Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение

высшего образования

«**Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные

системы»

**ОТЧЁТ**

**по лабораторной работе №18.10**

Дисциплина: «Информатика»

Тема: “Объектно-ориентированное программирование.

Сохранение данных в файле с использованием потоком.”

Вариант 13

Выполнил:

Студент группы ИВТ-20-2б

Сафронов Владислав Владиславович

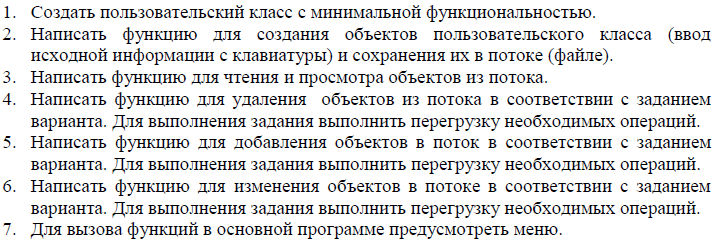
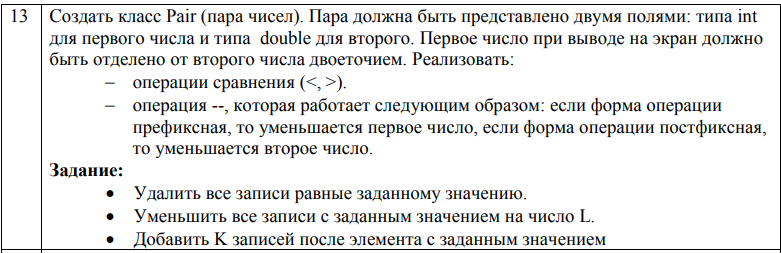
Проверила:

Доцент кафедры ИТАС

Полякова О.А.

**Пермь, 2021**

**Постановка задачи**

****

**Анализ задачи**

1. Для решения задачи необходимо:
   1. Использовать класс Pair из ЛР2.
   2. Реализовать функции для решения задач из варианта.
2. 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
3. Для решения задачи данные были представлены в следующем виде:
   1. Пары чисел записаны в файле Input.txt
4. Для решения задачи данные были представлены в следующем виде:
   1. Для открытия файла на чтение и запись использовать потоки
5. ifstream и ofstream.
6. ifstream input;
7. input.open(file);
8. ofstream output;
9. output.open("temp.txt");
   1. Для вывода пар чисел в консоль используется функция Show()
10. void Show(vector<Pair> a)
11. {
12. for (int i = 0; i < a.size(); i++)
13. {
14. cout << "Pair " << i + 1 << ": " << endl;
15. cout << a[i].GetFirst() << ":" << a[i].GetSecond() <<endl;
16. }
17. }
18. Поставленные задачи будут решены следующими действиями:
    1. Функция удаления всех элементов, равных заданному значению
19. void Delete\_Pairs(char\* file, int f, double s)
20. {
21. ifstream input;
22. input.open(file);
23. ofstream output;
24. output.open("temp.txt");
26. while (!input.eof())
27. {
28. Pair a;
29. input >> a;
30. if (a != Pair())
31. {
32. if (a.GetFirst() != f || a.GetSecond() != s)
33. {
34. output << a;
35. }
36. }
37. }
38. input.close();
39. output.close();
40. remove(file);
41. rename("temp.txt", file);
42. }
    1. Функция для уменьшения всех элементов с заданным значением на число L
43. void Decrease\_Pairs(char\* file, int f, double s, int L)
44. {
45. ifstream input;
46. input.open(file);
47. ofstream output;
48. output.open("temp.txt");
50. while (!input.eof())
51. {
52. Pair a;
53. input >> a;
54. if (a != Pair())
55. {
56. if (a.GetFirst() == f && a.GetSecond() == s)
57. {
58. a.SetFirst(a.GetFirst() - L);
59. a.SetSecond(a.GetSecond() - L);
60. }
61. output << a;
62. }
63. }
64. input.close();
65. output.close();
66. remove(file);
67. rename("temp.txt", file);
68. }
    1. Функция для добавления K записей после элемента с заданным значением.
69. void Add\_Pairs(char\* file, vector<Pair> pairs\_to\_add, int f, double s)
70. {
71. ifstream input;
72. input.open(file);
73. ofstream output;
74. output.open("temp.txt");
76. int flag = 0;
78. while (!input.eof())
79. {
80. Pair a;
81. input >> a;
82. if (a != Pair())
83. {
84. output << a;
85. if (a.GetFirst() == f && a.GetSecond() == s && !flag)
86. {
87. for (int i = 0; i < pairs\_to\_add.size(); i++)
88. {
89. output << pairs\_to\_add[i];
90. }
91. flag = 1;
92. }
93. }
94. }
95. input.close();
96. output.close();
97. remove(file);
98. rename("temp.txt", file);
99. }

**Блок-схема программы**

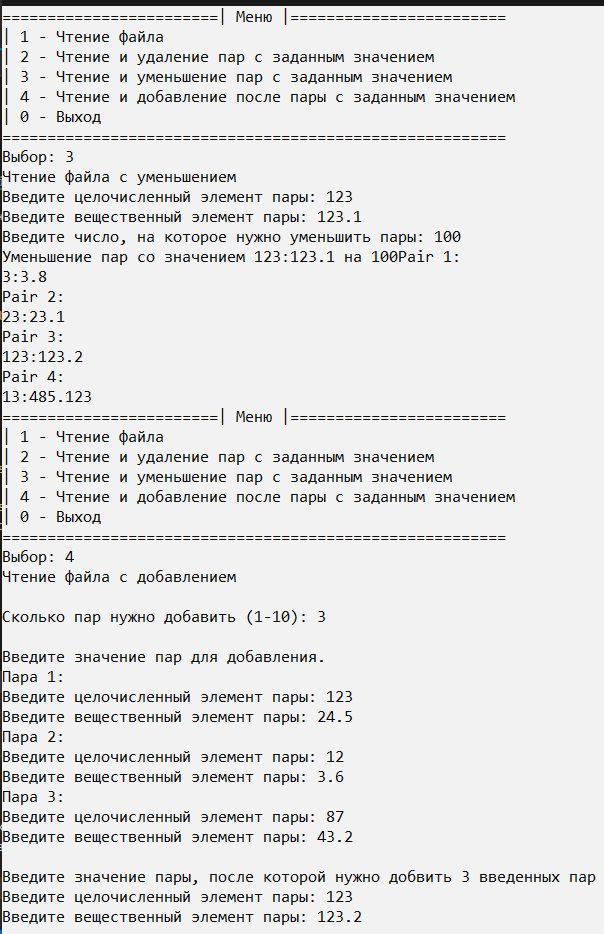
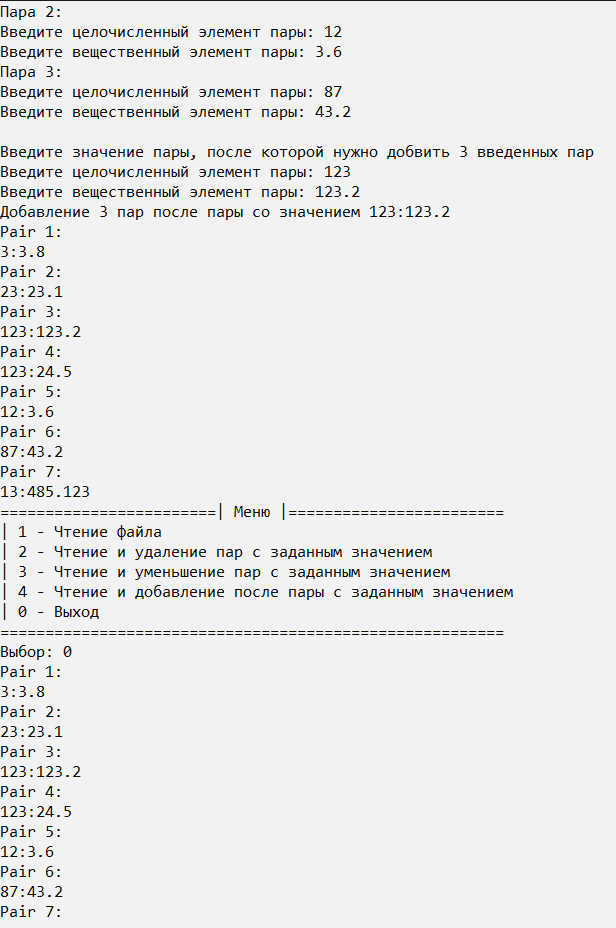
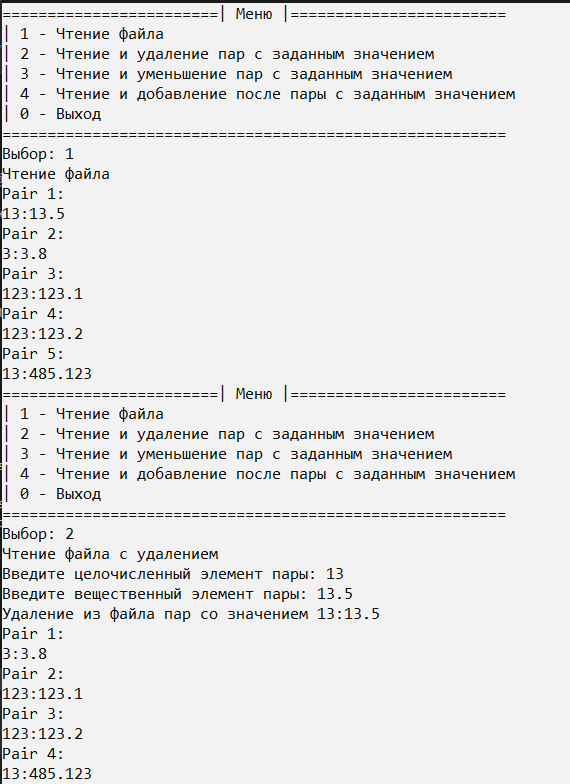
**Код программы**

**Файл Pair.h:**

1. #pragma once
2. #include <iostream>
3. using namespace std;
5. class Pair
6. {
7. int first;
8. double second;
9. public:
10. Pair()
11. {
12. first = 0;
13. second = 0;
14. }
16. Pair(int f, double s)
17. {
18. this->first = f;
19. this->second = s;
20. }
21. Pair(const Pair& obj)
22. {
23. first = obj.first;
24. second = obj.second;
25. }
27. void Show()
28. {
29. cout << first << ":" << second << endl;
30. }
31. void SetFirst(int n)
32. {
33. first = n;
34. }
35. void SetSecond(double n)
36. {
37. second = n;
38. }
39. int GetFirst()
40. {
41. return first;
42. }
43. double GetSecond()
44. {
45. return second;
46. }
48. friend bool operator>(const Pair& obj1, const Pair& obj2);
49. friend bool operator>=(const Pair& obj1, const Pair& obj2);
50. friend bool operator<(const Pair& obj1, const Pair& obj2);
51. friend bool operator<=(const Pair& obj1, const Pair& obj2);
52. friend bool operator==(const Pair& obj1, const Pair& obj2);
53. friend bool operator!=(const Pair& obj1, const Pair& obj2);
55. Pair& operator--() // префикс
56. {
57. first--;
58. return \*this;
59. }
60. Pair operator--(int) //постфикс
61. {
62. second--;
63. return \*this;
64. }
66. friend ostream& operator << (ostream& out, Pair& a)
67. {
68. out << a.first << endl << a.second << endl;
69. return out;
70. }
72. friend istream& operator >> (istream& in, Pair& a)
73. {
74. in >> a.first;
75. in >> a.second;
76. return in;
77. }
79. ~Pair() {};
80. };
82. bool operator>(const Pair& obj1, const Pair& obj2)
83. {
84. return (obj1.first > obj2.first && obj1.second > obj2.second);
85. }
87. bool operator>=(const Pair& obj1, const Pair& obj2)
88. {
89. return (obj1.first >= obj2.first && obj1.second >= obj2.second);
90. }
92. bool operator<(const Pair& obj1, const Pair& obj2)
93. {
94. return (obj1.first < obj2.first&& obj1.second < obj2.second);
95. }
97. bool operator<=(const Pair& obj1, const Pair& obj2)
98. {
99. return (obj1.first <= obj2.first && obj1.second <= obj2.second);
100. }
102. bool operator==(const Pair& obj1, const Pair& obj2)
103. {
104. return (obj1.first == obj2.first && obj1.second == obj2.second);
105. }
107. bool operator!=(const Pair& obj1, const Pair& obj2)
108. {
109. return (obj1.first != obj2.first || obj1.second != obj2.second);
110. }

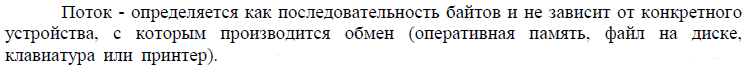
**Файл main.cpp:**

1. #include <iostream>
2. #include <fstream>
3. #include <vector>
4. #include <string>
5. #include "Pair.h"
6. using namespace std;
8. void Delete\_Pairs(char\* file, int f, double s)
9. {
10. ifstream input;
11. input.open(file);
12. ofstream output;
13. output.open("temp.txt");
15. while (!input.eof())
16. {
17. Pair a;
18. input >> a;
19. if (a != Pair())
20. {
21. if (a.GetFirst() != f || a.GetSecond() != s)
22. {
23. output << a;
24. }
25. }
26. }
27. input.close();
28. output.close();
29. remove(file);
30. rename("temp.txt", file);
31. }
32. void Decrease\_Pairs(char\* file, int f, double s, int L)
33. {
34. ifstream input;
35. input.open(file);
36. ofstream output;
37. output.open("temp.txt");
39. while (!input.eof())
40. {
41. Pair a;
42. input >> a;
43. if (a != Pair())
44. {
45. if (a.GetFirst() == f && a.GetSecond() == s)
46. {
47. a.SetFirst(a.GetFirst() - L);
48. a.SetSecond(a.GetSecond() - L);
49. }
50. output << a;
51. }
52. }
53. input.close();
54. output.close();
55. remove(file);
56. rename("temp.txt", file);
57. }
58. void Add\_Pairs(char\* file, vector<Pair> pairs\_to\_add, int f, double s)
59. {
60. ifstream input;
61. input.open(file);
62. ofstream output;
63. output.open("temp.txt");
65. int flag = 0;
67. while (!input.eof())
68. {
69. Pair a;
70. input >> a;
71. if (a != Pair())
72. {
73. output << a;
74. if (a.GetFirst() == f && a.GetSecond() == s && !flag)
75. {
76. for (int i = 0; i < pairs\_to\_add.size(); i++)
77. {
78. output << pairs\_to\_add[i];
79. }
80. flag = 1;
81. }
82. }
83. }
84. input.close();
85. output.close();
86. remove(file);
87. rename("temp.txt", file);
88. }
89. vector<Pair> Read\_to\_Vector(string file)
90. {
91. vector<Pair> tmp;
93. ifstream input;
94. input.open(file);
96. while (!input.eof())
97. {
98. Pair a;
99. input >> a;
100. if (a != Pair())
101. {
102. tmp.push\_back(a);
103. }
104. }
105. input.close();
106. return tmp;
107. }
108. void Show(vector<Pair> a)
109. {
110. for (int i = 0; i < a.size(); i++)
111. {
112. cout << "Pair " << i + 1 << ": " << endl;
113. cout << a[i].GetFirst() << ":" << a[i].GetSecond() << endl;
114. }
115. }
117. void GetMenu()
118. {
119. cout << "========================| Меню |========================\n";
120. cout << "| 1 - Чтение файла\n";
121. cout << "| 2 - Чтение и удаление пар с заданным значением\n";
122. cout << "| 3 - Чтение и уменьшение пар с заданным значением\n";
123. cout << "| 4 - Чтение и добавление после пары с заданным значением\n";
124. cout << "| 0 - Выход\n";
125. cout << "========================================================\n";
126. cout << "Выбор: ";
127. }
129. int main()
130. {
131. setlocale(LC\_ALL, "Russian");
132. char file[] = "input.txt";
133. vector<Pair> pairs;
135. int n = -1;
136. while (n != 0)
137. {
138. GetMenu();
139. cin >> n;
141. switch (n)
142. {
143. case 1: {
144. cout << "Чтение файла\n";
145. pairs = Read\_to\_Vector(file);
146. Show(pairs);
147. break;
148. }
149. case 2: {
150. cout << "Чтение файла с удалением\n";
151. int tmp1; double tmp2;
152. cout << "Введите целочисленный элемент пары: ";
153. cin >> tmp1;
154. cout << "Введите вещественный элемент пары: ";
155. cin >> tmp2;
156. cout << "Удаление из файла пар со значением " << tmp1 << ":" << tmp2 << "\n";
157. Delete\_Pairs(file, tmp1, tmp2);
158. pairs = Read\_to\_Vector(file);
159. Show(pairs);
160. break;
161. }
162. case 3: {
163. cout << "Чтение файла с уменьшением\n";
164. int tmp1, L; double tmp2;
165. cout << "Введите целочисленный элемент пары: ";
166. cin >> tmp1;
167. cout << "Введите вещественный элемент пары: ";
168. cin >> tmp2;
169. cout << "Введите число, на которое нужно уменьшить пары: ";
170. cin >> L;
171. cout << "Уменьшение пар со значением " << tmp1 << ":" << tmp2 << " на " << L;
172. Decrease\_Pairs(file, tmp1, tmp2, L);
173. pairs = Read\_to\_Vector(file);
174. Show(pairs);
175. break;
176. }
177. case 4: {
178. cout << "Чтение файла с добавлением\n";
179. vector<Pair> tmp;
180. int tmp1, k; double tmp2;
181. cout << "\nСколько пар нужно добавить (1-10): ";
182. cin >> k;
183. while (k < 1 || k > 10)
184. {
185. cout << "Ввод некорректен! Можно добавить от 1 до 10 пар.\n";
186. cout << "Сколько пар нужно добавить (1-10) ";
187. cin >> k;
188. }
189. cout << "\nВведите значение пар для добавления.\n";
190. for (int i = 0; i < k; i++)
191. {
192. cout << "Пара " << i + 1 << ":\n";
193. cout << "Введите целочисленный элемент пары: ";
194. cin >> tmp1;
195. cout << "Введите вещественный элемент пары: ";
196. cin >> tmp2;
197. tmp.push\_back(Pair(tmp1, tmp2));
198. }
199. cout << "\nВведите значение пары, после которой нужно добвить " << k << " введенных пар\n";
200. cout << "Введите целочисленный элемент пары: ";
201. cin >> tmp1;
202. cout << "Введите вещественный элемент пары: ";
203. cin >> tmp2;
204. cout << "Добавление " << k << " пар после пары со значением " << tmp1 << ":" << tmp2 << endl;
205. Add\_Pairs(file, tmp, tmp1, tmp2);
206. pairs = Read\_to\_Vector(file);
207. Show(pairs);
208. break;
209. }
210. }
211. }
212. Show(pairs);
214. return 0;
215. }

**Скриншоты результатов работы программы**

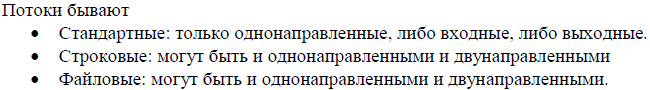
**Ответы на вопросы**



1. 



2.





3. 



4. 



5. 

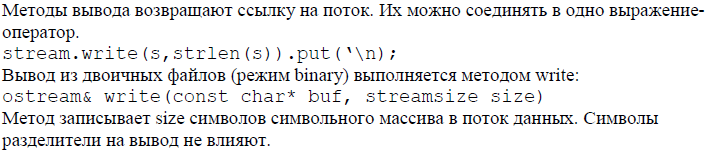


6. 



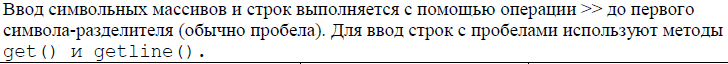
7. operator >>

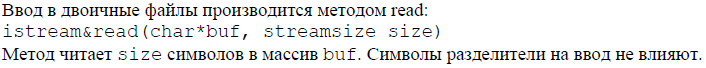


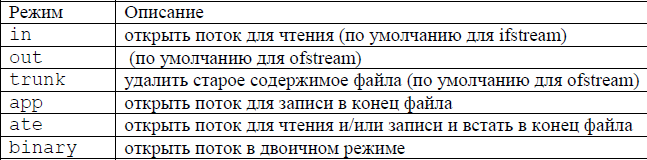
8. 

25



9.



10. 



11. 

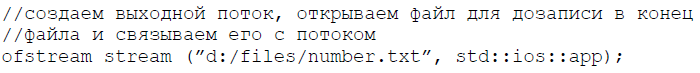


12. 



13. fstream file("f.txt", ios\_base::in | ios\_base::out)



14. 



15. 

26



16. 



17. 

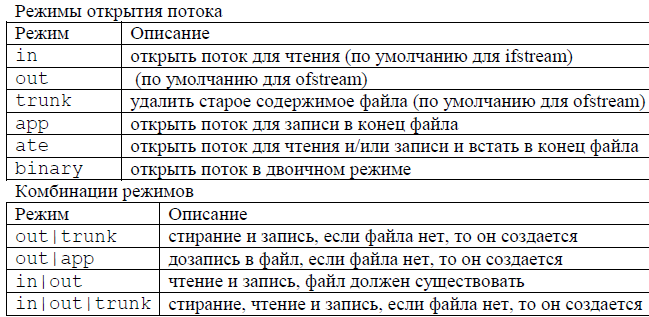


18. 



19. 



20. 



21. Перегрузка оператора в классе.



22. 



23. Перенесение в файл элементов, удовлетворяющих условию.

27



24. Запись в файл из массива элементов между записью элементов из входного файла.



25. Перезаписывание всех элементов в выходной файл с исправлением тех, что удовлетворяют условию.