой эпохи свой язык\n"; // оператор
    return 0; // оператор
} // конец тела функции

/* Это
длинный
многострочный
комментарий
*/
```

#Int. Описание переменных целого типа.

```
// intvars.cpp
// работа с переменными целого типа
#include <iostream>
using namespace std;
int main()
{
    int var1; // описание переменной var1
    int var2; // описание переменной var2
    var1 = 20; // присвоение значения переменной var1
    var2 = var1 + 10; // присвоение значения переменной var2
    cout << "var1 + 10 равно "; // вывод строки
    cout << var2 << endl; // вывод значения переменной var2
    return 0;
}
```

Кроме `int` существует также еще несколько целых типов, из которых наиболее употребительными являются `long` и `short`. Переменные типов `long` и `short` имеют фиксированный размер, не зависящий от используемой системы. Размер типа `long` всегда равен 4 байтам и совпадает с размером типа `int` в случае 32-разрядных систем, подобных Windows. Это означает, что диапазон значений типа `long` совпадает с диапазоном типа `int`: от -2 147 483 648 до 2 147 483 647. Тип `long` может быть описан как `long int` между двумя такими описаниями нет разницы. Если ваша операционная система 32-разрядная, то не важно, какой тип использовать — `int` или `long`, но в 16-разрядной системе тип `long` сохранит свой диапазон значений, в то время как тип `int` уже будет иметь диапазон, совпадающий с типом `short`.

Тип short в любой операционной системе имеет размер, равный двум байтам. Диапазон значений типа short — от -32 768 до 32 767. Использование типа short не имеет особого смысла на современных 32-разрядных операционных системах, за исключением тех случаев, когда необходима экономия.

#Char. Символьные переменные. Символьные константы.

```
// charvars.cpp
// применение символьных констант
#include <iostream> // для cout и т. п.
using namespace std;
int main()
{
    char charvar1 = 'A'; // символьная переменная
                        // со значением 'A'
    char charvar2 = '\t'; // символьная переменная со значением символа табуляции
    cout << charvar1; // вывод переменной на экран
    cout << charvar2; // вывод переменной на экран
    charvar1 = 'B'; // присваивание константного
                  // значения символьной
                  // переменной
    cout << charvar1; // вывод переменной на экран
    cout << '\n'; // переход на следующую строку
    return 0;
}
```

#Fahren. Ввод с помощью cin.

```
// fahren.cpp
// применение cin и \n
#include <iostream>
using namespace std;
int main()
{
    int ftemp; // температура по Фаренгейту
    cout << "Введите температуру по Фаренгейту: ";
    cin >> ftemp;
    int ctemp = (ftemp - 32) * 5 / 9;
    cout << "Температура по Цельсию равна " << ctemp << '\n';
    return 0;
}
```

#Float. Вещественный тип данных

Тип float способен хранить числа, содержащиеся в интервале от $3.4 \cdot 10^{-38}$ до $3.4 \cdot 10^{38}$, с точностью до семи знаков после запятой. Размер типа float равен 4 байтам (32 битам).

```
// circarea.cpp
// работа с переменными вещественного типа
#include <iostream> // для cout и т.д.
using namespace std;
int main()
{
    float rad; // переменная вещественного типа
    const float PI = 3.14159F; // вещественная константа
    cout << "Введите радиус окружности: "; // запрос
    cin >> rad; // получение радиуса
    float area = PI * rad * rad; // вычисление площади круга
    cout << "Площадь круга равна " << area << endl; // вывод результата на экран
    return 0;
}
```

#Манипулятор setw

Setw - изменяет ширину поля вывода.

Пример программы без Setw.

```
// width1.cpp
// демонстрирует необходимость применения манипулятора setw
#include <iostream>
using namespace std;
int main()
{
    long pop1 = 4789426, pop2 = 274124, pop3 = 9761;
    cout << "Город " << "Нас." << endl
    << "Москва " << pop1 << endl
    << "Киров " << pop2 << endl
    << "Угрюмовка " << pop3 << endl;
    return 0;
}
```

Вывод программы будет выглядеть следующим образом:

```
Город Нас.
Москва 4789426
Киров 274124
Угрюмовка 9761
```

Пример программы с использованием Setw.

```
// width2.cpp
// применение манипулятора setw
#include <iostream>
#include <iomanip> // для использования setw
using namespace std;
int main()
{
    long pop1 = 8425785, pop2 = 47, pop3 = 9761;

    cout << setw(9) << "Город" << setw(12)
    << "Население" << endl
    << setw(9) << "Москва" << setw(12) << pop1 << endl
    << setw(9) << "Киров" << setw(12) << pop2 << endl
    << setw(9) << "Угрюмовка" << setw(12) << pop3 << endl;
    return 0;
}

    Город    Население
    Москва    8425785
    Киров      47
    Угрюмовка    9761
```

#Преобразования типов

```
// mixed.cpp
// использование смешанных выражений
#include <iostream>
using namespace std;
int main()
{
    int count = 7;
    float avgWeight = 155.5F;
    double totalWeight = count * avgWeight;
    cout << "Вес равен " << totalWeight << endl;
    return 0;
}
```

Здесь переменная типа *int* умножается на переменную типа *float*, а результат присваивается переменной типа *double*. Компиляция программы происходит без ошибок, поскольку компиляторы допускают возможность перемножения (и выполнения других арифметических операций) с операндами разных типов.

Арифметические операции

#Остаток от деления

```
// remaind.cpp
// применение операции остатка от деления
#include <iostream>
using namespace std;

int main()
{
    cout << 6 % 8 << endl // 6
         << 7 % 8 << endl // 7
         << 8 % 8 << endl // 0
         << 9 % 8 << endl // 1
         << 10 % 8 << endl; // 2

    return 0;
}
```

#Арифметические операции с присваиванием

```
// assign1.cpp
// применение арифметических операций с присваиванием
#include <iostream>
using namespace std;
int main()
{
    int ans = 27;
    ans += 10; // то же самое, что ans = ans + 10;
    cout << ans << ", ";
    ans -= 7; // то же самое, что ans = ans - 7;
    cout << ans << ", ";
    ans *= 2; // то же самое, что ans = ans * 2;
    cout << ans << ", ";
    ans /= 3; // то же самое, что ans = ans / 3;
    cout << ans << ", ";
    ans %= 3; // то же самое, что ans = ans % 3;
    cout << ans << endl;
    return 0;
}
```

Результат работы такой программы будет следующим:

37, 30, 60, 20, 2

#Инкремент

```
// increm.cpp
// применение операции инкрементирования
#include <iostream>
using namespace std;
int main()
{
    int count = 10;

    cout << "count = " << count << endl; // вывод числа 10
    cout << "count = " << ++count << endl; // вывод числа 11 (префиксная форма)
    cout << "count = " << count << endl; // вывод числа 11
    cout << "count = " << count++ << endl; // вывод числа 11 (постфиксная форма)
    cout << "count = " << count << endl; // вывод числа 12
    return 0;
}
```

Упражнения

1. Считая, что кубический фут равен 7.481 галлона, написать программу, запрашивающую у пользователя число галлонов и выводящую на экран эквивалентный объем в кубических футах.
2. Напишите программу, выводящую следующую таблицу:

1990	135
1991	7290
1992	11300
1993	16200

В программе использовать только один оператор с `cout`.

3. Напишите программу, генерирующую следующий вывод:

```
10
20
19
```

Используйте представление числа 10 в виде целой константы. Для вывода числа 20 воспользуйтесь одной из арифметических операций с присваиванием, а для вывода числа 19 — операцией декремента.

4. Напишите программу, выводящую на экран ваше любимое стихотворение. Для разбиения на строки используйте подходящую управляющую последовательность.
5. Библиотечная функция `islower()` принимает в качестве аргумента один символ (букву) и возвращает ненулевое целое значение в том случае, если буква является строчной, и нулевое, если буква является заглавной. Описание функции хранится в файле `CTYPE.H`. Напишите программу, которая принимает букву от пользователя, а затем выводит нулевое или ненулевое значение в зависимости от того, является ли буква строчной или нет.
6. На биржевых торгах за 1 фунт стерлингов давали \$1.487, за франк — \$0.172, за немецкую марку — \$0.584, а за японскую йену — \$0.00955. Напишите программу, которая запрашивает денежную сумму в долларах, а затем выводит эквивалентные суммы в других валютах.
7. Температуру, измеренную в градусах по Цельсию, можно перевести в градусы по Фаренгейту путем умножения на $9/5$ и сложения с числом 32. Напишите программу, запрашивающую температуру в градусах по Цельсию и отображающую ее эквивалент по Фаренгейту.
8. Когда размер величины, выводимой на экран с помощью манипулятора `setw()`, оказывается меньше размера зарезервированного поля, по умолчанию незаполненные поля заполняются пробелами. Манипулятор `setfill()` принимает в качестве аргумента один символ, который замещает все пробелы на незаполненных позициях поля. Модифицируйте пример `WIDTH` так, чтобы символы, разделяющие пары значений из столбцов, были не пробелами, а точками, например

Москва 8425785

9. Две дроби a/b и c/d можно сложить следующим образом: $a/b + c/d = (a*d + b*c)/(b*d)$. Например, $1/4 + 2/3 = (1*3 + 4*2)/4*3 = 11/12$

Напишите программу, запрашивающую у пользователя значения двух дробей, а затем выводящую результат, также записанный в форме дроби. Взаимодействие программы с пользователем может выглядеть, например, следующим образом:

Введите первую дробь: 1/2

Введите вторую дробь: 2/5

Сумма равна 9/10

Вы можете использовать тот факт, что операция извлечения >> может считывать более одного значения за раз: `cin >> a >> dummychar >> b;`

10. Устаревшая денежная система Великобритании состояла из фунтов, шиллингов и пенсов. 1 фунт был равен 20 шиллингам, а 1 шиллинг — 12 пенсам. Для записи использовалась система, состоящая из знака £ и трех десятичных значений, разделенных точками. Например, запись £5.2.8 обозначала 5 фунтов, 2 шиллинга и 8 пенсов (пенс — множественное число от пенни). Современная денежная система, принятая в 50-е годы XX века, состоит только из фунтов и пенсов, причем один фунт равен 100 пенсам. Такой фунт называют десятичным. Таким образом, в новой денежной системе указанная сумма будет обозначаться как £5.13 (если быть точнее, £5.1333333). Напишите программу, которая будет преобразовывать сумму, записанную в старом формате (фунты, шиллинги, пенсы), в новый формат (фунты, пенсы). Форматом взаимодействия программы с пользователем может являться следующий:

Введите количество фунтов: 7

Введите количество шиллингов: 17

Введите количество пенсов: 9

Десятичных фунтов: £7.89

В большинстве компиляторов для представления знака £ используется десятичный код 156. Некоторые компиляторы позволяют скопировать знак фунта прямо из таблицы символов Windows.

11. По умолчанию форматирование вывода производится по правому краю поля.

Можно изменить форматирование текста на левостороннее путем использования манипулятора `setiosflags(ios::left)` (не беспокойтесь о смысле новой формы записи, встретившейся в манипуляторе). Используйте этот манипулятор вместе с `setw()` для того, чтобы произвести следующий вывод:

Фамилия	Имя	Адрес	Город
Петров	Василий	Кленовая 16	Санкт-Петербург
Иванов	Сергей	Осиновая 3	Находка
Сидоров	Иван	Березовая 21	Калининград

12. Напишите программу, выполняющую действия, обратные тем, которые описаны в упражнении 10, то есть запрашивающую у пользователя сумму, указанную в десятичных фунтах, и переводящую ее в старую систему фунтов, шиллингов и пенсов. Пример взаимодействия программы с пользователем может выглядеть так:

Введите число десятичных фунтов: 3.51

Эквивалентная сумма в старой форме записи: £3.10.2

Обратите внимание на то, что если вам придется присваивать вещественное значение (например, 12.34) переменной целого типа, то его дробная часть (0.34) будет потеряна, а целая переменная получит значение 12. Чтобы избежать предупреждения со стороны компилятора, используйте явное преобразование типов. Можно использовать операторы, подобные приведенным ниже:

```
float decpounds;    // сумма в десятичных фунтах
int pounds;        // сумма в старых фунтах
float decfrac;     // десятичная дробная часть
pounds = static_cast<int>(decpounds); // отбрасывание
// дробной части
decfrac = decpounds - pounds; // прибавление дробной части
```

Чтобы получить число шиллингов, следует умножить на 20 значение переменной `decfrac`. Аналогичным образом можно получить число пенсов.

Домашнее задание

1. Написать программу вычисления площади параллелограмма. Ниже приведен рекомендуемый вид экрана во время выполнения программы (данные, введенные пользователем, выделены полужирным шрифтом).

Вычисление площади прямоугольника

Введите исходные данные:

Длина (см) -> **9**

Ширина (см) -> **7.5**

Площадь параллелограмма: 67.50 кв.см.

2. Написать программу вычисления стоимости покупки, состоящей из нескольких тетрадей и карандашей. Ниже приведен рекомендуемый вид экрана во время выполнения программы (данные, введенные пользователем, выделены полужирным шрифтом).

Вычисление стоимости покупки.

Введите исходные данные:

Цена тетради (руб.) -> **2.75**

Количество тетрадей -> **5**

Цена карандаша (руб.) -> **0.85**

Количество карандашей -> **2**

Стоимость покупки: 15.45 руб.

3. Написать программу вычисления расстояния между населенными пунктами, изображенными на карте. Ниже приведен рекомендуемый вид экрана во время выполнения программы (данные, введенные пользователем, выделены полужирным шрифтом).

Вычисление расстояния между населенными пунктами.

Введите исходные данные:

Масштаб карты (количество километров в одном сантиметре) -> **120**

Расстояние между точками, изображающими населенные пункты (см) -> **3.5**

Расстояние между населенными пунктами 420 км.

4. Написать программу пересчета величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах. Ниже приведен рекомендуемый вид экрана во время выполнения программы (данные, введенные пользователем, выделены полужирным шрифтом).

Введите временной интервал (в минутах) -> **150**

150 минут — это 2ч . 30 мин.

5. Написать программу, которая преобразует введенное с клавиатуры дробное число в денежный формат. Например, число 12.5 должно быть преобразовано к виду 12 руб. 50 коп.

Преобразование числа в денежный формат.

Введите дробное число -> **23.6**

23.6 руб. — это 23 руб. 60 коп.