Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра 806 «Вычислительная информатика и программирование»

Факультет: «Информационные технологии и прикладная математика»

Лабораторная работа

Дисциплина: «Объектно-ориентированное программирование»

3 семестр

Задание 1: «Простые классы»

|  |  |
| --- | --- |
| Группа: | М8О-206Б-19 |
| Студент: | Кондратьев Егор Алексеевич |
| Преподаватель: | [Чернышов Лев Николаевич](https://mai.ru/education/schedule/ppc.php?guid=6c63f791-1d99-11e0-9baf-1c6f65450efa) |
| Оценка: |  |

Москва, 2020

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.

1. **GitHub**

<https://github.com/Dukend/OOP>

1. **Код программы на С++**

///////////////

// by Dukend

///////////////

#include <bits/stdc++.h>

class Long

{

private:

    uint64\_t first, second;

    uint64\_t glue() const

    { // склейка чисел

        uint64\_t result;

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

        return result;

    }

    void split(uint64\_t n)

    { // расклейка чисел

        first = n >> 32u;

        second = n & (0u - 1u);

    }

public:

    void read()

    {

        std::cin >> first >> second;

        if (this->first > std::numeric\_limits<uint32\_t>::max() || this->first < 0)

        {

            std::cout << "В числе ошибка" << std::endl;

            exit(1);

        }

        if (this->second > std::numeric\_limits<uint32\_t>::max() || this->second < 0)

        {

            std::cout << "В числе ошибка" << std::endl;

            exit(1);

        }

    }

    void write() const

    {

        if (first != 0)

        {

            std::cout << first << " ";

        }

        std::cout << second;

    }

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

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

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

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

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

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

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

};

Long Long::add(const Long &next) const

{

    Long result;

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

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

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

    {

        result.first += 1;

    }

    return result;

}

Long Long::sub(const Long &next) const

{

    Long result;

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

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

    if (this->second < next.second)

    {

        result.first -= 1;

    }

    return result;

}

Long Long::mult(const Long &next) const

{

    Long result;

    uint64\_t a = glue();

    uint64\_t b = next.glue();

    uint64\_t c = a \* b;

    result.split(c);

    return result;

}

Long Long::div(const Long &next) const

{

    Long result;

    uint64\_t a = glue();

    uint64\_t b = next.glue();

    uint64\_t c = a / b;

    result.split(c);

    return result;

}

bool Long::more(const Long &next) const

{

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

}

bool Long::less(const Long &next) const

{

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

}

bool Long::equal(const Long &next) const

{

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

}

int main()

{

    Long a, b;

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

    a.read();

    std::cout << '\n';

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

    b.read();

    std::cout << '\n';

    std::cout << '\n';

    std::cout << "Первое число: ";

    a.write();

    std::cout << '\n';

    std::cout << "Второе число: ";

    b.write();

    std::cout << '\n';

    std::cout << '\n';

    Long ans;

    ans = a.add(b);

    std::cout << "Сумма чисел: ";

    ans.write();

    std::cout << '\n';

    ans = a.sub(b);

    std::cout << "Разность чисел: ";

    ans.write();

    std::cout << '\n';

    ans = a.mult(b);

    std::cout << "Произведение чисел: ";

    ans.write();

    std::cout << '\n';

    ans = a.div(b);

    std::cout << "Частное чисел: ";

    ans.write();

    std::cout << '\n';

    std::cout << '\n';

    if (a.more(b))

        std::cout << "Первое число больше второго" << std::endl;

    if (a.less(b))

        std::cout << "Первое число меньше второго" << std::endl;

    if (a.equal(b))

        std::cout << "Числа равны" << std::endl;

    return 0;

}

test1.txt

13 23

4 56

test2.txt

22334 534

1234 7654

1. **Результаты выполнения тестов**

**test1.txt**

Введите первое число: 13 23

Введите второе число: 4 56

Первое число: 13 23

Второе число: 4 56

Сумма чисел: 17 79

Разность чисел: 8 4294967263

Произведение чисел: 820 1288

Частное чисел: 3

Первое число больше второго

**test2.txt**

Введите первое число: 22334 534

Введите второе число: 1234 7654

Первое число: 22334 534

Второе число: 1234 7654

Сумма чисел: 23568 8188

Разность чисел: 21099 4294960176

Произведение чисел: 171603392 4087236

Частное чисел: 18

Первое число больше второго

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.