## C语言实现维护多个课程数据

1. /**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
2. \*    名    称：学生成绩管理系统1.0                         \*
3. \*    创建日期：2025-4-18                                 \*
4. \*    最后修改：2025-4-18                                \*
5. \*    版    权：中国农业大学信息与电气工程学院           \*
6. \*    版 本 号：1.0                                      \*
7. \*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****\*/*
8. #include <iostream>
9. #include <cstring>
10. #include <Windows.h>
11. using namespace std;
12. #define N 20   //数组的长度
13. #define COURSEMAX 3 // 课程数量
14. enum class Course : int{Chinese, Math, English};
15. void clearScreen();
16. /**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
17. 功能:显示菜单
18. 参数：无
19. 返回值：无
20. \*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*/
21. void displayMenu()
22. {
24. cout<<"++++++++++++++++++++++++++++++++++++++++++++++++++++\n";
25. cout<<"+                 学生成绩管理系统                 +\n";
26. cout<<"+                                                  +\n";
27. cout<<"+                    1. 录入                       +\n";
28. cout<<"+                    2. 输出                       +\n";
29. cout<<"+                    3. 查询                       +\n";
30. cout<<"+                    4. 排序                       +\n";
31. cout<<"+                    5. 切换课程                   +\n";
32. cout<<"+                    6. 结束                       +\n";
33. cout<<"+                                                  +\n";
34. cout<<"++++++++++++++++++++++++++++++++++++++++++++++++++++\n";
36. }
37. /**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
38. 功能:登陆函数，验证口令
39. 参数：
40. 类型：char []
41. 说明：用户输入口令
42. 返回值：1或0
43. 类型：int
44. 说明：1表示通过验证；0表示没有通过
45. \*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****\*/*
46. int login()
47. {
48. char password[8];/\*用户输入的口令*\*/*
49. static char key[8]="1234";
50. int count = 0;
51. int s=0;
53. /\*====验证用户的口令，如果三次输入均错误，则不允许进入系统====*\*/*
54. do
55. {
56. count++;
57. cout<<"请输入口令：";
58. gets(password);
59. if(strcmp(password,key)==0)
60. s = 1;
61. else
62. s = 0;
63. }while(s==0 && count<3);
64. clearScreen();
65. cout<<" Reloading...\n";
66. cout<<" 系统登录成功! \n";
67. return s;
68. }
69. /**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
70. 功能:输入学生成绩到数组中
71. 参数：
72. 参数1：score[]
73. 类型：int []
74. 说明：学生成绩数组
75. 参数2：length
76. 类型：int
77. 说明：要输入的学生成绩数组的长度
78. 返回值：i
79. 类型：int
80. 说明：实际成绩数组的有效值个数
81. \*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*/
82. int inputScore(int score[],int length)
83. {
84. int i=0;
85. int tmp;
87. cout<<"请输入学生的成绩（以-1结束输入):\n";
88. do
89. {
90. cout<<"第"<<i+1<<"个学生：";
91. cin>>tmp;
92. if(tmp>=0&&tmp<=100)
93. {
94. score[i]=tmp;
95. i++;
96. }
97. }while(tmp!=-1 && i<length);
98. clearScreen();
99. cout<<" 信息录入成功，本次录入"<<i<<"条数据! "<<endl;
100. return i;
101. }
102. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
103. 功能:输出数组中的学生成绩
104. 参数：
105. 参数1：score[]
106. 类型：int []
107. 说明：学生成绩数组
108. 参数2：length
109. 类型：int
110. 说明：要输出的成绩数组的长度
111. 返回值：无
112. 类型：void
113. 说明：
114. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
115. void outputScore(int score[],int length)
116. {
117. if(length <=0){
118. cout<<" 信息为空! \n";
119. return ;
120. }
121. int i=0;
122. cout<<"学生成绩如下：\n";
123. for(i=0;i<length;i++)
124. cout<<i+1<<"\t"<<score[i]<<endl;
125. cout<<"\n";
126. return ;
127. }
128. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
129. 功能:查找某个学生成绩是否在数组中，如果找到，则输出下标，
130. 否则，显示没有找到。
131. 参数：
132. 参数1：score[]
133. 类型：int []
134. 说明：学生成绩数组
135. 参数2：length
136. 类型：int
137. 说明：要输入的学生成绩数组的长度
138. 参数2：xScore
139. 类型：int
140. 说明：要查找的学生成绩
141. 返回值：k
142. 类型：int
143. 说明：
144. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
145. int queryScore(int score[],int length,int xScore)
146. {
147. int i;
149. for(i=0;i<length;i++)
150. {
151. if(xScore==score[i])
152. {
153. return i;
154. }
155. }
157. return -1;
158. }
159. /**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
160. 功能:排序学生的成绩，按照从小到大顺序排序。
161. 参数：
162. 参数1：score[]
163. 类型：int []
164. 说明：学生成绩数组
165. 参数2：length
166. 类型：int
167. 说明：要输入的学生成绩数组的长度
168. 返回值：无
169. 类型：void
170. 说明：
171. \*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****\*/*
172. void sortScore(int score[ ], int length)
173. {
174. int i, j, t;
175. clearScreen();
176. cout<<"成绩排序如下：\n";
177. for(i = 0; i < length - 1; i++)
178. {
179. for(j = 0; j < length - i - 1; j++)
180. if(score[ j ]>score[ j + 1 ])
181. {
182. t = score[ j ];
183. score[ j ] = score[ j + 1 ];
184. score[ j + 1 ] = t;
185. }
186. }
187. }
188. Course switchCourse(){
189. cout << "请输入课程编号（0=Chinese, 1=Math, 2=English）: ";
190. int input;
191. while(1){
192. cin>>input;
193. if (input >= 0 && input <= 2) {
194. Course course = static\_cast<Course>(input);
195. switch (course) {
196. clearScreen();
197. case Course::Chinese:
198. std::cout << " 已切换到课程：语文" << std::endl;
199. break;
200. case Course::Math:
201. std::cout << " 已切换到课程：数学" << std::endl;
202. break;
203. case Course::English:
204. std::cout << " 已切换到课程：英语" << std::endl;
205. break;
206. }
207. break;
208. } else {
209. std::cerr << " 输入无效！请重新输入：";
210. }
211. }
213. return static\_cast<Course>(input);
214. }
215. void clearScreen() {
216. #ifdef \_WIN32
217. system("cls");   // Windows
218. #else
219. system("clear"); // Linux/macOS
220. #endif
221. }
222. /**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
223. 功能:主函数
224. 参数：无
225. 返回值：无
226. \*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*/
227. int main()
228. {
229. int choice = 0; /\*代表用户选择的操作数字 \*/
230. int s=0,datalen=0,x,find;
231. // 课程成绩数组
232. int \*course[COURSEMAX];
233. for (int i = 0; i < COURSEMAX; i++) {
234. course[i] = (int \*)malloc(sizeof(int)\*N);  // 动态分配内存
235. }
237. /\*====验证用户的口令====\*/
238. if (login( ) == 0)
239. {
240. cout<<" 口令错误，不允许进入本系统!\n";
241. exit( 0 );
242. }
244. /\*====根据用户的选择，执行相应的操作.====\*/
245. Course input = Course::Chinese;
246. while(1)
247. {
248. displayMenu();
249. cout << "\n 当前课程: ";
250. switch (input) {
251. case Course::Chinese:
252. std::cout << "语文" << std::endl;
253. break;
254. case Course::Math:
255. std::cout << "数学" << std::endl;
256. break;
257. case Course::English:
258. std::cout << "英语" << std::endl;
259. break;
260. }
262. cout<<"\n 请选择您的操作(1,2,3,4,5,6)：\n";
263. cin>>choice;
264. int i = static\_cast<int>(input);
265. switch(choice)
266. {
267. case 1:
268. datalen=inputScore(course[i],N);
269. break;
270. case 2:
271. outputScore(course[i],datalen);
272. break;
273. case 3:
274. cout<<"\n 请输入要查找的成绩：";
275. cin>>x;
276. find=queryScore(course[i],datalen,x);
277. if(find>=0)
278. cout<<"要查成绩在数组中的下标为:"<<find<<endl;
279. else
280. cout<<"没有要查找的成绩:\n";
281. break;
282. case 4:
283. sortScore(course[i],datalen);
284. outputScore(course[i],datalen);
285. break;
286. case 5:
287. input = switchCourse();
288. break;
289. case 6:
290. cout<<" 正在退出成绩管理系统,请稍后..."<<endl;
291. Sleep(500);
293. // 释放内存
294. for (int i = 0; i < COURSEMAX; i++) {
295. free(course[i]);  // 动态分配内存
296. }
297. exit(0);
298. default:
299. printf(" 选择错误，请重新选择!\n");
300. }
301. }
302. return 0;
303. }

## C++实现类封装成绩管理系统

1. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
2. \*    名    称：学生成绩管理系统1.0                      \*
3. \*    创建日期：2025-4-18                                \*
4. \*    最后修改：2025-4-18                                \*
5. \*    版    权：中国农业大学信息与电气工程学院           \*
6. \*    版 本 号：1.0             \*
7. \*               Developed By Kevin                           \*
8. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
9. #include <iostream>
10. #include <cstring>
11. #include <Windows.h>
12. using namespace std;
13. class Score;
14. #define N 20   *//数组的长度*
15. #define COURSEMAX 3 *// 课程数量*
16. enum class Course : int{Chinese, Math, English};
17. class Score{
18. private:
19. int score[N];
20. int dataSize;
21. *// Course course;*
22. public:
23. Score():dataSize(0){}
24. int inputScore();*// 成绩输入*
25. void outputScore();*// 成绩输出*
26. int queryScore(int xScore); *// 成绩查询*
27. void sortScore(); *// 成绩排序*
29. *//静态成员函数*
30. static Course switchCourse(); *// 课程切换*
31. static int login(); *// 登录*
32. static void displayMenu(); *// 主菜单*
33. static void clearScreen(); *// 清屏函数*
34. };
35. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
36. 功能:显示菜单
37. 参数：无
38. 返回值：无
39. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
40. void Score::displayMenu()
41. {
43. cout<<"++++++++++++++++++++++++++++++++++++++++++++++++++++\n";
44. cout<<"+                 学生成绩管理系统                 +\n";
45. cout<<"+                                                  +\n";
46. cout<<"+                    1. 录入                       +\n";
47. cout<<"+                    2. 输出                       +\n";
48. cout<<"+                    3. 查询                       +\n";
49. cout<<"+                    4. 排序                       +\n";
50. cout<<"+                    5. 切换课程                   +\n";
51. cout<<"+                    6. 结束                       +\n";
52. cout<<"+                                                  +\n";
53. cout<<"++++++++++++++++++++++++++++++++++++++++++++++++++++\n";
55. }
56. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
57. 功能:登陆函数，验证口令
58. 参数：
59. 类型：char []
60. 说明：用户输入口令
61. 返回值：1或0
62. 类型：int
63. 说明：1表示通过验证；0表示没有通过
64. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
65. int Score::login()
66. {
67. char password[8];*/\*用户输入的口令\*/*
68. static char key[8]="1234";
69. int count = 0;
70. int s=0;
72. */\*====验证用户的口令，如果三次输入均错误，则不允许进入系统====\*/*
73. do
74. {
75. count++;
76. cout<<"请输入口令：";
77. gets(password);
78. if(strcmp(password,key)==0)
79. s = 1;
80. else
81. s = 0;
82. }while(s==0 && count<3);
83. clearScreen();
84. cout<<" Reloading...\n";
85. Sleep(500);
86. cout<<" 系统登录成功! \n";
87. return s;
88. }
89. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
90. 功能:输入学生成绩到数组中
91. 参数：
92. 参数1：score[]
93. 类型：int []
94. 说明：学生成绩数组
95. 参数2：length
96. 类型：int
97. 说明：要输入的学生成绩数组的长度
98. 返回值：i
99. 类型：int
100. 说明：实际成绩数组的有效值个数
101. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
102. int Score::inputScore()
103. {
104. int i=0;
105. int tmp;
107. cout<<"请输入学生的成绩（以-1结束输入):\n";
108. do
109. {
110. cout<<"第"<<i+1<<"个学生：";
111. cin>>tmp;
112. if(tmp>=0&&tmp<=100)
113. {
114. score[i]=tmp;
115. i++;
116. }
117. }while(tmp!=-1 && i<N);
118. clearScreen();
119. cout<<" 信息录入成功，本次录入"<<i<<"条数据! "<<endl;
120. dataSize = i;
121. return dataSize;
122. }
123. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
124. 功能:输出数组中的学生成绩
125. 参数：
126. 参数1：score[]
127. 类型：int []
128. 说明：学生成绩数组
129. 参数2：length
130. 类型：int
131. 说明：要输出的成绩数组的长度
132. 返回值：无
133. 类型：void
134. 说明：
135. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
136. void Score::outputScore()
137. {
138. if(dataSize <=0){
139. cout<<" 信息为空! \n";
140. return ;
141. }
142. int i=0;
143. cout<<"学生成绩如下：\n";
144. for(i=0;i<dataSize;i++)
145. cout<<i+1<<"\t"<<score[i]<<endl;
146. cout<<"\n";
147. return ;
148. }
149. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
150. 功能:查找某个学生成绩是否在数组中，如果找到，则输出下标，
151. 否则，显示没有找到。
152. 参数：
153. 参数1：score[]
154. 类型：int []
155. 说明：学生成绩数组
156. 参数2：length
157. 类型：int
158. 说明：要输入的学生成绩数组的长度
159. 参数2：xScore
160. 类型：int
161. 说明：要查找的学生成绩
162. 返回值：k
163. 类型：int
164. 说明：
165. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
166. int Score::queryScore(int xScore)
167. {
168. int i;
170. for(i=0;i<dataSize;i++)
171. {
172. if(xScore==score[i])
173. {
174. return i;
175. }
176. }
177. return -1;
178. }
179. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
180. 功能:排序学生的成绩，按照从小到大顺序排序。
181. 参数：
182. 参数1：score[]
183. 类型：int []
184. 说明：学生成绩数组
185. 参数2：length
186. 类型：int
187. 说明：要输入的学生成绩数组的长度
188. 返回值：无
189. 类型：void
190. 说明：
191. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
192. void Score::sortScore()
193. {
194. int i, j, t;
195. clearScreen();
196. cout<<"成绩排序如下：\n";
197. for(i = 0; i < dataSize - 1; i++)
198. {
199. for(j = 0; j < dataSize - i - 1; j++)
200. if(score[ j ]>score[ j + 1 ])
201. {
202. t = score[ j ];
203. score[ j ] = score[ j + 1 ];
204. score[ j + 1 ] = t;
205. }
206. }
207. }
208. Course Score::switchCourse(){
209. cout << "请输入课程编号（0=Chinese, 1=Math, 2=English）: ";
210. int input;
211. while(1){
212. cin>>input;
213. if (input >= 0 && input <= 2) {
214. Course course = static\_cast<Course>(input);
215. clearScreen();
216. switch (course) {
217. case Course::Chinese:
218. std::cout << " 已切换到课程：语文" << std::endl;
219. break;
220. case Course::Math:
221. std::cout << " 已切换到课程：数学" << std::endl;
222. break;
223. case Course::English:
224. std::cout << " 已切换到课程：英语" << std::endl;
225. break;
226. }
227. break;
228. } else {
229. std::cerr << " 输入无效！请重新输入：";
230. }
231. }
233. return static\_cast<Course>(input);
234. }
235. void Score::clearScreen() {
236. #ifdef \_WIN32
237. system("cls");   *// Windows*
238. #else
239. system("clear"); *// Linux/macOS*
240. #endif
241. }
242. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
243. 功能:主函数
244. 参数：无
245. 返回值：无
246. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
247. int main()
248. {
249. int choice = 0; */\*代表用户选择的操作数字 \*/*
250. int s=0,datalen=0,x,find;
251. *// 课程成绩数组*
252. Score course[COURSEMAX];
254. */\*====验证用户的口令====\*/*
255. if (Score::login( ) == 0)
256. {
257. cout<<" 口令错误，不允许进入本系统!\n";
258. exit( 0 );
259. }
261. */\*====根据用户的选择，执行相应的操作.====\*/*
262. Course input = Course::Chinese;
263. while(1)
264. {
265. Score::displayMenu();
266. cout << "\n 当前课程: ";
267. switch (input) {
268. case Course::Chinese:
269. std::cout << "语文" << std::endl;
270. break;
271. case Course::Math:
272. std::cout << "数学" << std::endl;
273. break;
274. case Course::English:
275. std::cout << "英语" << std::endl;
276. break;
277. }
279. cout<<"\n 请选择您的操作(1,2,3,4,5,6)：\n";
280. cin>>choice;
281. int i = static\_cast<int>(input);
282. switch(choice)
283. {
284. case 1:
285. datalen=course[i].inputScore();
286. break;
287. case 2:
288. Score::clearScreen();
289. course[i].outputScore();
290. break;
291. case 3:
292. cout<<"\n 请输入要查找的成绩：";
293. cin>>x;
294. find=course[i].queryScore(x);
295. if(find>=0)
296. cout<<"要查成绩在数组中的下标为:"<<find<<endl;
297. else
298. cout<<"没有要查找的成绩:\n";
299. break;
300. case 4:
301. course[i].sortScore();
302. course[i].outputScore();
303. break;
304. case 5:
305. input = Score::switchCourse();
306. break;
307. case 6:
308. cout<<" 正在退出成绩管理系统,请稍后..."<<endl;
309. Sleep(500);
311. exit(0);
312. default:
313. printf(" 选择错误，请重新选择!\n");
314. }
315. }
316. return 0;
317. }