**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 804 «Теория вероятностей и компьютерное моделирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Ермакова Анна

Группа: 80-201

Преподаватель: Чернышов Л.Н.

Дата: 24.09.2020

Оценка:

Москва, 2020

1. Постановка задачи (Вариант №13):

**Создать класс Long** для работы с целыми беззнаковыми числами из 64 бит. Число должно быть представлено двумя полями unsigned int. Должны быть реализованы арифметические операции, присутствующие в С++, и сравнения.

2. Описание программы:

Программа состоит из

1. main.cpp - реализовано меню, с помощью switch, в котором представлены вывод текущего числа, как элемента класса Long,  вывод текущего элемента, как числа в 10-чной системе счисления, ввод нового текущего числа, унарные и бинарные арифметические операции и сравнения, выход из меню. Число принимается в виде unsigned long int и делится на два Long побитовым сдвигом.
2. Long.cpp - реализованы методы класса :конструктор, присваивание (Long= Int,Int и Long = Long), инкремент, декремент, сложение, вычитание, умножение, деление, остаток от деления, проверка на равенство двух чисел, вывод большего из двух чисел, меньшего.
3. Long.hpp - описан класс Long (представленный двумя полями unsigned int), в котором объявлены его атрибуты (unsigned int high; unsigned int low;)и методы( Long(); void PrintINT(); void Print(); Long EqualInt(int high, int low); Long Eq(Long b); Long Increment(); Long Decrement(); Long Add(Long a); Long Subtract(Long a); Long Multiply(Long a); Long Divide(Long a); Long Mod(Long a); bool EqualLong(Long b); Long More(Long a); Long Less(Long a);).

\*программа работает как калькулятор ,т.е. есть текущее число, с которым производятся необходимые преобразования, по умолчанию число равно нулю, чтобы ввести новое, необходимо в меню выбрать поле 3-Enter the number.

3. Набор тестов

Файлы Long.cpp, Long.hpp, main.cpp, CMakeLists.txt на компьютере расположены в папке oop\_lr1. Long.cpp - статическая библиотека в которой реализованы методы класса, описанного в Long.hpp. В main.cpp реализовано меню с выбором действия над числом.

Листинг CMakeLists.txt :

cmake\_minimum\_required(VERSION 2.8)

project(OOP1)

set(SOURCE\_EXE main.cpp)

set(SOURCE\_LIB Long.cpp)

add\_library(Long STATIC ${SOURCE\_LIB})

add\_executable(oop\_exercise\_01 ${SOURCE\_EXE})

target\_link\_libraries(oop\_exercise\_01 Long)

На вход

- позиция меню (int) в соответствии с выбранной операцией

- число№1 (unsigned long int) - как в калькуляторе, это число над которым будут проводится последующие операции

- при использовании бинарных операций появится объявление ввести число№2 (unsigned long int)

На выход

- всегда результат операции, кроме проверки на равенство (будет выведен ответ да/нет)

Для управления программой необходимо вводить числа: 1/2 для вывода текущего значения, 3 для ввода нового, 4,5 для унарных операций, 6-10 для бинарных операций (с последующим вводом второго операнда), 11 для проверки на равенство, 12,13 для определения большего/меньшего числа пары, 14 для выхода.

Последнее число в выходных данных - всегда текущее число.

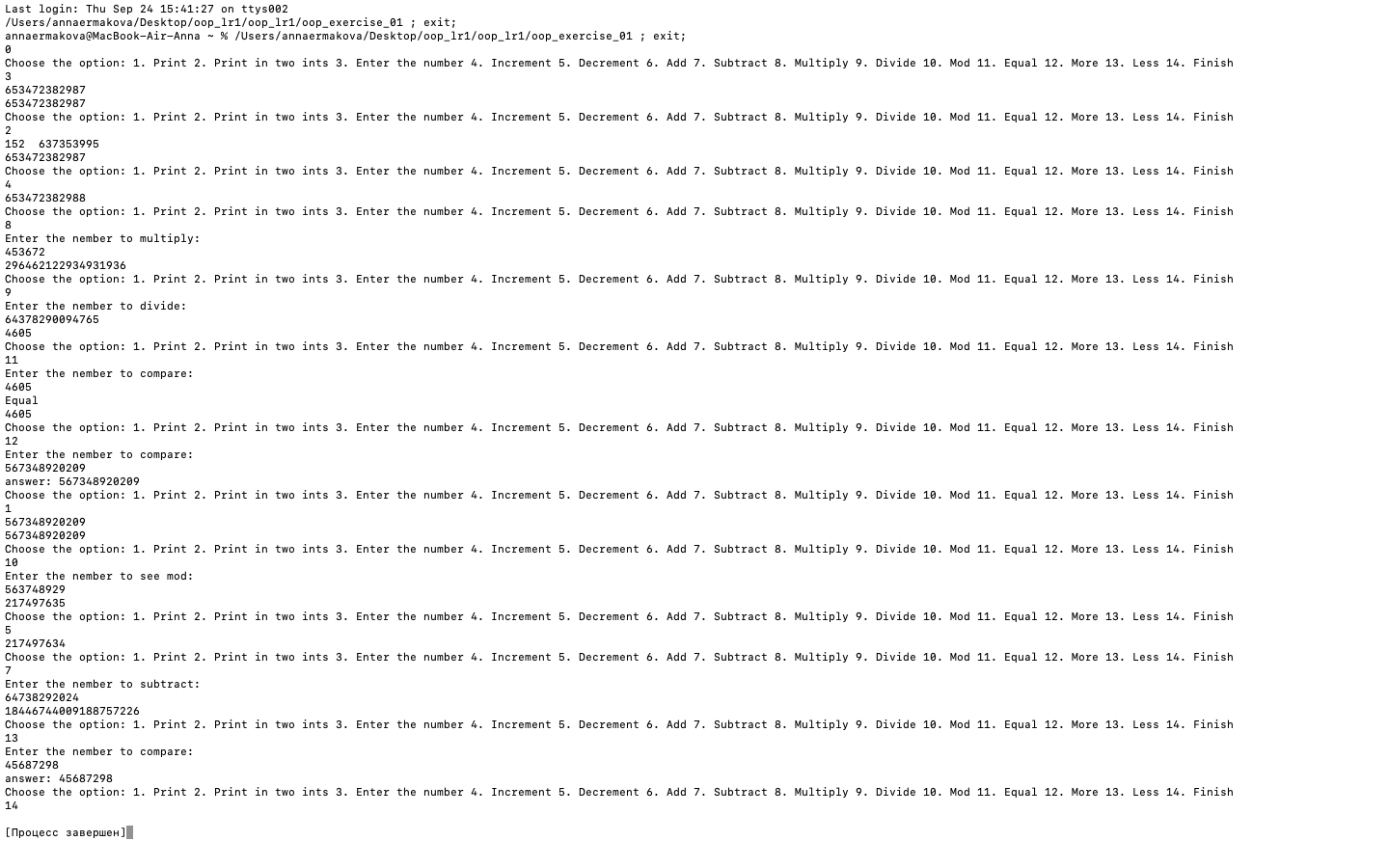
После // комментарий.

Красным обозначены входные данные, относящиеся к управлению меню.

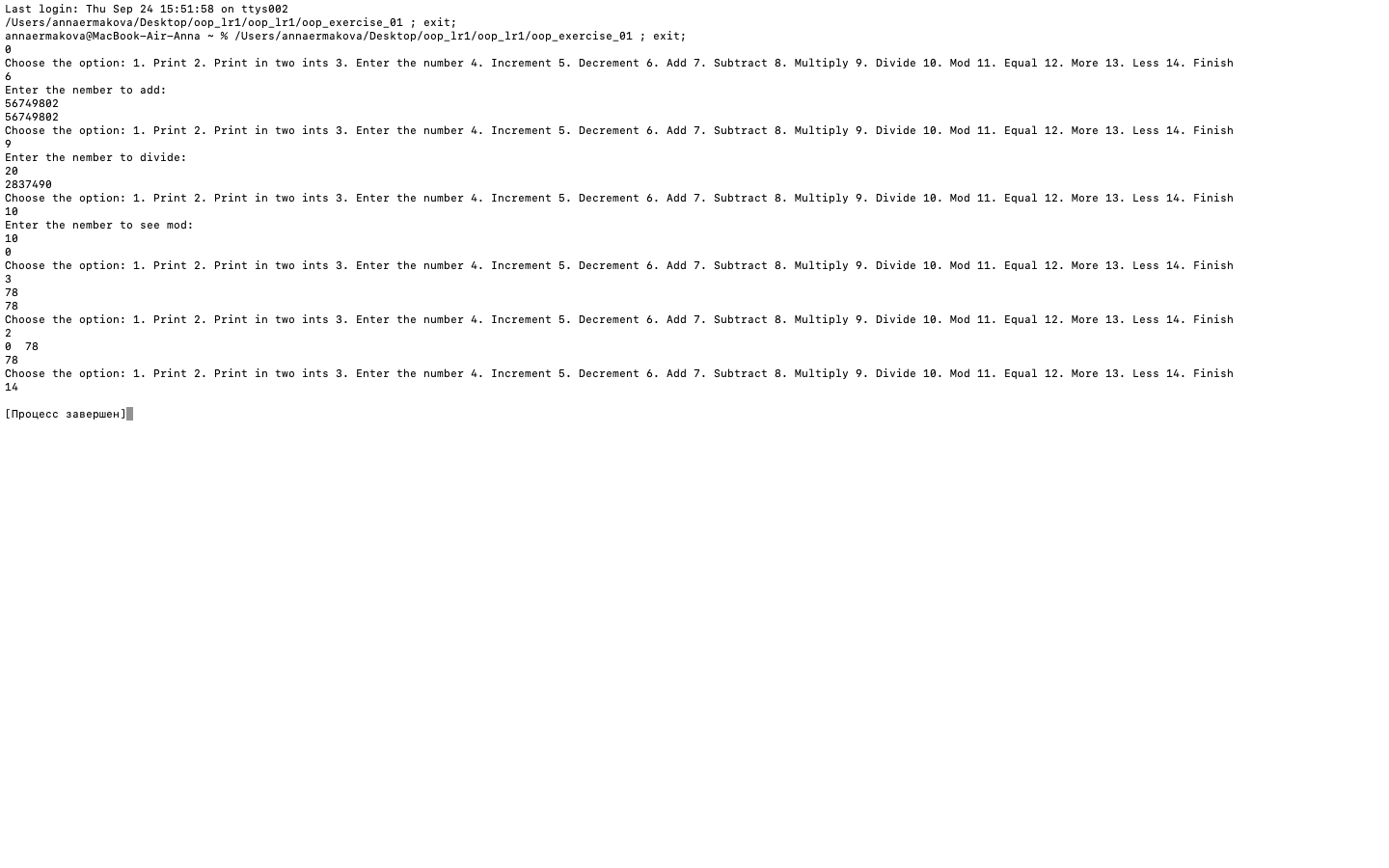
|  |  |  |
| --- | --- | --- |
|  | Входные данные: | Выходные данные: |
| тест 1 | 3//ввод  653472382987  2//печать в Long  4//++  8//\*  453672  9//:  64378290094765  11//=?  4605  12//>  567348920209  1//print  10//%  563748929  5//--  7//-  64738292024  13//<  45687298  14//finish | 653472382987  152 637353995  653472382987  653472382988  296462122934931936  4605  Equal  4605  answer: 567348920209  567348920209  567348920209  217497635  217497634  18446744009188757226  answer: 45687298 |
| тест 2 | 6//+  56749802  9//:  20  10//%  10  3//ввод  78  2//печать в Long  14 | 56749802  2837490  0  78  0 78  78 |
|  |  |  |

4. Результаты выполнения тестов.

**Тест 1:**

****

**Тест 2:**

****

5. Листинг программы:

main.cpp:

*//Ермакова Анна*

*//М80-201Б-19*

*//Создать класс Long для работы с целыми беззнаковыми числами из 64 бит. Число должно быть представлено двумя полями unsigned int. Должны быть реализованы арифметические операции, присутствующие в С++, и сравнения.*

#include "Long.hpp"

**enum** Options {

PRINT,

PRINTINT,

ENTERNUMBER,

INCREMENT,

DECREMENT,

ADD,

SUBTRACT,

MULTIPLY,

DIVIDE,

MOD,

EQUAL,

MORE,

LESS

};

Long menu(Options option, Long a){

*//реализация вынесена в отдельный блок, для большего удобства, switch перключается в зависимости от тебуемой операции*

switch(option){

case PRINT:

a.Print();

**break**;

case PRINTINT:

a.PrintINT();

**break**;

case ENTERNUMBER:

{

**unsigned** **long** int x = 0;

int high = 0, low = 0;

cin >> x;

*//для того чтобы разбить long на два int используем побитовый сдвиг на 32 (тк поделить надо пополам, а в Long 64)*

high = x >> 32;

low = (x << 32) >> 32;

a.EqualInt(high, low);

return a;

}

case INCREMENT:

{

a = a.Increment();

return a;

}

case DECREMENT:

{

a = a.Decrement();

return a;

}

case ADD:

{

Long b;

cout << "Enter the nember to add:" << endl;

b = menu(ENTERNUMBER, b);

a = a.Add(b);

return a;

}

case SUBTRACT:

{

Long b;

cout << "Enter the nember to subtract:" << endl;

b = menu(ENTERNUMBER, b);

a = a.Subtract(b);

return a;

}

case MULTIPLY:

{

Long b;

cout << "Enter the nember to multiply:" << endl;

b = menu(ENTERNUMBER, b);

a = a.Multiply(b);

return a;

}

**break**;

case DIVIDE:

{

Long b;

cout << "Enter the nember to divide:" << endl;

b = menu(ENTERNUMBER, b);

a = a.Divide(b);

return a;

}

**break**;

case MOD:

{

Long b;

cout << "Enter the nember to see mod:" << endl;

b = menu(ENTERNUMBER, b);

a = a.Mod(b);

return a;

}

**break**;

case EQUAL:

{

Long b;

cout << "Enter the nember to compare:" << endl;

b = menu(ENTERNUMBER, b);

if(a.EqualLong(b))

cout << "Equal" << endl;

else

cout << "Not equal" << endl;

return a;

}

case MORE:

{

Long b;

cout << "Enter the nember to compare:" << endl;

b = menu(ENTERNUMBER, b);

a = a.More(b);

cout << "answer: ";

return a;

}

case LESS:

{

Long b;

cout << "Enter the nember to compare:" << endl;

b = menu(ENTERNUMBER, b);

a = a.Less(b);

cout << "answer: ";

return a;

}

}

return a;

}

int main() {

int c = 1;

Long a;

while(c == 1){

*//цикл для неоднокраного использования меню в ходе работы программы*

int t;

a.Print();

cout << "Choose the option: 1. Print 2. Print in two ints 3. Enter the number 4. Increment 5. Decrement 6. Add 7. Subtract 8. Multiply 9. Divide 10. Mod 11. Equal 12. More 13. Less 14. Finish" << endl;

cin >> t;

switch(t){

*//меню в соответсвии с введенным пользователем числом*

case 1:

menu(PRINT, a);

**break**;

case 2:

menu(PRINTINT, a);

**break**;

case 3:

a = menu(ENTERNUMBER, a);

**break**;

case 4:

a = menu(INCREMENT, a);

**break**;

case 5:

a = menu(DECREMENT, a);

**break**;

case 6:

a = menu(ADD, a);

**break**;

case 7:

a = menu(SUBTRACT, a);

**break**;

case 8:

a = menu(MULTIPLY, a);

**break**;

case 9:

a = menu(DIVIDE, a);

**break**;

case 10:

a = menu(MOD, a);

**break**;

case 11:

a = menu(EQUAL, a);

**break**;

case 12:

a = menu(MORE, a);

**break**;

case 13:

a = menu(LESS, a);

**break**;

case 14:

c = 0;

**break**;

default:

cout << "Try again" << endl;

**break**;

}

}

return 0;

}

Long.hpp:

#ifndef Long\_hpp

#define Long\_hpp

#include <iostream> *//для ввода/вывода*

#include <math.h> *//для использования математических операции в Long.cpp*

**using** **namespace** std; *//все функции из пространства имен std в глобальные*

**const** **unsigned** int MAX = pow(2,32);

**class** Long {

private:

*//атрибуты класса*

**unsigned** int high;

**unsigned** int low;

public:

*//методы класса*

Long();

**void** PrintINT();

**void** Print();

Long EqualInt(int high, int low);

Long Eq(Long b);

Long Increment();

Long Decrement();

Long Add(Long a);

Long Subtract(Long a);

Long Multiply(Long a);

Long Divide(Long a);

Long Mod(Long a);

**bool** EqualLong(Long b);

Long More(Long a);

Long Less(Long a);

};

#endif */\* Long\_hpp \*/*

Long.cpp :

#include "Long.hpp"

Long::Long() {*//конструктор*

**this**->high = 0;

**this**->low = 0;

}

**void** Long::Print() {*//вывод число в 10-чной сс, поэтому используется побитовый сдвиг для преобразования элемента класса Long в unsigned long int*

**unsigned** **long** int x = 0;

x = **this**->high;

x = x << 32;

x = x + **this**->low;

cout << x << endl;

}

**void** Long::PrintINT() {*//вывод элемента класса Long*

cout << high << " " << low << endl;

}

Long Long::Eq(Long b) {*//призваивание Long = Long*

**this**->high = b.high;

**this**->low = b.low;

return \***this**;

}

Long Long::EqualInt(int high, int low) {*//призваивание Long = Long*

**this**->high = high;

**this**->low = low;

return \***this**;

}

**bool** Long::EqualLong(Long b) {*//проверка на равенство*

if (**this**->high == b.high && **this**->low == b.low)

return 1;

else

return 0;

}

Long Long::Increment() {*//+1*

Long b;

b.low = 1;

**this**->Add(b);

return \***this**;

}

Long Long::Decrement() {*//-1*

Long a = \***this**;

Long b;

b.low = 1;

return a.Subtract(b);

}

Long Long::Add(Long b) {*//сложение*

Long a = \***this**;

**this**->low += b.low;

**this**->high += b.high;

*//сложили верхнюю и нижнюю части*

if (**this**->low < a.low || **this**->low < b.low)

*//если часть low при сложении уменьшилась, значит произошло переполнение -> необходимо увеличить на 1 часть high*

**this**->high++;

return \***this**;

}

Long Long::Subtract(Long b) {*//вычитание*

Long x = \***this**;

Long y = b;

*//начинаем с части low как при вычитании в столбик*

if (x.low >= y.low)

x.low -= y.low;

else if (x.low < y.low)

{*//если уменьшаемое меньше вычитаемого занимаем 1 у части high (в этой сист счисл 1 = 2^32)*

x.low = MAX - y.low + x.low;

x.high--;

}

x.high -= y.high;

return x;

}

Long Long::Multiply(Long b) {*//умножение*

**unsigned** int alow = **this**->low;

**unsigned** int ahigh = **this**->high;

**unsigned** int blow = b.low;

**unsigned** int bhigh = b.high;

*//отдельно обозначили high и low части множителей*

**unsigned** int p = 0;

Long res;

if(b.high != 0 || b.low !=0)*//если умножение не на ноль*

{

int k = 1;

int c = 0; *//счетчик переполнения*

*//имитируем умножение в столбик*

*// ah al*

*// bh bl*

*//-----------------------*

*// al\*bh al\*bl*

*//ah\*bh ah\*bl*

*//ah\*bh уже не входит в Long*

*//сначала a.low\*b*

*//для этого небходимо a.low\*b и посчитать сколько раз произойтет переполнение ->*

*//используем цикл - умножение представим как увеличение на b а раз*

while (alow != k){

p = blow;

blow += b.low;

if (blow < p)*//если после очередного суммирования а уменьшилось-> произошло переполнение ->c++*

c++;

k++;

}

res.low = blow;

*//чтобы досчитать осталось a.low\*b.high*

res.high = alow \* bhigh + c;

*//далее a.high\*b.low*

blow = b.low;

blow \*= ahigh;

res.high += blow;

}

*//а если было умножение на 0 то res изначально и есть 0*

return res;

}

Long Long::Divide(Long b) {*//деление*

Long a = \***this**;

Long p;

Long res;

if(a.EqualLong(b))

res = res.Increment(); *//необходимо, тк дальше программа приравных а и b пропустит один раз увеличение*

if(a.EqualLong(a.More(b)))*// если делимое больше делителя*

{

int c = 0;

*//деление придставим как вычитание делителя из делимого res раз*

while (c == 0){

p = p.Eq(a);

a = a.Subtract(b); *//вычитаем из а b*

if (a.EqualLong(p.More(a)) || (a.low == 0 && a.high == 0)) *//если после вычитания число увеличилость или стало 0 -> произошло переполнение или число поделилось нацело*

{c++; res = res.Decrement();}

res = res.Increment();

}

}

return res;

}

Long Long::Mod(Long b) {*//остаток от деления*

Long a = \***this**;

Long p;

if(a.EqualLong(a.Less(b)))*//если делитель больше делимого то остаток равен делимому*

{

return a;

}

if(a.EqualLong(a.More(b)))

{

int c = 0;

*//повтор алгоритма деления, только теперь ответом будет не счетчик, а то, что останется от делимого*

while (c == 0){

p = p.Eq(a);

a = a.Subtract(b);

if (a.EqualLong(p.More(a)) || (a.low == 0 && a.high == 0))

{c++; }

}

if(a.low == 0 && a.high == 0) return a;

}

return p;

}

Long Long::Less(Long b) {*//вернет меньшее из двух чисел*

if (**this**->high < b.high)

return \***this**;

else if (**this**->high == b.high && **this**->low < b.low)

return \***this**;

else return b;

}

Long Long::More(Long b) { *//вернет большее из двух чисел*

if(b.EqualLong((\***this**).Less(b)))

return \***this**;

else return b;

}

6. Вывод:

Во время лабораторной работы я поняла как работать с классами, вызывать методы и функции класса. Как создание класса помогает работать с объектами ,с одинаковыми свойствами.

7. Список литературы.

1.Представление информации в эвм [Электронный ресурс] URL: <https://studfile.net/preview/4114123/page:7/> (дата обращения: 20.09.2020).

2.Целочисленные типы данных: short, int и long [Электронный ресурс] URL: <https://ravesli.com/urok-31-tselochislennyj-tip-dannyh-integer/> (дата обращения: 20.09.2020).

3. Введение в CMake [Электронный ресурс] URL: <https://habr.com/ru/post/155467/> (дата обращения: 23.09.2020).