**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. 3373 |  | Бугров С. В. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2023

**Цель работы.**

Познакомиться с основными типами данных в языке C++. На практике изучить их внутреннее представление в памяти.

**Основные теоретические положения.**

Целочисленные. Виды данных в программировании делятся на знаковые и беззнаковые. Уже понятно из наименования: в знаковых могут храниться все действительные числа, а также ноль, а в беззнаковых – только положительные (больше нуля). У беззнаковых данных диапазон больше в 2 раза, чем у знаковых. Это – из-за компьютерного восприятия: в знаковых типах бит отражает знак числа, где 0 является положительным значением, а 1 – отрицательным. Тип short (короткий целый.) Для него в памяти отведено 16 бит, то есть 2 байта (216 = 65 536). Диапазон значений, который может принять тип short со знаком – это [-32 768; 32 767]. Переменный тип long (длинный целый). Этому типу выделено 64 бита, то есть 8 байт. (264 = 1,8 446 744 \* 1 019). Он имеет внушительный диапазон: в случае знакового типа это [-9 223 372 036 854 775 808 9 223 372 036 854 775 807]. Также модификатор long может использоваться в связке с другими типами (long будет указан перед наименованием типа, допустим, long double). Благодаря этому увеличивается диапазон возможных значений.

Вещественные. Значения этого типа имеют плавающую запятую. Плавающая запятая — форма представления действительных чисел, в которой число хранится в форме мантиссы и показателя степени. Если говорить на языке программирования, то каждое число может быть представлено в следующей форме: N = M ∗ 10p, где N - записываемое число; M - мантисса; p (целое) — порядок. float - с плавающей запятой; double - с плавающей запятой двойной точности.

Символьный тип данных в программировании. В символьном типе переменная имеет только один символ, целое число. В соответствии с кодировкой, он преобразуется в некий символ. Символьному виду данных в программировании присущ только размер выделяемой под них памяти.

Логический тип данных в программировании. У этого типа данных могут быть следующие значения: false (ложь) или true (правда).

Массивы. Массив – это последовательно выстроенная и имеющая общее имя структура данных, в которой хранятся элементы одного типа. Его можно представить как набор пронумерованных ячеек, в каждую из которых поместили какие-то данные (один элемент данных в конкретную ячейку). Индексом массива является целое число, ссылающее на определенную часть массива. Индекс, как правило, имеет вид int.

**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.**

Сама программа состоит из главной функции main:

#include <iostream>

using namespace std;

int main()

{

unsigned int order = sizeof(int)\*8;

int choice;

int choice2;

unsigned int mask = 1 << order - 1;

int a;

float b;

double c;

setlocale(0, "");

while (true)

{

cout << "Что будем делать?" << "\n";

cout << "1. Размеры типов данных" << "\n";

cout << "2. Двоичное представление целого числа" << "\n";

cout << "3. Двоичное представление типа float" << "\n";

cout << "4. Двоичное представление типа double" << "\n";

cin >> choice;

switch (choice)

{

case 1:

system("cls");

cout << "Размер типа int: " << sizeof(int) << " байта" << "\n";

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

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

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

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

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

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

cout << endl << endl;

cout << "Что-нибудь ещё ?" << "\n";

cout << "1. Продолжить" << "\n";

cout << "2. Выйти" << "\n";

cin >> choice2;

switch (choice2)

{

case 1:

system("cls");

break;

case 2:

return 0;

break;

default:

system("cls");

break;

}

break;

case 2:

system("cls");

cout << "Введите целое число: ";

cin >> a;

for (int i = 0; i < 32; i++) {

cout << ((a & mask) ? 1 : 0);

a = a << 1;

if (i == 0 or i == 7 or i == 15 or i == 23)

cout << " ";

}

cout << endl << endl;

cout << "Что-нибудь ещё ?" << "\n";

cout << "1. Продолжить" << "\n";

cout << "2. Выйти" << "\n";

cin >> choice2;

switch (choice2)

{

case 1:

system("cls");

break;

case 2:

return 0;

break;

default:

system("cls");

break;

}

break;

case 3:

system("cls");

union {

int uFloat;

float b;

};

cout << "Введите число типа float: ";

cin >> b;

for (int i = 0; i < 32; i++) {

cout << ((uFloat & mask) ? 1 : 0);

uFloat = uFloat << 1;

if (i == 0 or i == 1 or i == 8)

cout << " ";

}

cout << endl << endl;

cout << "Что-нибудь ещё ?" << "\n";

cout << "1. Продолжить" << "\n";

cout << "2. Выйти" << "\n";

cin >> choice2;

switch (choice2)

{

case 1:

system("cls");

break;

case 2:

return 0;

break;

default:

system("cls");

break;

}

break;

case 4:

system("cls");

union {

int uDouble[2];

double c;

};

cout << "Введите число типа double: ";

cin >> c;

for (int i = 0; i < 64; i++)

{

if (i - 32 < 0)

{

cout << ((uDouble[1] & mask) ? 1 : 0);

uDouble[1] = uDouble[1] << 1;

if (i == 0 or i == 1 or i == 11)

cout << " ";

}

else

{

cout << ((uDouble[0] & mask) ? 1 : 0);

uDouble[0] = uDouble[0] << 1;

}

}

cout << endl << endl;

cout << "Что-нибудь ещё ?" << "\n";

cout << "1. Продолжить" << "\n";

cout << "2. Выйти" << "\n";

cin >> choice2;

switch (choice2)

{

case 1:

system("cls");

break;

case 2:

return 0;

break;

default:

system("cls");

break;

}

break;

default:

system("cls");

cout << "Введите число от 1 до 4!!!" << "\n" << "\n";

}

}

return 1;

}

Данная процедура выводит список выполняемых пунктов и предлагает пользователю выбрать один из них.

При выборе 1-го пункта программа переходит в case 1. В нем с помощью функции sizeof() выводятся размеры всех типов данных.

При выборе 2-го пункта программа переходит в case 2. Далее программа запрашивает у пользователя число и с помощью побитового сравнения выводит битовое представление данного числа в компьютере. Также для читаемости все разряды разделяются пробелами: отдельно бит знака и далее оставшиеся биты по 8.

При выборе 3-го пункта программа переходит в case 3. Для начала программа с помощью директивы union выделяет память под хранение двух переменных: вещественного и целого типа. Далее программа запрашивает у пользователя вещественное число и с помощью побитового сравнения выводит битовое представление данного числа в компьютере. Также для читаемости все разряды разделяются пробелами: отдельно бит знака мантиссы, бит знака порядка, а также отдельно биты порядка и биты мантиссы.

При выборе 4-го пункта программа переходит в case 4. Для начала программа с помощью директивы union выделяет память под хранение двух переменных: вещественного числа и массива из двух элементов. Далее программа запрашивает у пользователя вещественное число и с помощью побитового сравнения каждого из элементов выводит битовое представление данного числа в компьютере. Также для читаемости все разряды разделяются пробелами: отдельно бит знака мантиссы, бит знака порядка, а также отдельно биты порядка и биты мантиссы.

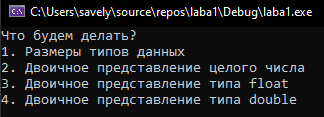
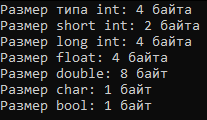
  
Рисунок 1 – меню  
  
Рисунок 2 – case 1, вывод размеров всех типов данных  
  
Рисунок 3 – case 2, двоичное представление целого числа

  
Рисунок 4 – case 3, двоичное представление числа типа float

  
Рисунок 5 – case 4, двоичное представление числа типа double

**Выводы**

По результатам выполнения данной лабораторной работы:

1) Я узнал, какую размерность имеют переменные в моей среде программирования.

2) Я разобрался, как представляются целые и вещественные числа в моей среде программирования

3) Я выяснил, что отрицательные целые числа в моей среде программирования задаются в дополнительном коде, а порядок в вещественных числах – смещенный.

Приложение А

рабочий код

#include <iostream>

using namespace std;

int main()

{

unsigned int order = sizeof(int)\*8;

int choice;

int choice2;

unsigned int mask = 1 << order - 1;

int a;

float b;

double c;

setlocale(0, "");

while (true)

{

cout << "Что будем делать?" << "\n";

cout << "1. Размеры типов данных" << "\n";

cout << "2. Двоичное представление целого числа" << "\n";

cout << "3. Двоичное представление типа float" << "\n";

cout << "4. Двоичное представление типа double" << "\n";

cin >> choice;

switch (choice)

{

case 1:

system("cls");

cout << "Размер типа int: " << sizeof(int) << " байта" << "\n";

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

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

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

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

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

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

cout << endl << endl;

cout << "Что-нибудь ещё ?" << "\n";

cout << "1. Продолжить" << "\n";

cout << "2. Выйти" << "\n";

cin >> choice2;

switch (choice2)

{

case 1:

system("cls");

break;

case 2:

return 0;

break;

default:

system("cls");

break;

}

break;

case 2:

system("cls");

cout << "Введите целое число: ";

cin >> a;

for (int i = 0; i < 32; i++) {

cout << ((a & mask) ? 1 : 0);

a = a << 1;

if (i == 0 or i == 7 or i == 15 or i == 23)

cout << " ";

}

cout << endl << endl;

cout << "Что-нибудь ещё ?" << "\n";

cout << "1. Продолжить" << "\n";

cout << "2. Выйти" << "\n";

cin >> choice2;

switch (choice2)

{

case 1:

system("cls");

break;

case 2:

return 0;

break;

default:

system("cls");

break;

}

break;

case 3:

system("cls");

union {

int uFloat;

float b;

};

cout << "Введите число типа float: ";

cin >> b;

for (int i = 0; i < 32; i++) {

cout << ((uFloat & mask) ? 1 : 0);

uFloat = uFloat << 1;

if (i == 0 or i == 1 or i == 8)

cout << " ";

}

cout << endl << endl;

cout << "Что-нибудь ещё ?" << "\n";

cout << "1. Продолжить" << "\n";

cout << "2. Выйти" << "\n";

cin >> choice2;

switch (choice2)

{

case 1:

system("cls");

break;

case 2:

return 0;

break;

default:

system("cls");

break;

}

break;

case 4:

system("cls");

union {

int uDouble[2];

double c;

};

cout << "Введите число типа double: ";

cin >> c;

for (int i = 0; i < 64; i++)

{

if (i - 32 < 0)

{

cout << ((uDouble[1] & mask) ? 1 : 0);

uDouble[1] = uDouble[1] << 1;

if (i == 0 or i == 1 or i == 11)

cout << " ";

}

else

{

cout << ((uDouble[0] & mask) ? 1 : 0);

uDouble[0] = uDouble[0] << 1;

}

}

cout << endl << endl;

cout << "Что-нибудь ещё ?" << "\n";

cout << "1. Продолжить" << "\n";

cout << "2. Выйти" << "\n";

cin >> choice2;

switch (choice2)

{

case 1:

system("cls");

break;

case 2:

return 0;

break;

default:

system("cls");

break;

}

break;

default:

system("cls");

cout << "Введите число от 1 до 4!!!" << "\n" << "\n";

}

}

return 1;

}