**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, определяемые пользователем. Структуры

|  |  |  |
| --- | --- | --- |
| Студентка гр. 2373 | Чекстер А. А. |  |
| Преподаватель | Глущенко А. Г. |  |

Санкт-Петербург

2022

**Цель работы.**

Изучение строения и хранение в памяти компьютера различных типов данных; использование циклов, условий на языке С++; закрепление теоретического материала о структуре данных.

**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 предопределенные (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).

Sizeof - это унарный оператор в языках программирования C и C++. Он генерирует размер хранилища выражения или типа данных, измеряемый в количестве единиц измерения размера символов.

#include <iostream>

using namespace std;

int main() {

cout << sizeof (int);

}

Данная программа выведет размер, который занимает тип данных int (в байтах).

Объединение – это группирование переменных, которые разделяют одну и ту же область памяти. В зависимости от интерпретации осуществляется обращение к той или другой переменной объединения. Все переменные, что включены в объединение начинаются с одной границы.

union *имя\_типа\_объединения*

{

тип *переменная1*;

тип *переменная2*;

...

тип *переменнаяN*;

};

**Постановка задачи.**

Необходимо 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.

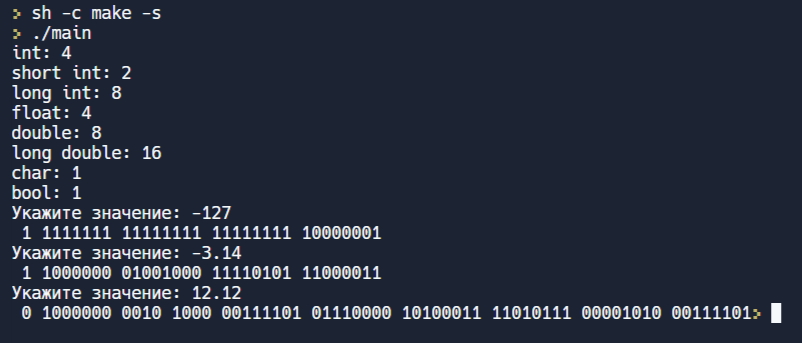
Блок описания кода и использованных алгоритмов

1. Вывод размера различных типов переменных, таких как:

* int
* short int
* long int
* float
* double
* long double
* char
* bool.

1. Задается переменная int(float, double) для последующих заданий
2. Значение переменной задается с клавиатуры
3. Задается кол-во разрядов, а также маска
4. С помощь цикла значение с клавиатуры переводиться в представление в памяти
5. Побитовый сдвиг числа

Блок скриншотов работы программы



Блок таблицы с тестовыми данными

|  |  |
| --- | --- |
| Задание №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. | При выполнении программы на консоль выводяться названия и размер типов данных. |
| Задание №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. | На кансоли выводиться запрос произвольного значения:  После указания значениия на консоль выводиться двоичное представление в памяти (все разряды) целого числа: |
| Задание №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. | На кансоли выводиться запрос произвольного значения:  После указания значениия на консоль выводиться двоичное представление в памяти (все разряды) целого числа: |
| Задание №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. | На кансоли выводиться запрос произвольного:  двоичное представление в памяти (все разряды) типа double: |
|  |  |

**Выводы.**

В данной практической работе используются методы программирования на языке С++, изучаются типы данных и их внутреннее представления в памяти; получение практических навыков работы с типами данных; определили, каким образом типы данных представляются на компьютере.

Приложение А

рабочий код

#include <iostream>

using namespace std;

int main() {

setlocale(0,"");

/\* Задание 1 \*/

cout << "int: " << sizeof(int) << "\n";

cout << "short int: " << sizeof(short) << "\n";

cout << "long int: " << sizeof(long) << "\n";

cout << "float: " << sizeof(float) << "\n";

cout << "double: " << sizeof(double) << "\n";

cout << "long double: " << sizeof(long double) << "\n";

cout << "char: " << sizeof(char) << "\n";

cout << "bool: " << sizeof(bool) << "\n";

/\* Задание 2 \*/

cout << "Укажите значение: ";

int number;

cin >> number;

int digit = 32;

int mask = 1 << digit - 1;

for (int i = 0; i < digit; i++) {

if (i == 1 || i % 8 == 0)

cout << " ";

cout << (number & mask ? '1' : '0');

number <<= 1;

}

/\* Задание 3 \*/

cout << "\n";

cout << "Укажите значение: ";

union {

int numb;

float number\_f;

};

cin >> number\_f;

int digit\_f = 32;

int mask\_f = 1 << digit\_f - 1;

for (int i = 0; i < digit\_f; i++) {

if (i == 1 || i % 8 == 0)

cout << " ";

cout << (numb & mask\_f ? '1' : '0');

numb <<= 1;

}

/\* Задание 4 \*/

cout << "\n";

cout << "Укажите значение: ";

union {

int num[3];

double number\_d;

};

cin >> number\_d;

int digit\_d = 32;

int mask\_d = 1 << digit\_d - 1;

for (int i = 0; i < digit\_d; i++) {

if (i == 1 || i % 8 == 0 || i == 12)

cout << " ";

cout << (num[1] & mask\_d ? '1' : '0');

num[1] <<= 1;

}

for (int i = 0; i < digit\_d; i++) {

if (i % 8 == 0)

cout << " ";

cout << (num[0] & mask\_d ? '1' : '0');

num[0] <<= 1;

}

}