Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.10**

Дисциплина: Основы теории алгоритмов и структуры данных.

Тема: “ Объектно-ориентированное программирование.

Сохранение данных в файле с использованием потоков.”

Вариант.12

Выполнила работу:

студент группы ИВТ-20-2Б

Ананина Арина Юрьевна

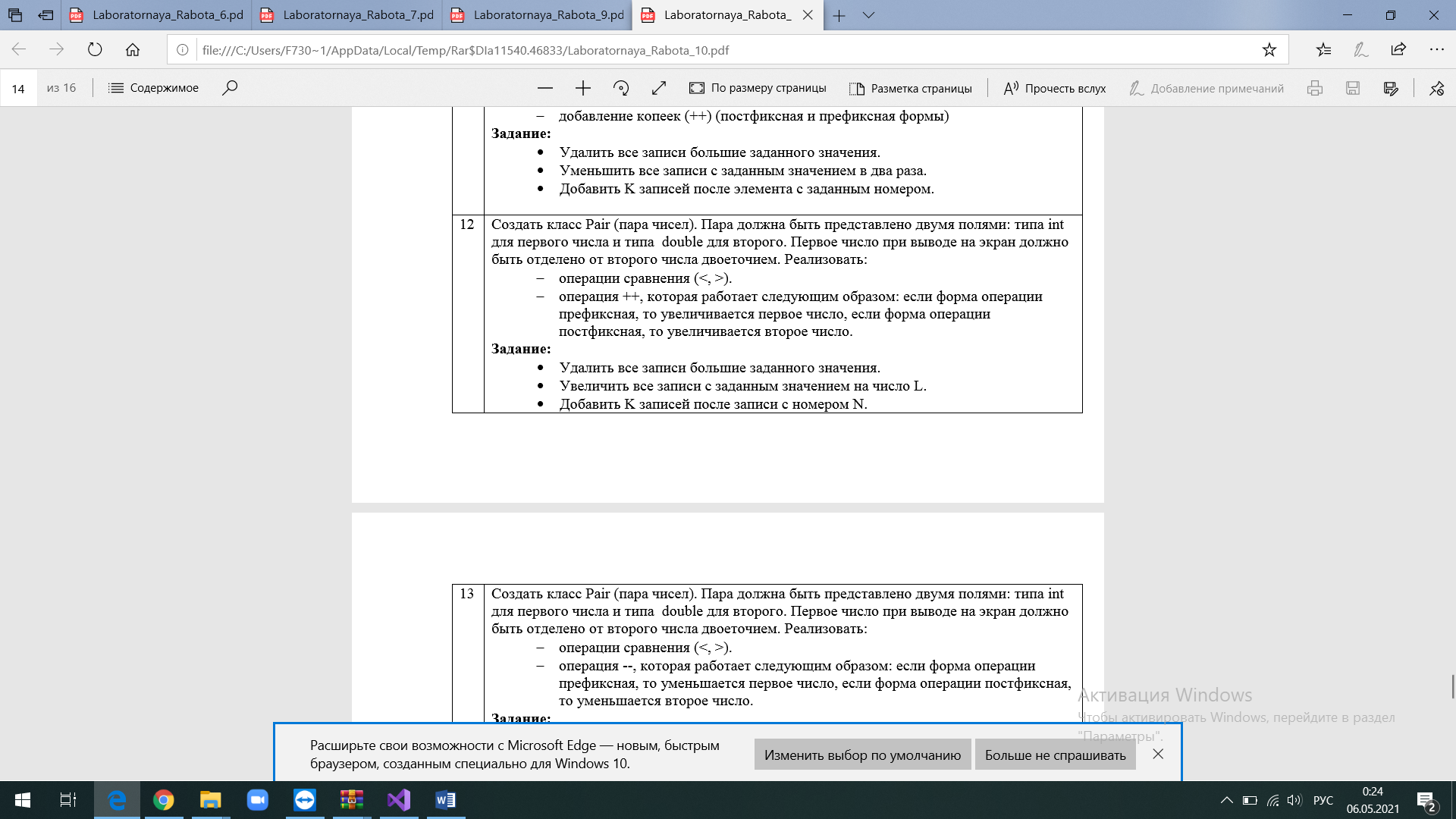
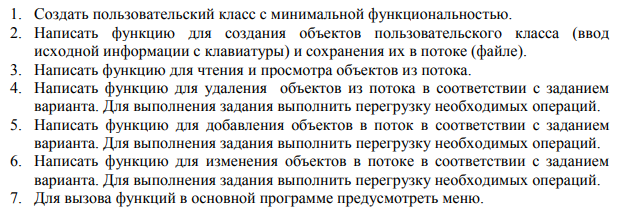
Проверила:

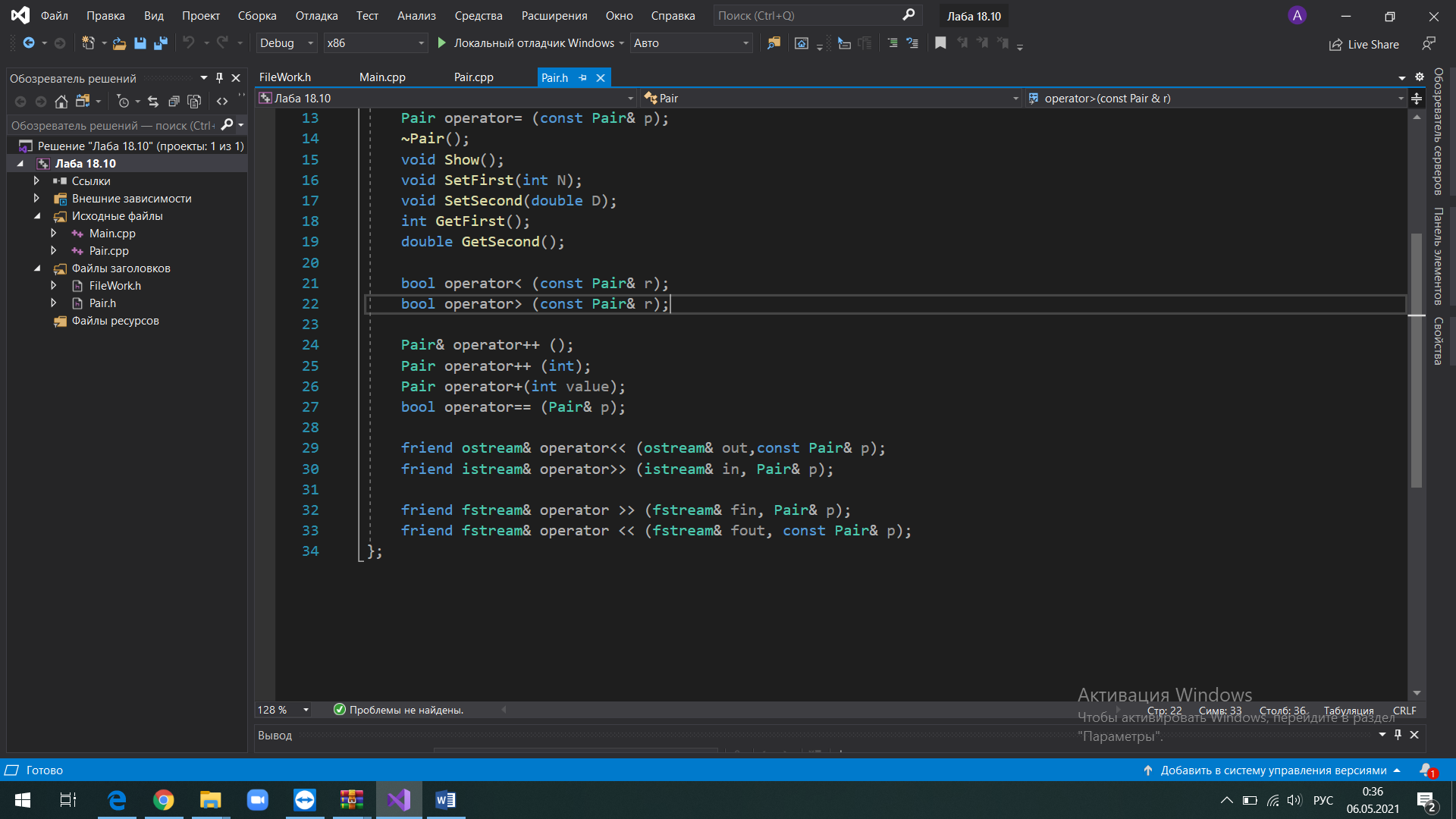
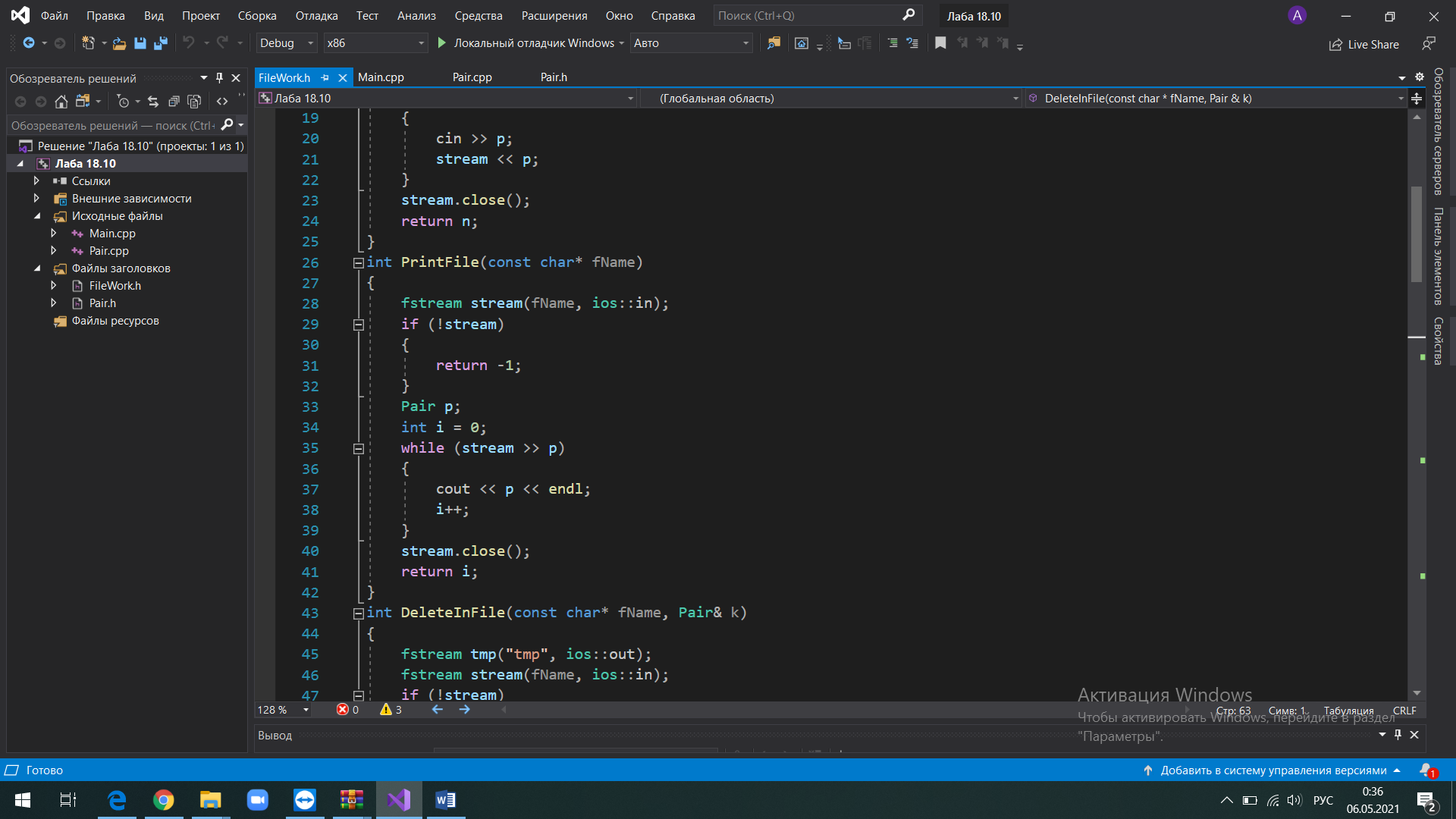
Доцент кафедры ИТАС

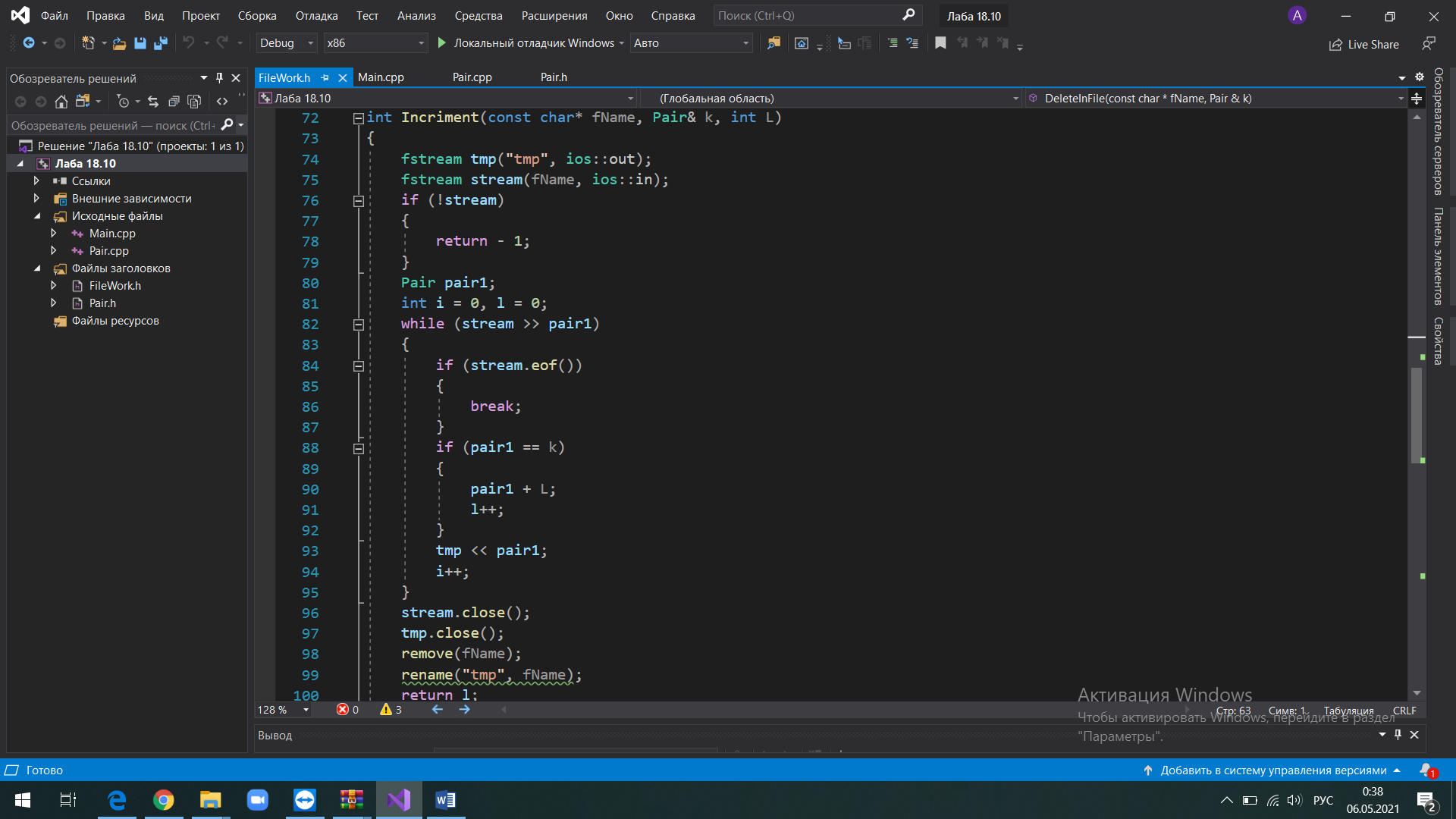
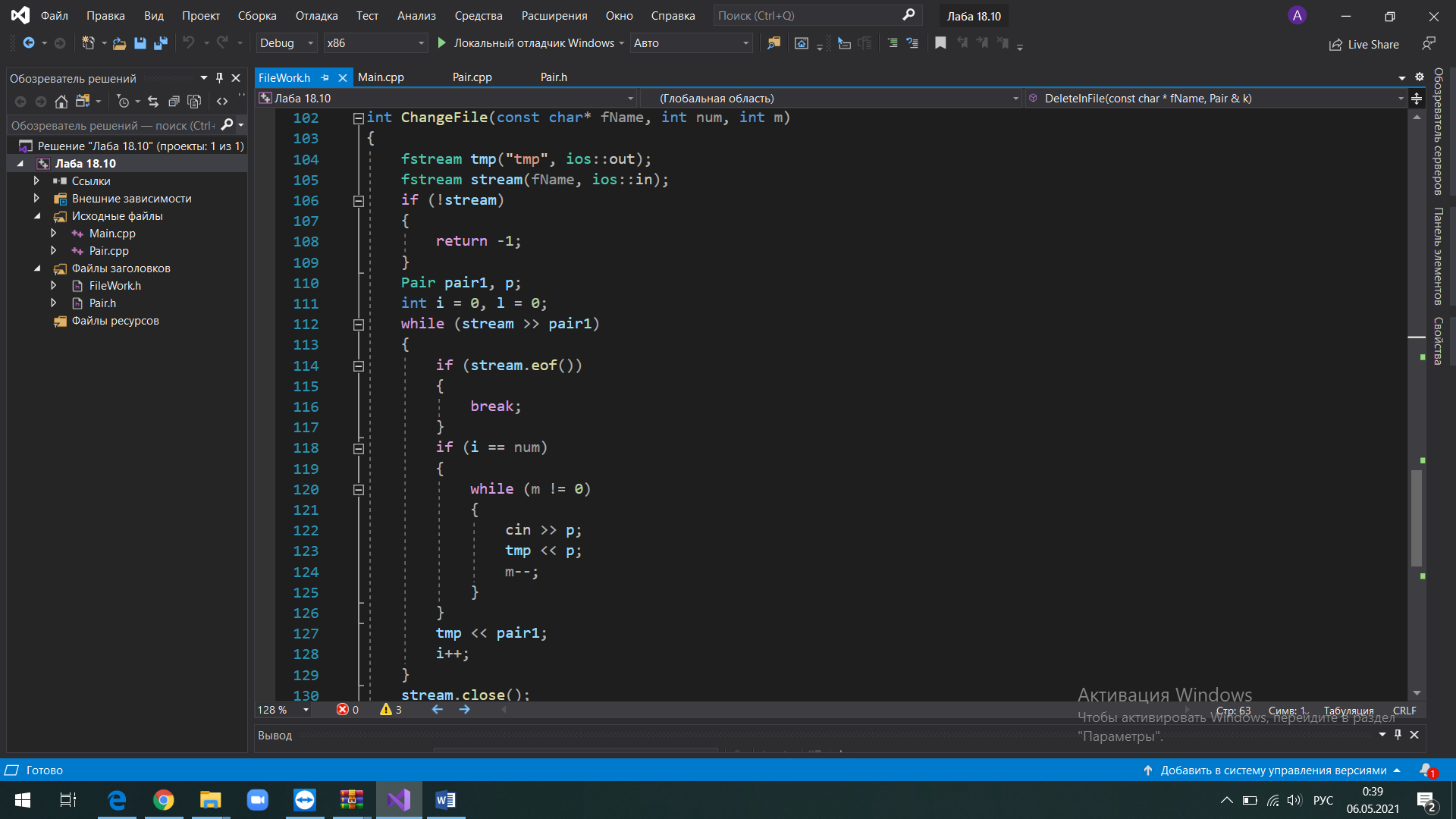
Полякова О.А.

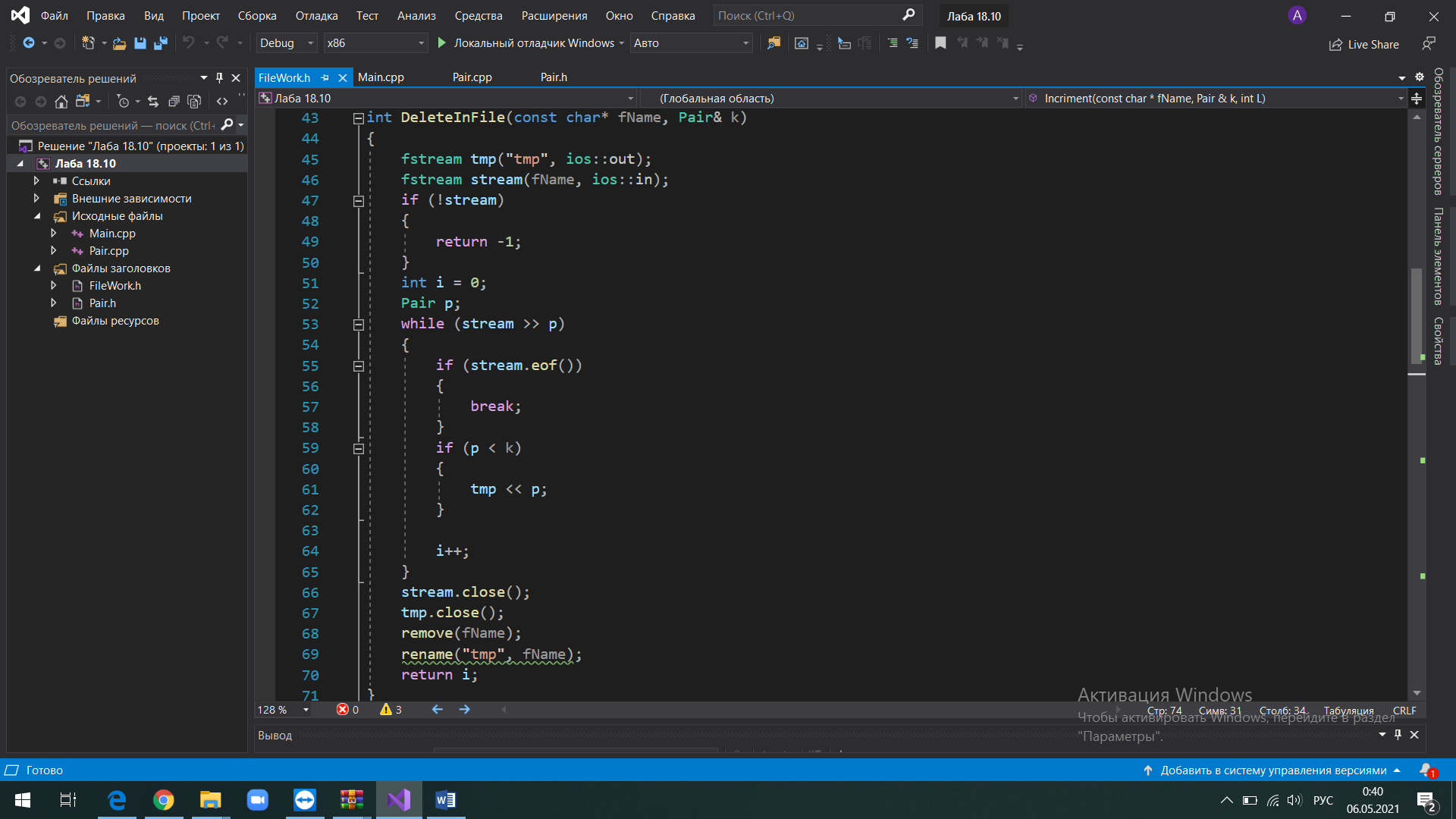
Пермь, 2021

1. **Постановка задачи**



1. **Анализ задачи**
2. Для решения задачи необходимо:
   1. Использовать класс Pair.
   2. Реализовать функции для решения задач из варианта.
3. В ходе работы были использованы следующие типы данных:
   1. Класс Pair для хранения чисел типа int и double.
   2. Переменная типа string для хранения имени файла
4. Для решения задачи данные были представлены в следующем виде:
   1. Пары чисел записаны в файл.
5. Для операций ввода и вывода использовались следующие операторы и функции:
   1. Для открытия файла на чтение и запись использовать потоки ifstream и ofstream.  
      
   2. Для вывода пар чисел на консоль использовать функцию PrintFile().  
      

1. Поставленные задачи будут решены следующими действиями:
   1. Функция добавления константы ко всем парам из файла   
      
   2. Функция добавления пары в файл после определенной позиции  
      
   3. Функция удаления пар из файла



1. **Код программы**

FileWork.h

#pragma once

#include <iostream>

#include "Pair.h"

#include <string>

#include <fstream>

using namespace std;

int MakeFile(const char\* fName)

{

fstream stream(fName, ios::out | ios::trunc);

if (!stream)

{

return -1;

}

int n;

Pair p;

cout << "Введите кол-во пар чисел: ";

cin >> n;

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

{

cin >> p;

stream << p;

}

stream.close();

return n;

}

int PrintFile(const char\* fName)

{

fstream stream(fName, ios::in);

if (!stream)

{

return -1;

}

Pair p;

int i = 0;

while (stream >> p)

{

cout << p << endl;

i++;

}

stream.close();

return i;

}

int DeleteInFile(const char\* fName, Pair& k)

{

fstream tmp("tmp", ios::out);

fstream stream(fName, ios::in);

if (!stream)

{

return -1;

}

int i = 0;

Pair p;

while (stream >> p)

{

if (stream.eof())

{

break;

}

if (p < k)

{

tmp << p;

}

i++;

}

stream.close();

tmp.close();

remove(fName);

rename("tmp", fName);

return i;

}

int Incriment(const char\* fName, Pair& k, int L)

{

fstream tmp("tmp", ios::out);

fstream stream(fName, ios::in);

if (!stream)

{

return - 1;

}

Pair pair1;

int i = 0, l = 0;

while (stream >> pair1)

{

if (stream.eof())

{

break;

}

if (pair1 == k)

{

pair1 + L;

l++;

}

tmp << pair1;

i++;

}

stream.close();

tmp.close();

remove(fName);

rename("tmp", fName);

return l;

}

int ChangeFile(const char\* fName, int num, int m)

{

fstream tmp("tmp", ios::out);

fstream stream(fName, ios::in);

if (!stream)

{

return -1;

}

Pair pair1, p;

int i = 0, l = 0;

while (stream >> pair1)

{

if (stream.eof())

{

break;

}

if (i == num)

{

while (m != 0)

{

cin >> p;

tmp << p;

m--;

}

}

tmp << pair1;

i++;

}

stream.close();

tmp.close();

remove(fName);

rename("tmp", fName);

return m;

}

Main.cpp

#include "Pair.h"

#include "FileWork.h"

#include <iostream>

#include <fstream>

using namespace std;

void main()

{

setlocale(LC\_ALL, "rus");

Pair pair1, pair2, pair3, number;

int k, c;

char fName[30];

do

{

cout << "0 - Выход." << endl;

cout << "1 - Создание файла." << endl;

cout << "2 - Распечатать файл." << endl;

cout << "3 - Удалить все записи с значением больше заданного." << endl;

cout << "4 - Увеличение всех элементов." << endl;

cout << "5 - Добавление записей." << endl;

cin >> c;

switch (c)

{

case 1: cout << "Введите имя файла: "; cin >> fName;

k = MakeFile(fName);

if (k < 0)

{

cout << "Ошибка!";

}

break;

case 2: cout << "Введите имя файла: "; cin >> fName;

k = PrintFile(fName);

if (k == 0)

{

cout << "Файл пуст" << endl;

}

if (k < 0)

{

cout << "Ошибка!";

}

break;

case 3: cout << "Введите имя файла: "; cin >> fName;

cout << "Ввчедите пару для сравнения: ";

cin >> number;

k = DeleteInFile(fName, number);

if (k < 0)

{

cout << "Ошибка";

}

break;

case 4: cout << "Введите имя файла: "; cin >> fName;

cout << "Ввчедите пару для сравнения: ";

cin >> number;

cout << "Введите L: ";

int L;

cin >> L;

k = Incriment(fName, number, L);

if (k < 0)

{

cout << "Ошибка";

}

break;

case 5: cout << "Введите имя файла: "; cin >> fName;

cout << "Ввчедите номер элемента, после которого необходимо выполнить вставку: ";

int num;

cin >> num;

cout << "Введите размер вставки: ";

int m;

cin >> m;

k = ChangeFile(fName, num, m);

if (k != 0)

{

cout << "Ошибка";

}

break;

}

} while (c != 0);

}

Pair.h

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Pair

{

int First;

double Second;

public:

Pair();

Pair(int First, double Second);

Pair(const Pair& p);

Pair operator= (const Pair& p);

~Pair();

void Show();

void SetFirst(int N);

void SetSecond(double D);

int GetFirst();

double GetSecond();

bool operator< (const Pair& r);

bool operator> (const Pair& r);

Pair& operator++ ();

Pair operator++ (int);

Pair operator+(int value);

bool operator== (Pair& p);

friend ostream& operator<< (ostream& out,const Pair& p);

friend istream& operator>> (istream& in, Pair& p);

friend fstream& operator >> (fstream& fin, Pair& p);

friend fstream& operator << (fstream& fout, const Pair& p);

};

Pair.cpp

#include <iostream>

#include "Pair.h"

#include <string>

#include <fstream>

using namespace std;

Pair::Pair()

{

First = 1;

Second = 1;

}

Pair::Pair(int First, double Second)

{

this->First = First;

this->Second = Second;

}

Pair::Pair(const Pair& p)

{

this->First = p.First;

this->Second = p.Second;

}

Pair Pair:: operator= (const Pair& p)

{

if (&p == this)

{

return \*this;

}

this->First = p.First;

this->Second = p.Second;

}

Pair::~Pair() {}

void Pair::Show()

{

cout << First << " : " << Second << endl;

}

void Pair::SetFirst(int N)

{

First = N;

}

void Pair::SetSecond(double D)

{

Second = D;

}

int Pair::GetFirst()

{

return First;

}

double Pair::GetSecond()

{

return Second;

}

bool Pair::operator< (const Pair& r) {

return (First < r.First) && (Second < r.Second);

}

bool Pair::operator> (const Pair& r) {

return (First > r.First) && (Second > r.Second);

}

Pair& Pair::operator++ ()

{

First++;

return \*this;

}

Pair Pair::operator++ (int)

{

Pair old(First, Second);

Second++;

return old;

}

Pair Pair::operator+(const int value)

{

First += value;

Second += value;

return \*this;

}

bool Pair::operator== (Pair& p)

{

return (First == p.First && Second == p.Second);

}

ostream& operator << (ostream& out, const Pair& p)

{

out << p.First << " : " << p.Second << endl;

return out;

}

istream& operator>> (istream& in, Pair& p)

{

cout << "Введите первое число: "; in >> p.First;

cout << "Введите второе число: "; in >> p.Second;

return in;

}

fstream& operator>> (fstream& fin, Pair& p)

{

fin >> p.First;

fin >> p.Second;

return fin;

}

fstream& operator<< (fstream& fout,const Pair& p)

{

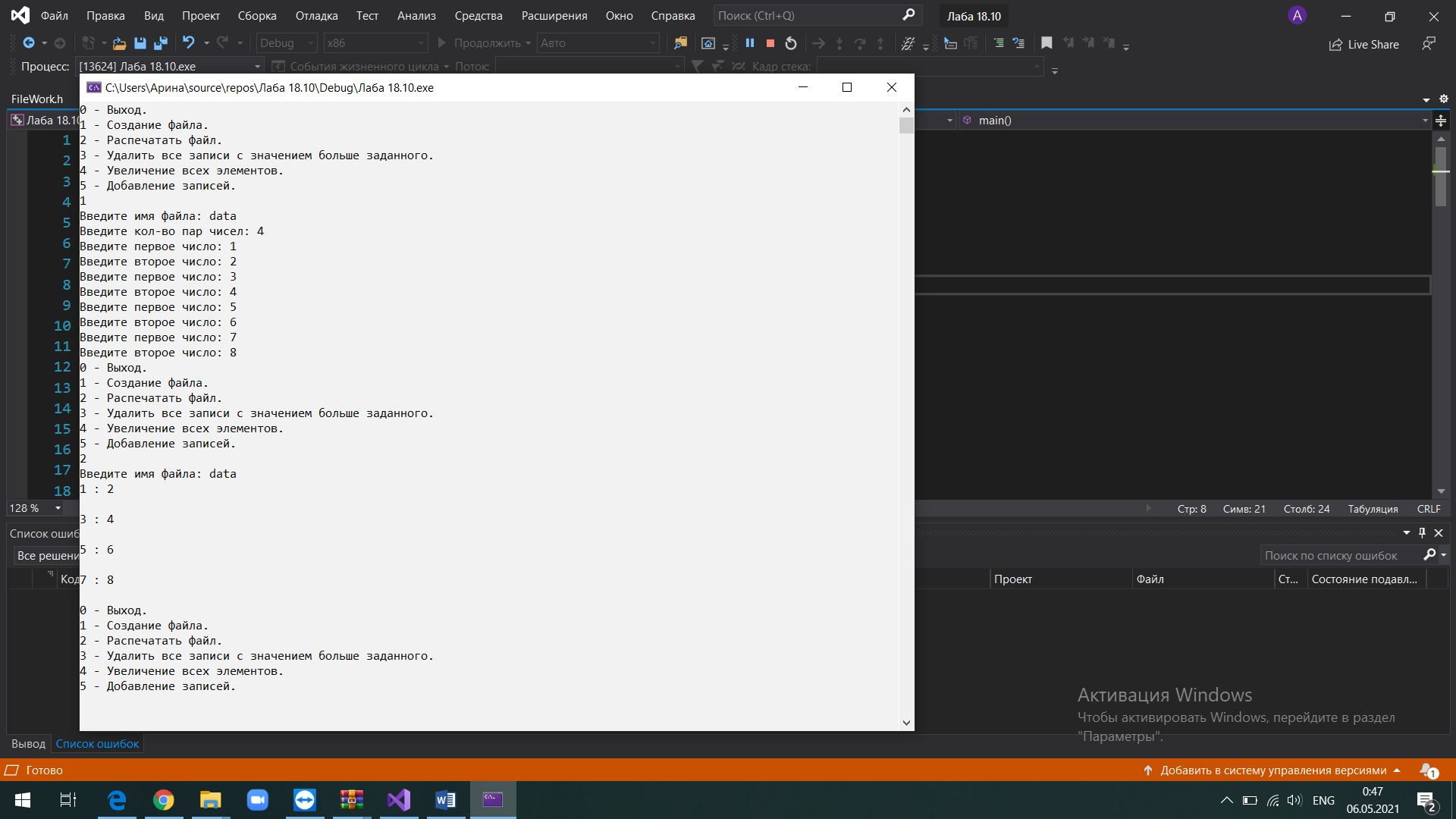
fout << p.First << " ";

fout << p.Second << endl;

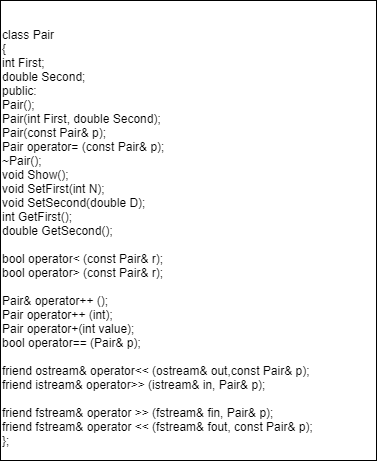
return fout;

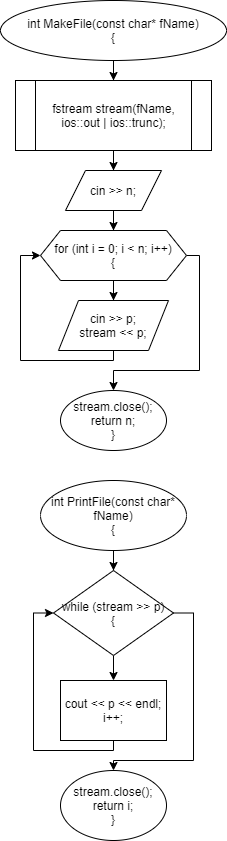
}

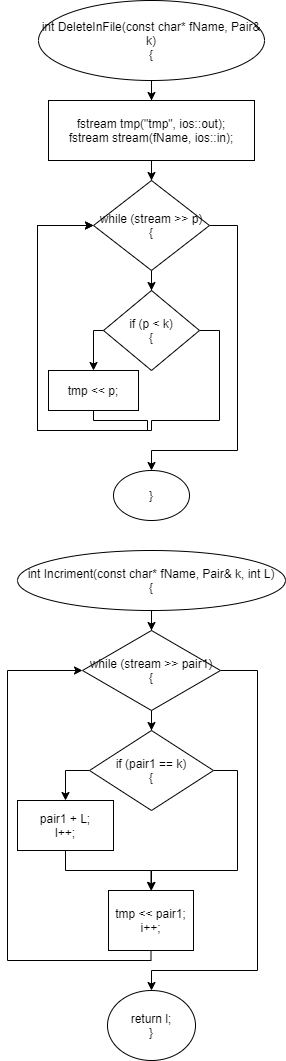
1. **Работа кода**



1. **Блок-схема**







1. **Ответы на вопросы**

