**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №1**

**по дисциплине «Программирование»**

**ТИПЫ ДАННЫХ И ИХ ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ В ПАМЯТИ**

|  |  |  |
| --- | --- | --- |
| Студент гр. 4373 |  | Барсуков А. С. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2021

**Цель работы.**

Изучение основ программирования на C++; получение практических навыков работы с языком программирования, знакомство с типами данных и операциями с переменными, а также с циклами и условными конструкциями.

**Основные теоретические положения.**

Любая C++ программа представляет собой одну или несколько функций. Размер типа **int** не определяется стандартом, а зависит от компьютера и компилятора. Спецификатор **short** перед именем типа указывает компилятору, что под число требуется отвести 2 байта независимо от разрядности процессора.

Спецификатор **long** означает, что целая величина будет занимать 4 байта. Таким образом, на 16-разрядном компьютере эквиваленты **int** и **short int**, а на 32-разрядном — **int** и **long int**.

Стандарт C++ определяет три типа данных для хранения вещественных значений: **float**, **double** и **long double**. Все эти типы предназначены для представления отрицательных и положительных значений (спецификатор **unsigned**к ним не применим) в разных диапазонах.

Величины логического типа могут принимать только значения **true** и **false**, являющиеся зарезервированными словами. Внутренняя форма представления значения **false** - О (нуль). Любое другое значение интерпретируется как **true**. При преобразовании к целому типу **true** имеет значение **1**(единица).

Под величину символьного типа отводится количество байт, достаточное для размещения любого символа из набора символов для данного компьютера, что и обусловило название типа. Как правило, это 1 байт. Тип **char**, как и другие целые типы, может быть со знаком или без знака.

**Постановка задачи.**

Разработать алгоритм и написать программу, которая позволяет:

1) Вывести, сколько памяти (в байтах) на вашем компьютере отводится под различные типы данных со спецификаторами и без: int, short int, long int, float, double, long double, char и bool.

2) Вывести на экран двоичное представление в памяти (все разряды) целого числа. При выводе необходимо визуально обозначить знаковый разряд и значащие разряды отступами или цветом.

3) Вывести на экран двоичное представление в памяти (все разряды) типа float. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок.

4) Вывести на экран двоичное представление в памяти (все разряды) типа double. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок.

**Выполнение работы.**

Код программы представлен в приложении А.

Выполнение 1 задания представляет собой умение выводить количество байт, необходимое для хранения определенного типа данных в программе. Сделать это можно с помощью операции sizeof().

Выполнение 2 задания подразумевает умение пользоваться циклами, условными конструкциями и побитовыми операциями. Чтобы получить нужный результат, мы инициализируем переменную и маску, которая будет равна 1 << 31(в двоичном представлении это эквивалентно 1 на последнем разряде и далее 31 нулю). В самом цикле мы выполняем побитовое И, и если на месте 1 в маске будет единица в обрабатываемом числе, то выведется 1, иначе 0. После выполнения итерации мы сдвигаем маску на один бит вправо и повторяем те же махинации. В итоге мы получаем “слепок” числа.

Можно сказать, что выполнение 3 задания идентично со 2, но есть один небольшой ньюанс, который нуждается в освещении. В памяти компьютера очень непросто получить двочиное представление вещественных типов данных, поэтому мы прибегаем к использованию union, который, по сути своей, объединяет ячейки памяти для целочисленного и вещественного числа, и благодаря этому получаем число, которое в двоичном виде эквивалентно числу типаfloat.

Для выполнения 4 задания мы также используем union, но объединяем double с массивом int Arr[2], так как для хранения double отводится 8 байт, а на тип int 4 байта.

**Выводы.**

Получили практические навыки при работе с типами данных, научились получать представление разных типов данных в памяти компьютера, использовав при этом условные конструкции, циклы и побитовые операции.

Приложение А

рабочий код



