

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. 1323

Князев И. А.

Преподаватель

Глущенко А. Г.

Санкт-Петербург

2022

Цель работы.

Изучение типов данных и их внутреннего представления в памяти; получение практических навыков работы с типами данных; определить, каким образом типы данных представляются на компьютере.

Основные теоретические положения.

Программе необходимо точно представлять какие данные хранятся в этом байте памяти.

Для разрешения подобных коллизий в языках программирования введено **понятие типов данных**.

Тип данных для каждого программного объекта, представляющего данные, определяет:

- характер данных (число, со знаком или без знака, целое или с дробной частью, одиночный символ или текст, представляющий последовательность символов и т.д.);
- объем памяти, который занимают в памяти эти данные;
- диапазон или множество возможных значений;
- правила обработки этих данных (например, допустимые операции)

В разных языках программирования определены разные наборы типов данных, но, в целом, типы данных можно разделить на две группы: простые и структурированные типы. Простые типы данных представляют неразделимые данные, не имеющие внутренней структуры (это, например, числа, символы и т.д.). Структурированные типы данных, как это вытекает из их названия, имеют внутреннюю структуру (иногда достаточно сложную). Структурированные типы строятся на основе простых типов данных.

Другой уровень классификации разделяет все типы данных на predetermined (изначально встроенные в язык программирования) и пользовательские (типы данных, определяемые программистом) типы данных.

Основные (предопределенные) типы данных часто называют арифметическими, поскольку их можно использовать в арифметических операциях.

Типы **int**, **bool** и **char** относят к группе целочисленных (целых) типов, а **float** и **double** - к группе вещественных типов - типов с плавающей точкой. Код, который формирует компилятор для обработки целых величин, отличается от кода для величин с плавающей точкой.

Существует четыре спецификатора типа, уточняющих внутреннее представление и диапазон значений стандартных типов: **short** (короткий); **long** (длинный); **signed** (знаковый); **unsigned** (без знаковый).

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.

1. На экран выводится, сколько памяти (в байтах и битах) на компьютере отводится под различные типы данных со спецификаторами и без в этом порядке:

'int'

'short int'

'long int'

'long long int'

'float'

'double'

'long double'

'char'

'bool'

2. Программа просит ввести символ 'Y' или 'N' и считывает его из потока ввода. В зависимости от введённого символа значения для последующих заданий будут либо генерироваться случайно, либо вводятся пользователем. Дальнейшие пункты подразумевают, что значения вводятся вручную.
3. Программа просит ввести целое число и считывает его из потока ввода.
4. Программа рассчитывает двоичную строку для положительного и отрицательного значения введённого целого числа типа 'int' с помощью битовых сдвигов и бинарной маски, и выводит полученную строку в поток вывода.

5. Программа просит ввести вещественное число и считывает его из потока ввода.
6. Программа рассчитывает двоичную строку для положительного и отрицательного значения введённого вещественного числа типа 'float' с помощью объединения, битовых сдвигов и бинарной маски, и выводит полученную строку в поток вывода.
7. Программа просит ввести вещественное число и считывает его из потока ввода.
8. Программа рассчитывает двоичную строку для положительного и отрицательного значения введённого вещественного числа типа 'double' с помощью объединения, битовых сдвигов и бинарной маски, и выводит полученную строку в поток вывода.

Результаты работы программы.

```
PS C:\Users\acc13\Documents\cpp\lab\01> .\main.exe  
Sizes of primitive types:  
'int'           4 bytes / 32 bits  
'short int'     2 bytes / 16 bits  
'long int'      4 bytes / 32 bits  
'long long int' 8 bytes / 64 bits  
'float'         4 bytes / 32 bits  
'double'        8 bytes / 64 bits  
'long double'   16 bytes / 128 bits  
'char'          1 bytes / 8 bits  
'bool'          1 bytes / 8 bits  
  
Generate random numbers? (Y/N): N  
Binary string of 'signed int' (sign[1]|numeric[31]):  
Enter an integer: 12345  
12345       -> 0|0000000000000000000011000000111001  
-12345      -> 1|1111111111111111111100111111000111  
Binary string of 'float' (sign[1]|exponent[8]|mantissa[23]):  
Enter a float: 3.1415  
3.1415      -> 0|10000000|10010010000111001010110  
-3.1415     -> 1|10000000|10010010000111001010110  
Binary string of 'double' (sign[1]|exponent[11]|mantissa[52]):  
Enter a double: 9.93123402376412  
9.93123     -> 0|10000000010|001111011100110010101101001011100111111101010010000  
-9.93123    -> 1|10000000010|001111011100110010101101001011100111111101010010000  
Press any key to continue ...  
PS C:\Users\acc13\Documents\cpp\lab\01>
```

Рисунок 1. Результат программы.

Вывод.

Во время выполнения данной работы были изучены типы данных и их внутреннее представление в памяти; было определено, каким образом типы данных представляются на компьютере. Полученная программа способна выводить на экран представление в памяти чисел различного (вещественного и целого) типа.