**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Кондратьев Егор Алексеевич

Группа: 80-206

Преподаватель: Чернышов Л.Н.

Дата:

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.

1. **Описание программы**

считывает 4 числа

первые 2 числа - первое число unsigned int

вторые 2 числа - второе число unsigned int

далее происходит склейка или разделение чисел(если необходимо)

Далее выводит

сами числа

сумма чисел

разность чисел

произведение чисел

частное чисел

и

сравнение

1. **Набор тестов**

**test1.txt**

Первое число: 45123123123

Второе число: 44123612362

**test2.txt**

Первое число: 124524

Второе число: 43865823643846

1. **Результаты выполнения тестов**

**test1.txt**

Первое число: 45123123123

10 2173450163

Второе число: 44123612362

10 1173939402

Сумма чисел: 89246735485

20 3347389565

Разность чисел: 999510761

0 999510761

Произведение чисел: 17193577355128823614

4003191682 1319591742

Частное чисел: 1

0 1

Первое число больше второго

**test2.txt**

Первое число: 124524

0 124524

Второе число: 43865823643846

10213 1322649798

Сумма чисел: 43865823768370

10213 1322774322

Разность чисел: 43865823519322

10213 1322525274

Произведение чисел: 5462347823426279304

1271801959 2532546440

Частное чисел: 0

0 0

Первое число меньше второго

1. **Листинг программы**

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

// by Dukend

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

#include <bits/stdc++.h>

class Long

{

private:

    uint32\_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() // чтение и проверка

    {

        uint64\_t tmp;

        std::cin >> tmp;

        if (tmp > std::numeric\_limits<uint64\_t>::max() || tmp < 0)

        {

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

            exit(1);

        }

        first = tmp >> 32u;

        second = tmp & (0u - 1u);

    }

    void write() const

    {

        uint64\_t a = glue();

        std::cout << a << '\n';

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

    uint64\_t a = glue();

    uint64\_t b = next.glue();

    if (a >= b)

    {

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

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

        if (this->second < next.second)

        {

            result.first -= 1;

        }

    }

    else

    {

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

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

        if (next.second < this->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;

}

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.

1. **Литература**

Класс C++ - https://code-live.ru/post/cpp-classes/