Пермский национальный исследовательский политехнический университет.

Лабораторная работа № 10 по ООП.

«Сохранение данных в файле с использованием потоков».

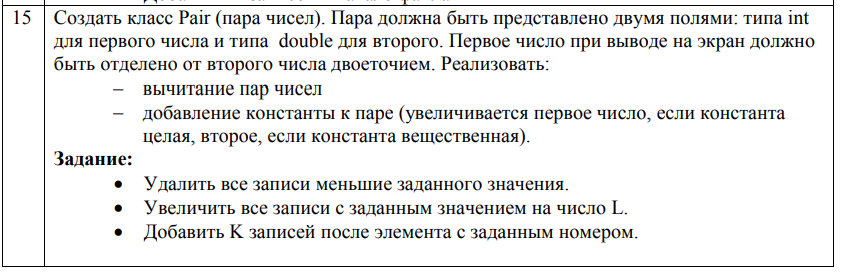
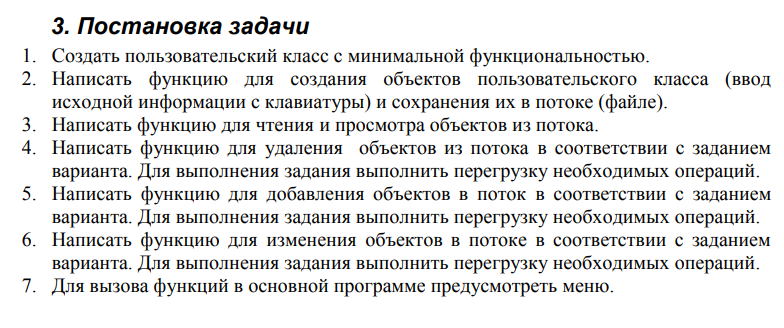
Выполнил: студент группы РИС-23-2б

Вековшинин Иван Николаевич

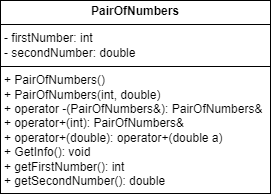
Проверила: доцент кафедры ИТАС

О.А. Полякова.

2024 г.

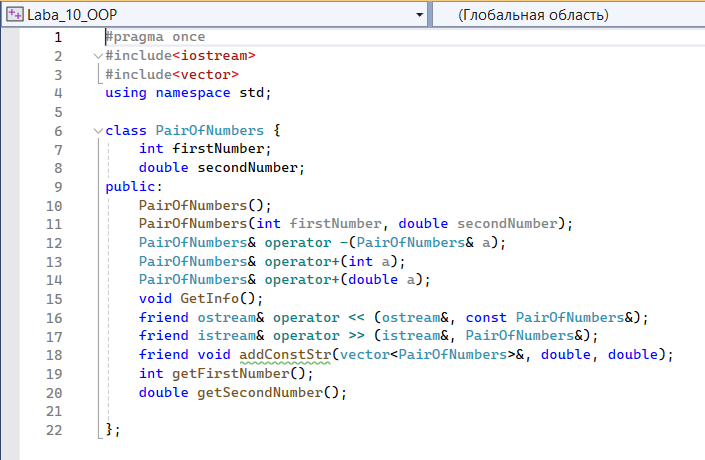


UML Diagram:



Код:

**PairOfNumbers.h**

****

**PairOfNumbers.cpp**

****

**Functions.h**

#pragma once

#include<iostream>

#include<fstream>

#include<string>

#include<stdlib.h>

using namespace std;

void ShowInfo(const char\* path) { //вывод потока на экран

fstream file\_read(path, fstream::in);

string temp;

while (!file\_read.eof()) {

temp = "";

getline(file\_read, temp);

cout << temp << endl;

}

}

void SetInfo(ofstream& write, int k) { //ввод k элементов потока

PairOfNumbers a;

for (int i = 0; i < k; ++i) {

cout << "Введите данные объекта (через пробел): ";

cin >> a;

write << a;

write << '\n';

}

cout << endl;

}

void deleteValue(const char\* path, string value) { //удалить все строки меньше value

fstream temp("temp", fstream::out); //запись файла

fstream file\_read(path, fstream::in); //чтение файла

string tmp;

while (!file\_read.eof()) {

tmp = "";

getline(file\_read, tmp);

if (tmp == "") break;

string num1 = "", num2 = "";

int count = 0;

for (int i = 0; tmp[i] != ' '; ++i) {

num1 += tmp[i]; count += 1;

}

for (int i = count + 3; i < tmp.size(); ++i)

num2 += tmp[i];

if (num1 >= value && num2 >= value) {

temp << tmp << '\n';

}

}

temp.close(); file\_read.close(); //закрыть файлы

remove(path); //удалить файл с названием path

rename("temp", path); //переименовать временный файл в изначальное название

}

void addConst(const char\* path, string key, string value) { //прибавить value к числам равным key

fstream temp("temp", fstream::out);

fstream file\_read(path, fstream::in);

string tmp;

while (!file\_read.eof()) {

tmp = "";

getline(file\_read, tmp);

if (tmp == "") break;

string num1 = "", num2 = "";

int count = 0;

for (int i = 0; tmp[i] != ' '; ++i) {

num1 += tmp[i]; count += 1;

}

for (int i = count + 3; i < tmp.size(); ++i)

num2 += tmp[i];

if (num1 == key || num2 == key) {

temp << atof(num1.c\_str()) + atof(value.c\_str()) << " : " << atof(num2.c\_str()) + atof(value.c\_str()) << '\n';

}

else

temp << num1 << " : " << num2 << '\n';

}

temp.close(); file\_read.close();

remove(path);

rename("temp", path);

}

void addElements(const char\* path, int number, int k) { //добавление k элементов после указанного номера

fstream temp("temp", fstream::out);

fstream file\_read(path, fstream::in);

string tmp;

int count = 0;

while (!file\_read.eof()) {

tmp = "";

getline(file\_read, tmp);

PairOfNumbers a;

if (count == number) {

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

{

cout << "Введите данные объекта (через пробел): ";

cin >> a;

temp << a;

temp << '\n';

}

}

count += 1;

temp << tmp << '\n';

}

temp.close(); file\_read.close();

remove(path);

rename("temp", path);

}

**Main.cpp**

#include<iostream>

#include<fstream>

#include<string>

#include<vector>

#include"PairOfNumbers.h"

#include"functions.h"

using namespace std;

int main() {

system("chcp 1251 > null");

ofstream file\_write;

file\_write.open("File.txt");

cout << "Ввдеите количество объектов: ";

int k; cin >> k;

SetInfo(file\_write, k);

file\_write.close();

k = 0;

while (k != 4) {

system("cls");

ShowInfo("File.txt"); cout << endl;

cout << "Выберите пункт в меню: " << endl;

cout << "1. Удалить все записи меньшие заданного значения\n";

cout << "2. Увеличить все строки с заданным значением на число k\n";

cout << "3. Добавить k записей после элемента с заданным номером\n";

cout << "4. Выход\n";

cin >> k;

string value, Const, AddConst;

int number, Count;

system("cls");

switch (k) {

case 1:

cout << "Введите значение: ";

cin >> value;

deleteValue("File.txt", value);

break;

case 2:

cout << "Введите заданное значение: "; cin >> Const;

cout << "Введите k: "; cin >> AddConst;

addConst("File.txt", Const, AddConst);

case 3:

cout << "Введите k: "; cin >> Count;

cout << "Введите номер элемента: "; cin >> number;

addElements("File.txt", number, Count);

break;

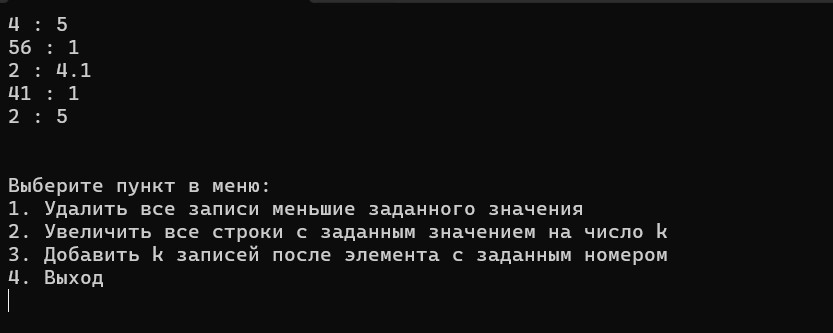
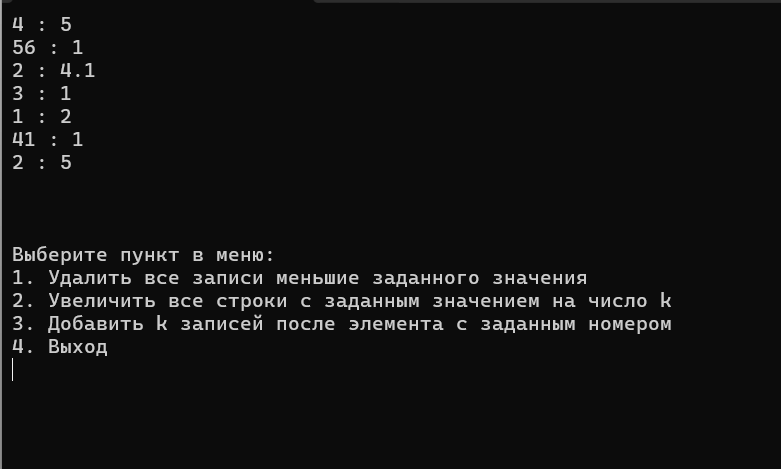
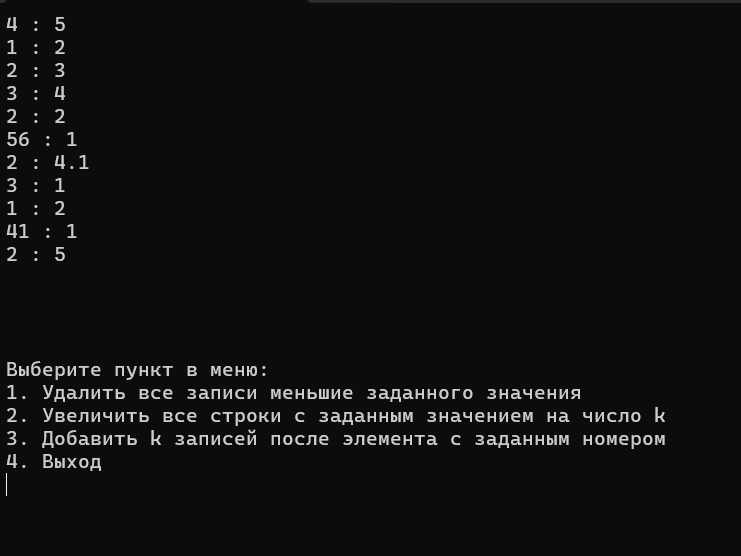
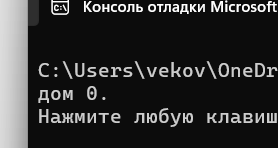
case 4: k = 4; break;

}

}

}

Результат работы программы:

Ответы на вопросы:

1)Поток – последовательность байтов

2)Стандартные: только однонаправленные (входные или выходные)

Строковые: могут быть и однонаправленными, и двунаправленными

Файловые: могут быть и однонаправленными, и двунаправленными

3)библиотека iostream

4)библиотека fstream

5)библиотека sstream

6)осуществляется перегруженной операцией сдвига влево << (вывод)

7)осуществляется перегруженной операцией сдвига вправо >> (ввод)

8)методы: ostream& put (char c) – записывает в потом символ c

ostream& write(const char\* buf, int size) – записывает в потом символ buf, также может записывать строку символов.

9)методы: istream& read(char\* buf, int size) – ввод из потока одного символа

istream& get (char\* str, streamsize count) – ввод из потока строки с пробелами (пока не встречен признак окончания, или не введено count символов)

istream& getline(char\* str, streamsize count) – работает как get, но ввод до переноса на следующую строку.

10)Режимы:

in – чтение; out – запись; trunk – удалить старое содержимое файла; app – открыть поток для записи в конец файла; ate – открыть поток для записи\чтения и встать в конец потока; binary – открыть поток в двоичном режиме.

11)Режим app.

12)режим in – чтение.

13) in | out – запись\чтение.

14) out – запись.

15)поток открывается на запись, и добавление в конец файла.

16)поток открывается на запись и удаление старое содержимого из файла.

17)файла открывается на чтение, запись, и удаляет старое содержимое файла.

18)либо создать объект класса ifstream, либо указать флаг ios::in.

19)либо создать объект класса ofstream, либо указать флаг ios::out.

20)

|  |
| --- |
| fstream file("file.txt", ios::in); //файл на чтение  fstream file("file.txt", ios::out | ios::in); //файл на чтение\запись  fstream file("file.txt", ios::out | ios::app); //файл на запись в конец файла |

21) чтение может осуществляться через объект класса ifstream, или оператором <<.

22)запись может осуществляться через объект класса ofstream или оператором >>.

23)Можно открыть файл с использованием флага trunk и тогда содержимое файла удалиться все содержимое. Или можно создать временный файл записать туда только информацию, которую пользователь хочет оставить, удалить первоначальный файл, и переименовать временный.

24)Чтобы добавить записи в файл, можно открыть файл с использованием флагов out, app и информация добавиться в конец файла.

25)Сначала нужно открыть файл на чтение, считать информацию из потока. После форматировать каким-либо образом этот поток и записать в новый файл измененный потом данных, изначальный файл удалить, новый файл переименовть в название первого файла.