**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №1**

**по дисциплине «Программирование»**

## **Тема:** ТИПЫ ДАННЫХ И ИХ ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ В ПАМЯТИ

|  |  |  |
| --- | --- | --- |
| Студент гр. 2373 |  | Гермаш А. В. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2022

**Цель работы.**

Изучение числовых типов данных языка c++; изучение представления типов данных в памяти; перевод чисел в двоичный код.

**Основные теоретические положения.**

Тип данных — фундаментальное понятие теории программирования. Тип данных определяет множество значений, набор операций, которые можно применять к таким значениям и, возможно, способ реализации хранения значений и выполнения операций. Любые данные, которыми оперируют программы, относятся к определённым типам.

В языке С++ основными типами данных являются int, float, double, bool, char. Также присутствуют различные диапазоны, такие как short, long, long long, signed, unsigned и др.

Для определения размера типа данных следует использовать оператор sizeof().

Переменная - поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным и изменять значение в ходе выполнения программы.

Переменные в языке C++ чаще всего представлены в виде:

[класс памяти] <тип данных> <идентификатор> [инициализатор]; , например: short int a = 3;

Числовые переменные в c++ можно представить в виде двоичного кода.

У signed типов данных первый бит отводится под знак.

Для int – 1 знаковый бит, 31 под двоичное представление числа.

Для float – 1 знаковый бит, 8 под экспоненту и 23 под мантиссу.

Для double – 1 знаковый бит, 11 под экспоненту и 52 под мантиссу.

Мантисса — это, по сути, число, записанное без точки. Экспонента — это степень, в которую нужно возвести некое число N (как правило, N = 2), чтобы при перемножении на мантиссу получить искомое число.

x = m \* N ^ e, где m и e — целые числа, записанные в бинарном виде в выделенных под них битах.

**Постановка задачи**

Необходимо создать программу, позволяющую:

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.

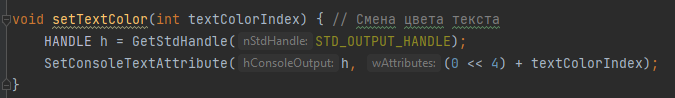
2) Вывести на экран двоичное представление в памяти (все разряды) целого числа. При выводе необходимо визуально обозначить знаковый разряд и значащие разряды отступами или цветом.

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.

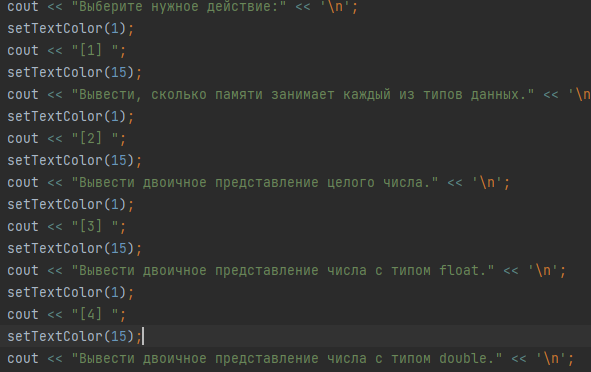
4) Вывести на экран двоичное представление в памяти (все разряды) типа double. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок.

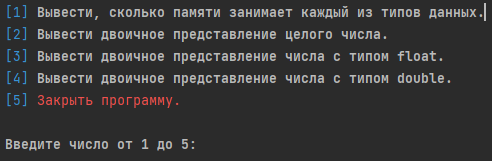
**Выполнение работы.**

1. Создание функции setTextColor(#цвета) для изменения цвета вывода текста в консоли

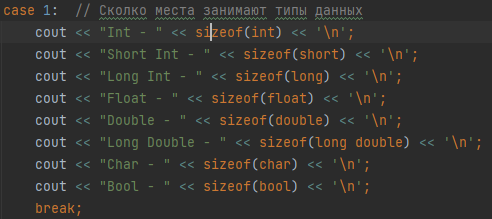


1. Создание меню программы

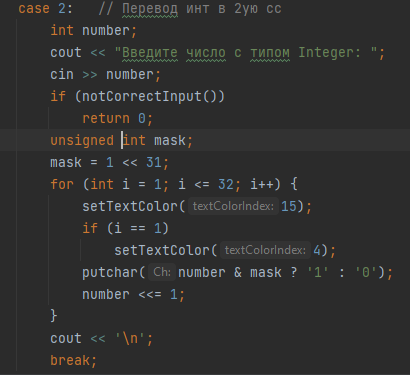


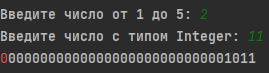


1. Создание блока switch для всех случаев введенных значений.
2. Создание “case 1”: вывод всех значений занимаемой памяти для заданных типов данных

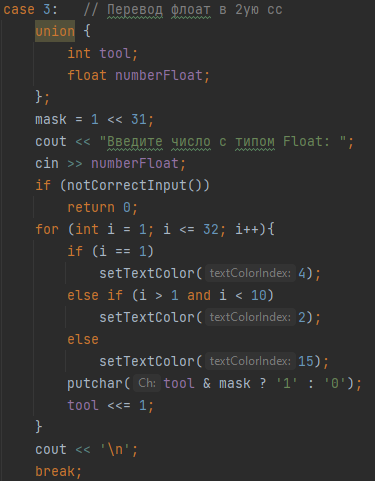


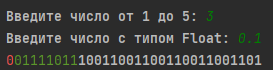
1. Создание “case 2”, написание алгоритма для перевода целого числа в двоичное представление в памяти:



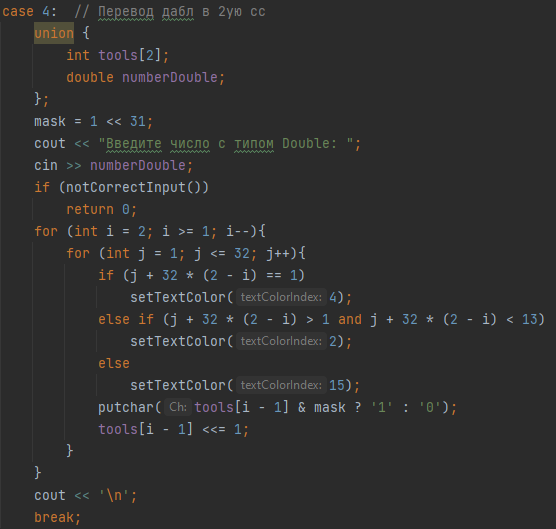


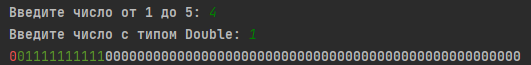
1. Создание “case 3”, написание алгоритма для перевода float в двоичную систему счисления:





1. Создание “case 4”, написание алгоритма для перевода double в двоичную систему счисления:

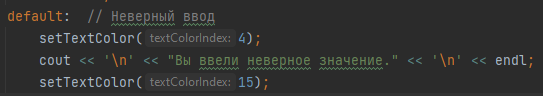


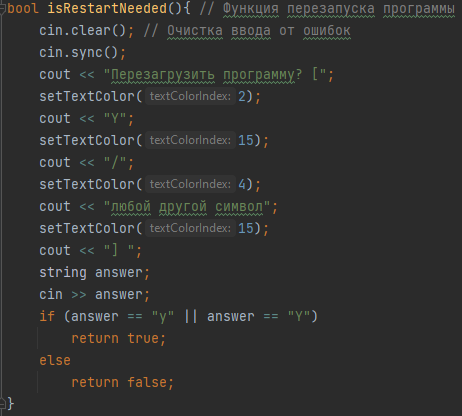


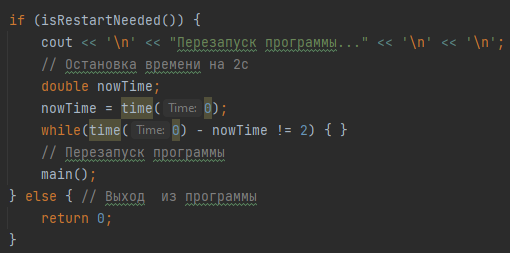
1. Создание “case 5”, условия для выхода из программы

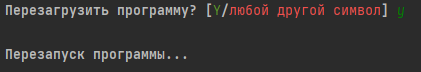


1. Создание “default”, алгоритма для предотвращения ввода неверного числа

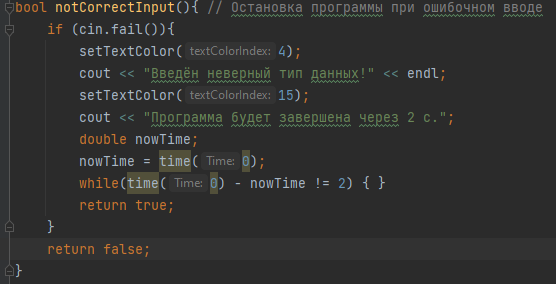


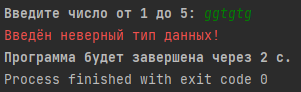
1. Создание функции для возможности перезапуска программы и условия для рекурсивного алгоритма





1. Создание алгоритма для отсеивания ошибочного ввода:





1. Исправление предупреждений, улучшение качества кода, добавление комментариев.

**Выводы.**

Мы познакомились с различными типами данных в C++, узнали их характеристики, написали алгоритмы для перевода в их двоичное представление в памяти.