

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**Пермский национальный исследовательский
политехнический университет**

Факультет электротехнический

Кафедра ИТАС

ОТЧЁТ

о лабораторной работе №10 по классам

Выполнил:
Студент группы ИВТ-23-1Б
Пискунов Д. А.

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2024

Задача:

15	<p>Создать класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа <code>int</code> для первого числа и типа <code>double</code> для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием. Реализовать:</p> <ul style="list-style-type: none">– вычитание пар чисел– добавление константы к паре (увеличивается первое число, если константа целая, второе, если константа вещественная). <p>Задание:</p> <ul style="list-style-type: none">• Удалить все записи меньше заданного значения.• Увеличить все записи с заданным значением на число L.• Добавить K записей после элемента с заданным номером.
----	---

Текст программы файл Work.h

```
1  #pragma once
2  #include "Pair.h"
3  #include <iostream>
4
5  char f_name[30];
6
7  int value;
8
9  void check_k(int k) {
10     if (k < 0) {
11         cout << "Невозможно прочесть файл!\n";
12     }
13 }
14
15 int make_file() {
16     cout << "Введите название файла: ";
17     cin >> f_name;
18     fstream stream(f_name, ios::out | ios::trunc);
19     if (!stream) {
20         return -1;
21     }
22     int n;
23     Pair m_Pair;
24     cout << "Введите количество элементов: ";
25     cin >> n;
26     for (int i = 0; i < n; i++) {
27         cin >> m_Pair;
28         stream << m_Pair << '\n';
29     }
30     stream.close();
31     return n;
32 }
33
34 int print_file() {
35     cout << "Введите название файла: ";
36     cin >> f_name;
37     fstream stream(f_name, ios::in);
38     if (!stream) {
39         return -1;
40     }
41     Pair m_Pair;
42     int i = 0;
43     while (stream >> m_Pair) {
44         cout << m_Pair << '\n';
45         i++;
46     }
```

```

46     }
47     if (i == 0) {
48         cout << "Файл пуст!\n";
49     }
50     stream.close();
51     return i;
52 }
53
54 ✓ int del_file() {
55     cout << "Введите название файла: ";
56     cin >> f_name;
57     cout << "Введите значение: ";
58     cin >> value;
59     fstream temp("temp", ios::out);
60     fstream stream(f_name, ios::in);
61     if (!stream) {
62         return -1;
63     }
64     int i = 0;
65     Pair m_Pair;
66     while (stream >> m_Pair) {
67         if (stream.eof()) {
68             break;
69         }
70         i++;
71         if (m_Pair >= value) {
72             temp << m_Pair;
73         }
74     }
75     stream.close();
76     temp.close();
77     remove(f_name);
78     rename("temp", f_name);
79     return i;
80 }
81
82 ✓ int add_file() {
83     int number;
84     cout << "Введите название файла: ";
85     cin >> f_name;
86     cout << "Введите номер: ";
87     cin >> number;
88     cout << "Введите количество записей: ";
89     cin >> value;
90     fstream temp("temp", ios::out);
91     fstream stream(f_name, ios::in);
92     int i = 1;

```

```

92         int i = 1;
93         if (!stream) {
94             return -1;
95         }
96         Pair pair;
97         while (i != number && stream >> pair) {
98             if (stream.eof()) {
99                 break;
100             }
101             i++;
102             temp << pair;
103         }
104         if (i == number) {
105             for (int j = 0; j < value; j++) {
106                 Pair m_Pair;
107                 cin >> m_Pair;
108                 temp << m_Pair;
109             }
110         }
111         while (stream >> pair) {
112             if (stream.eof()) {
113                 break;
114             }
115             temp << pair;
116         }
117         temp.close();
118         stream.close();
119         remove(f_name);
120         rename("temp", f_name);
121         return value;
122     }
123
124     int change_file() {
125         cout << "Введите название файла: ";
126         cin >> f_name;
127         cout << "Введите значение: ";
128         cin >> value;
129         fstream temp("temp", ios::out);
130         fstream stream(f_name, ios::in);
131         if (!stream) {
132             return -1;
133         }
134         Pair pair;

```

```
134         Pair pair;
135         int i = 0;
136         while (stream >> pair) {
137             temp << pair + value;
138         }
139         temp.close();
140         stream.close();
141         remove(f_name);
142         rename("temp", f_name);
143         return i;
144     }
```

Файл Lab_Main.cpp

```
1  #include<iostream>
2  #include <string>
3  #include <functional>
4  #include "file_work.h"
5
6  using namespace std;
7
8  ✓ int main() {
9      system("chcp 1251 > NULL");
10     int choose = 6;
11     function<int()> f[5] = { make_file, print_file, del_file, change_file, add_file };
12     do {
13         if (choose != 6) {
14             check_k(f[choose - 1]());
15         }
16         cout << "Введите номер команды:\n";
17         cout << "1. Создать файл\n";
18         cout << "2. Печать файла\n";
19         cout << "3. Удалить все записи меньше заданного значения\n";
20         cout << "4. Увеличить все записи с заданным значением на число L\n";
21         cout << "5. Добавить K записей перед элементом с заданным номером\n";
22         cout << "6. Выход\n";
23         cin >> choose;
24         system("cls");
25     } while (choose != 6);
26     return 0;
27 }
```

Файл Pair.cpp

```
1  #include "Pair.h"
2
3  Pair::Pair(int first, double second) {
4      this->first = first;
5      this->second = second;
6  }
7  istream& operator>>(istream& in, Pair& m_pair) {
8      cout << "Введите данные в формате:\nцелое число вещественное число\n";
9      in >> m_pair.first;
10     in >> m_pair.second;
11     return in;
12 }
13 ostream& operator<<(ostream& out, const Pair& m_pair) {
14     return (out << m_pair.first << " : " << m_pair.second);
15 }
16 fstream& operator>>(fstream& fin, Pair& m_Pair) {
17     fin >> m_Pair.first;
18     fin >> m_Pair.second;
19     return fin;
20 }
21 fstream& operator<<(fstream& fout, const Pair& m_Pair) {
22     fout << m_Pair.first << ' ' << m_Pair.second << '\n';
23     return fout;
24 }
25 Pair::Pair() {
26     first = 0;
27     second = 0;
28 }
29 Pair::~Pair() {
30 }
31 Pair::Pair(const Pair& m_pair) {
32     first = m_pair.first;
33     second = m_pair.second;
34 }
35 Pair& Pair::operator=(const Pair& m_pair) {
36     if (&m_pair != this) {
37         first = m_pair.first;
38         second = m_pair.second;
39     }
40     return *this;
41 }
42 Pair Pair::operator-(const Pair& m_pair) {
43     Pair result(first - m_pair.first, second - m_pair.second);
44     return result;
45 }
46 }
47 Pair& Pair::operator+(int first) {
48     this->first += first;
49     return *this;
```

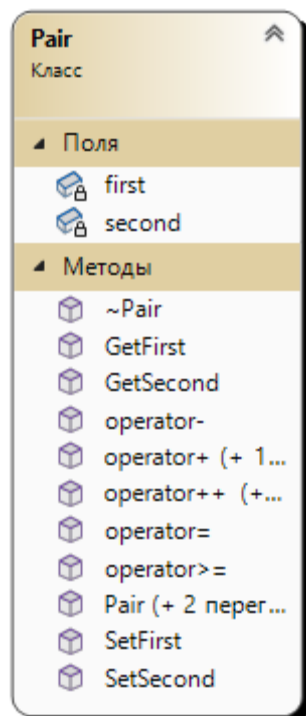


```
49         return *this;
50     }
51     Pair& Pair::operator+(double second) {
52         this->second += second;
53         return *this;
54     }
55     Pair& Pair::operator++() {
56         ++first;
57         ++second;
58         return *this;
59     }
60     Pair Pair::operator ++(int) {
61         Pair temp = *this;
62         this->first++;
63         this->second++;
64         return temp;
65     }
66
67     bool Pair::operator>=(int value) {
68         return (this->first >= value && this->second >= value);
69     }
70
71     int Pair::GetFirst() const {
72         return first;
73     }
74
75     double Pair::GetSecond() const {
76         return second;
77     }
78
79     void Pair::SetFirst(int m_first) {
80         first = m_first;
81     }
82
83     void Pair::SetSecond(double m_second) {
84         second = m_second;
85     }
```

Файл Pair.h

```
1  #pragma once
2  #include <iostream>
3  #include <fstream>
4  using namespace std;
5
6  class Pair {
7      int first;
8      double second;
9  public:
10     friend istream& operator>>(istream& in, Pair& p);
11     friend ostream& operator<<(ostream& out, const Pair& p);
12     Pair(int, double);
13     Pair();
14     Pair(const Pair&);
15     ~Pair();
16     Pair& operator=(const Pair&);
17     Pair operator-(const Pair&);
18     Pair& operator+(int);
19     Pair& operator+(double);
20     Pair& operator++();
21     Pair operator++(int);
22     friend fstream& operator >> (fstream& fin, Pair& m_Pair);
23     friend fstream& operator << (fstream& fout, const Pair& m_Pair);
24     bool operator>=(int value);
25     void SetFirst(int m_first);
26     void SetSecond(double m_second);
27     int GetFirst() const;
28     double GetSecond() const;
29 };
```

UML-диаграмма



Тест

Введите номер команды:

1. Создать файл
2. Печать файла
3. Удалить все записи меньше заданного значения
4. Увеличить все записи с заданным значением на число L
5. Добавить K записей перед элементом с заданным номером
6. Выход

1

Введите название файла: rr

Введите количество элементов: 1

Введите данные в формате:

целое число вещественное число

2

2.4

Введите номер команды:

1. Создать файл
2. Печать файла
3. Удалить все записи меньше заданного значения
4. Увеличить все записи с заданным значением на число L
5. Добавить K записей перед элементом с заданным номером
6. Выход

2

Введите название файла: rr

2 : 2.4

Введите номер команды:

1. Создать файл
2. Печать файла
3. Удалить все записи меньше заданного значения
4. Увеличить все записи с заданным значением на число L
5. Добавить K записей перед элементом с заданным номером
6. Выход

Введите название файла: rr

Введите значение: 1

Введите номер команды:

1. Создать файл
2. Печать файла
3. Удалить все записи меньше заданного значения
4. Увеличить все записи с заданным значением на число L
5. Добавить K записей перед элементом с заданным номером
6. Выход

-

Ответы на вопросы

1. Что такое поток?

Поток - последовательность байтов, не зависящая от конкретного устройства, с которым производится обмен

2. Какие типы потоков существуют?

- Стандартные
- Строковые
- Файловые

Все они делятся на однонаправленные и двунаправленные.

3. Какую библиотеку надо подключить при использовании стандартных потоков?

`iostream`

4. Какую библиотеку надо подключить при использовании файловых потоков?

`fstream`

5. Какую библиотеку надо подключить при использовании строковых потоков?

`sstream`

6. Какая операция используется при выводе в форматированный поток?

`<<`

7. Какая операция используется при вводе из форматированных потоков?

`>>`

8. Какие методы используются при выводе в форматированный поток?

Прототип	Пример	
<code>ostream& put(char c)</code>	<code>char c='a'; stream.put(c);</code>	Записывает в поток stream символ c
<code>ostream& write(const char* buf, int size)</code>	<code>char c='a'; stream.write(&c);</code>	Записывает в поток stream символ c
<code>ostream& write(const char* buf, int size)</code>	<code>char s[]="string1"; stream.write(s,strlen(s));</code>	Записывает в поток stream строку символов