Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра 806 «Вычислительная информатика и программирование»

Факультет: «Информационные технологии и прикладная математика»

Лабораторная работа

Дисциплина: «Объектно-ориентированное программирование»

I I I семестр

Задание 1: «Операторы, литералы»

|  |  |
| --- | --- |
| Группа: | М8О-208Б-18, №13 |
| Студент: | Ли Алиса Игоревна |
| Преподаватель: | Журавлёв Андрей Андреевич |
| Оценка: |  |
| Дата: | 25.11.2019 |

Москва, 2019

1. **Задание**

**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.

Операции необходимо реализовать в виде перегрузки операторов.

Необходимо реализовать пользовательский литерал для работы с константами типа Long.

1. **Адрес репозитория на GitHub**

<https://github.com/elips0n/oop_laba2>

1. **Код программы на С++**

main.cpp

#include <iostream>

#include "Long.h"

int main() {

Long a, b;

std::cout << "Введите первое число: ";

a.read(std::cin);

std::cout << '\n';

std::cout << "Введите второе число: ";

b.read(std::cin);

std::cout << '\n';

std::cout << "Первое число: ";

a.write(std::cout);

std::cout << '\n';

std::cout << "Второе число: ";

b.write(std::cout);

std::cout << '\n';

Long sum = a.add(b);

std::cout << "Сумма чисел: ";

sum.write(std::cout);

std::cout << '\n';

Long razn = a.sub(b);

std::cout << "Разность чисел: ";

razn.write(std::cout);

std::cout << '\n';

Long umn = a.mult(b);

std::cout << "Произведение чисел: ";

umn.write(std::cout);

std::cout << '\n';

Long del = a.div(b);

std::cout << "Частное чисел: ";

del.write(std::cout);

std::cout << '\n';

if (a.less(b)) std::cout << "Первое число меньше второго" << std::endl;

if (a.more(b)) std::cout << "Первое число больше второго" << std::endl;

if (a.equal(b)) std::cout << "Числа равны" << std::endl;

}

Long.h

#ifndef LABA1\_LONG\_H

#define LABA1\_LONG\_H

#include <cstdint>

#include <iostream>

#include <cmath>

class Long {

private:

uint32\_t first, second;

uint64\_t glue() const;

void split(uint64\_t n);

public:

void read(std::istream& is);

void write(std::ostream& os) const;

Long add (const Long& o) const; // сложение

Long sub (const Long& o) const; // модуль разности

Long mult (const Long& o) const;// умножение

Long div (const Long& o) const; // деление

bool equal (const Long& o) const; // равно

bool more (const Long& o) const; // больше

bool less (const Long& o) const; // меньше

};

#endif //LABA1\_LONG\_H

Long.cpp

#include <climits>

#include "Long.h"

void Long::read(std::istream& is) {

is >> first >> second;}

void Long::write(std::ostream &os) const {

if (first != 0) {

os << first << " ";

}

os << second;

}

int f(uint32\_t n) { // подсчет разрядов числа

int result = 0;

while (n >= 10){

++result;

n /= 10;

}

return result;

}

uint64\_t Long::glue() const{ // склейка чисел

uint64\_t result;

result = ((uint64\_t)first << 32u) + (uint64\_t)second;

return result;

}

void Long::split(uint64\_t n) { // расклейка чисел

first = n >> 32u;

second = n & (0u - 1u);

}

Long Long::add(const Long& o) const {

Long result;

result.second = this->second + o.second;

result.first = this->first + o.first;

if (std::numeric\_limits<uint32\_t>::max() - this->second < o.second) {

result.first += 1;

}

return result;

}

Long Long::sub(const Long& o) const {

Long result;

result.first = this->first - o.first;

result.second = this->second - o.second;

if (this->second < o.second) {

result.first -= 1;

}

return result;

}

Long Long::mult(const Long &o) const {

Long result;

uint64\_t a = glue();

uint64\_t b = o.glue();

uint64\_t c = a \* b;

result.split(c);

return result;

}

Long Long::div(const Long& o) const {

Long result;

uint64\_t a = glue();

uint64\_t b = o.glue();

uint64\_t c = a / b;

result.split(c);

return result;

}

bool Long::less(const Long& o) const {

return (this->first < o.first || (this->first == o.first && this->second < o.second));

}

bool Long::more(const Long& o) const {

return (this->first > o.first || (this->first == o.first && this->second > o.second));

}

bool Long::equal(const Long& o) const {

return (this->first == o.first && this-> second == o.second);

}

CMakeLists.txt

cmake\_minimum\_required(VERSION 3.14)

project(laba1)

set(CMAKE\_CXX\_STANDARD 14)

add\_executable(laba1 main.cpp Long.cpp Long.h)

test1.txt

5 5

3 7

test2.txt

9 9

2 17

1. **Результаты выполнения тестов**

test1.txt

Введите первое число: 5 5

Введите второе число: 3 7

Первое число: 5 5

Второе число: 3 7

Сумма чисел: 8 12

Разность чисел: 1 4294967294

Произведение чисел: 50 35

Частное чисел: 1

Первое число больше второго

test2.txt

Введите первое число: 9 9

Введите второе число: 2 17

Первое число: 9 9

Второе число: 2 17

Сумма чисел: 11 26

Разность чисел: 6 4294967288

Произведение чисел: 171 153

Частное чисел: 4

Первое число больше второго

1. **Объяснение результатов работы программы**

Программа просит на вход два параметра, после чего выводятся вводимые данные, вычисления, производимые с ними, и выход.

1. **Вывод**

Были изучены основы объектно-ориентированного программирования, методы, классы. Написан простой класс Long, который реализует арифметические действия и сравнения с целыми беззнаковыми числами из 64 бит, которые представлены двумя полями unsigned int. Изучены операторы перегрузки и пользовательские литералы. Во время работы над лабораторной возникало множество трудностей, но благодаря невероятному терпению преподавателя и его умению объяснять, было найдено их решение. ☺