МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Белгородский государственный технологический университет

им. В.Г. Шухова»

**ИНСТИТУТ ЭНЕРГЕТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**И УПРАВЛЯЮЩИХ СИСТЕМ**

**Кафедра программного обеспечения вычислительной**

**техники и автоматизированных систем**

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине: Объектно-ориентированное программирование

на тему:

Модульное программирование. Интерфейсы.

Работу выполнил:

студент группы ВТ-22

Воскобойников И. С.

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Буханов Д.Г.

Белгород 2020

**Цель работы**: Получение навыков модульной декомпозиции предметной области, создания модулей. Разработка интерфейсов.

**Задание**: Разработать программу, состоящую из трех модулей в соответствии с указанным вариантом задания. Первый модуль – основной код программы; второй содержит интерфейсы; третий модуль – реализацию этих интерфейсов. Количество структур данных ("объектов") не менее пяти.

Вариант 3



Main.сpp

1. #include <iostream>
2. #include "interface.h"
3. #include <stdio.h>
4. #include <string.h>
5. #include <stdlib.h>
6. #include <iostream>
7. #include <windows.h>
8. #include <locale.h>
9. using namespace std;
10. int main()
11. {
12. SetConsoleCP(1251);
13. SetConsoleOutputCP(1251);
14. setlocale(LC\_ALL, "Russian");
15. play r;
16. r.run();
17. }

interface.h

1. #ifndef INTERFACE\_H
2. #define INTERFASE\_H
3. #include <fstream>
4. #include <string>
5. #include <iostream>
6. const int N = 30;
7. struct info\_film {
8. char\* adress;
9. std::string name;
10. FILE\* f;
11. int time;
12. int year;
13. void set\_name(std::string );
14. void set\_time(int);
15. void set\_year(int);
16. void set\_adress(char \*);
17. void print\_name();
18. void print\_year();
19. void print\_time();
20. void print\_all();
21. void play\_film();
22. void off\_film();
23. void creatе\_film();
24. };
25. struct film\_collection {
26. film\_collection();
27. void add\_film (info\_film);
28. void dell\_film(std::string s);
29. int search\_film(std::string s);
30. void output();
31. info\_film film[N];
32. int n;
33. };
34. struct sort\_film {
35. film\_collection film;
36. void name\_sort();
37. void year\_sort();
38. };
39. struct unity\_film {
40. void year\_unity(info\_film t[], int n);
41. };
42. struct play {
43. film\_collection film;
44. int run\_film(std::string s);
45. void run();
46. };
47. #endif

interface.сpp

1. #include "interface.h"
2. #include <iostream>
3. #include <string>
4. #include <fstream>
5. #include <stdlib.h>
6. #include <locale.h>
7. using namespace std;
8. info\_film::info\_film() {
9. name = "";
10. time = 0;
11. year = 0;
12. }
13. void info\_film::set\_name(std::string s)
14. {
15. name = s;
16. }
17. void info\_film::set\_time(int n)
18. {
19. time = n;
20. }
21. void info\_film::set\_year(int n)
22. {
23. year = n;
24. }
25. void info\_film::print\_name()
26. {
27. std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;
28. std::cout << "Название фильма: " << name << endl;
29. }
30. void info\_film::print\_time()
31. {
32. std::cout << "Продолжительность фильма: " << time << " min"<< endl;
33. }
34. void info\_film::print\_year()
35. {
36. std::cout << "Год выпуска: " << year << endl;
37. }
38. void info\_film::print\_all()
39. {
40. print\_name();
41. cout << " ";
42. print\_year();
43. cout << " ";
44. print\_time();
45. cout << " ";
46. }
47. void info\_film::set\_adress(char\* s)
48. {
49. adress = s;
50. }
51. void info\_film::play\_film()
52. {
53. cout << "======================= Фильм =======================" << endl;
54. }
55. void info\_film::off\_film()
56. {
57. system("CLS");
58. }
59. int play::run\_film(std::string name)
60. {
61. int t = film.search\_film(name);
62. cout << t;
63. if (t != -1)
64. film.film[t].play\_film();
65. else
66. cout << "Фильм не найден!" << endl;
67. return t;
68. }
69. film\_collection::film\_collection()
70. {
71. n = 3;
72. char\* s = (char\*)"D:\\movies\\1.mp4";
73. char\* s1 = (char\*)"D:\\movies\\2.mp4";
74. char\* s2 = (char\*)"D:\\movies\\3.mp4";
75. film[0].set\_name("sherlock homs");
76. film[0].set\_adress(s);
77. film[0].set\_year(1975);
78. film[0].set\_time(320);
79. film[1].set\_name("titanic");
80. film[1].set\_adress(s1);
81. film[1].set\_year(1995);
82. film[1].set\_time(250);
83. film[2].set\_name("joker");
84. film[2].set\_adress(s2);
85. film[2].set\_year(2019);
86. film[2].set\_time(120);
87. }
88. void film\_collection::add\_film(info\_film t)
89. {
90. if (search\_film(t.name) == -1)
91. {
92. film[n] = t;
93. n++;
94. }
95. else
96. std::cout << "фильм уже добавлен!" << endl;
97. }
98. void film\_collection::dell\_film(std::string s)
99. {
100. int k = search\_film(s);
101. if (k != -1)
102. {
103. film[k] = film[n - 1];
104. --n;
105. }
106. else
107. std::cout << "фильм не найден!" << endl;
108. }
109. int film\_collection::search\_film(std::string s)
110. {
111. int i = 0;
112. while (i < n)
113. {
114. if (film[i].name == s)
115. {
116. return i;
117. }
118. i++;
119. }
120. return -1;
121. }
122. void info\_film::creatе\_film()
123. {
124. std::string s;
125. int n;
126. float n1;
127. cout << "Введите название : ";
128. cin >> s;
129. set\_name(s);
130. cout << "Введите год выпуска: ";
131. cin >> n;
132. set\_year(n);
133. cout << "ВВедите продолжительность: ";
134. cin >> n1;
135. set\_time(n1);
136. }
137. void sort\_film::name\_sort()
138. {
139. for (int i = 0; i < film.n - 1; i++)
140. for (int j = i + 1; j < film.n; j++)
141. {
142. if (film.film[i].name > film.film[j].name)
143. std::swap(film.film[i], film.film[j]);
144. }
145. }
146. void sort\_film::year\_sort()
147. {
148. for (int i = 0; i < film.n - 1; i++)
149. for (int j = i + 1; j < film.n; j++)
150. {
151. if (film.film[i].year > film.film[j].year)
152. std::swap(film.film[i], film.film[j]);
153. }
154. }
155. void unity\_film::year\_unity(info\_film film[], int n)
156. {
157. int f = 0, k = 1, i = 0, j = 1;
158. std::cout << "-------------- 1 папка -------------- " << endl;
159. while (i <= n - 1 && j <= n)
160. {
161. if (film[i].year == film[j].year)
162. {
163. if (!f) film[i].print\_all();
164. f = 1;
165. film[j].print\_all();
166. j++;
167. continue;
168. }
169. else
170. {
171. if (f)
172. {
173. k++;
174. std::cout << "-------------- " << k << " папка -------------- " << endl;
175. }
176. f = 0;
177. i = j;
178. j = i + 1;
179. }
180. }
181. i = 0; j = 1;
182. while (i <= n - 1 && j <= n)
183. {
184. if (film[i].year == film[j].year)
185. {
186. i += 2;
187. j += 2;
188. }
189. else
190. {
191. film[i].print\_all();
192. i++; j++;
193. }
194. }
195. }
196. void film\_collection::output()
197. {
198. for (int i = 0; i < n; i++)
199. {
200. film[i].print\_all();
201. }
202. }
203. void play::run()
204. {
205. int n = 0, m, v, f = 0;
206. std::string s, s1, y;
207. sort\_film c;
208. while (n != 6)
209. {
210. if (!f) c.film.output();
211. f = 0;
212. cout << "Включить фильм - 1" << endl;
213. cout << "Добавить фильм в фильмотеку - 2" << endl;
214. cout << "Удалить фильм из фильмотеки - 3" << endl;
215. cout << "Отсортировать фильмотеку - 4" << endl;
216. cout << "Объеденить фильмы по папкам - 5" << endl;
217. cout << "Выйти из домашней фильмотеки - 6" << endl;
218. cout << "Ваш выбор: ";
219. cin >> n;
220. switch (n)
221. {
222. case 1:
223. {
224. cout << "Какой фильм включить? " << endl;
225. cin >> s;
226. cout << endl;
227. int t = run\_film(s);
228. if (t != -1) {
229. cout << endl;
230. cout << "Выключить фильм? - 1" << endl;
231. c.film.film[t].off\_film(); break;
232. }
233. break;
234. }
235. case 2:
236. {
237. info\_film t;
238. t.creatе\_film();
239. c.film.add\_film(t);
240. break;
241. }
242. case 3:
243. {
244. cout << "Какой фильм удалить? " << endl;
245. cin >> s1;
246. c.film.dell\_film(s1);
247. break;
248. }
249. case 4:
250. {
251. cout << "Отсортировать по названию фильма - 1 " << endl;
252. cout << "Отсортировать по году создания - 2 " << endl;
253. cin >> m;
254. switch (m)
255. {
256. case 1:
257. {
258. c.name\_sort();
259. break;
260. }
261. case 2:
262. {
263. c.year\_sort();
264. break;
265. break;
266. }
267. }
268. break;
269. }
270. case 5:
271. {
272. cout << "Объеденить фильмы в папку по году их создания - 1" << endl;
273. cin >> m;
274. c.year\_sort();
275. unity\_film u;
276. u.year\_unity(c.film.film, c.film.n);
277. f = 1;
278. }
279. }
280. }
281. }

