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. #include <vector>
13. #include <algorithm>
14. using namespace std;
15. class Score;
16. #define N 20   *//数组的长度*
17. #define COURSEMAX 3 *// 课程数量*
18. enum class Course : int{Chinese, Math, English};
19. class Score{
20. private:
21. vector<int> score;
22. int dataSize;
23. *// Course course;*
24. public:
25. Score():dataSize(0){}
26. int inputScore();*// 成绩输入*
27. void outputScore();*// 成绩输出*
28. void deleteScore(int pos); *// 成绩删除*
29. int queryScore(int xScore); *// 成绩查询*
30. void sortScore(); *// 成绩排序*
32. *//静态成员函数*
33. static Course switchCourse(); *// 课程切换*
34. static int login(); *// 登录*
35. static void displayMenu(); *// 主菜单*
36. static void clearScreen(); *// 清屏函数*
37. };
38. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
39. 功能:显示菜单
40. 参数：无
41. 返回值：无
42. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
43. void Score::displayMenu()
44. {
46. cout<<"++++++++++++++++++++++++++++++++++++++++++++++++++++\n";
47. cout<<"+                 学生成绩管理系统                 +\n";
48. cout<<"+                                                  +\n";
49. cout<<"+                    1. 录入                       +\n";
50. cout<<"+                    2. 输出                       +\n";
51. cout<<"+                    3. 查询                       +\n";
52. cout<<"+                    4. 排序                       +\n";
53. cout<<"+                    5. 切换课程                   +\n";
54. cout<<"+                    6. 删除                       +\n";
55. cout<<"+                    0. 结束                       +\n";
56. cout<<"+                                                  +\n";
57. cout<<"++++++++++++++++++++++++++++++++++++++++++++++++++++\n";
59. }
60. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
61. 功能:登陆函数，验证口令
62. 参数：
63. 类型：char []
64. 说明：用户输入口令
65. 返回值：1或0
66. 类型：int
67. 说明：1表示通过验证；0表示没有通过
68. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
69. int Score::login()
70. {
71. char password[8];*/\*用户输入的口令\*/*
72. static char key[8]="1234";
73. int count = 0;
74. int s=0;
76. */\*====验证用户的口令，如果三次输入均错误，则不允许进入系统====\*/*
77. do
78. {
79. count++;
80. cout<<"请输入口令：";
81. gets(password);
82. if(strcmp(password,key)==0)
83. s = 1;
84. else
85. s = 0;
86. }while(s==0 && count<3);
87. clearScreen();
88. cout<<" Reloading...\n";
89. Sleep(500);
90. cout<<" 系统登录成功! \n";
91. return s;
92. }
93. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
94. 功能:输入学生成绩到数组中
95. 参数：
96. 参数1：score[]
97. 类型：int []
98. 说明：学生成绩数组
99. 参数2：length
100. 类型：int
101. 说明：要输入的学生成绩数组的长度
102. 返回值：i
103. 类型：int
104. 说明：实际成绩数组的有效值个数
105. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
106. int Score::inputScore()
107. {
108. int i=0;
109. int tmp;
111. cout<<"请输入学生的成绩（以-1结束输入):\n";
112. do
113. {
114. cout<<"第"<<i+1<<"个学生：";
115. cin>>tmp;
116. if(tmp>=0&&tmp<=100)
117. {
118. score.push\_back(tmp);
119. i++;
120. }
121. else{
122. cout<<" 成绩不在合理范围内，请输入(0-100)以内的成绩\n";
123. }
124. }while(tmp!=-1 && i<N);
125. clearScreen();
126. cout<<" 信息录入成功，本次录入"<<i<<"条数据! "<<endl;
127. dataSize = i;
128. return dataSize;
129. }
130. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
131. 功能:输出数组中的学生成绩
132. 参数：
133. 参数1：score[]
134. 类型：int []
135. 说明：学生成绩数组
136. 参数2：length
137. 类型：int
138. 说明：要输出的成绩数组的长度
139. 返回值：无
140. 类型：void
141. 说明：
142. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
143. void Score::outputScore()
144. {
145. if(dataSize <=0){
146. cout<<" 信息为空! \n";
147. return ;
148. }
149. int i=0;
150. cout<<"学生成绩如下：\n";
151. for(i=0;i<dataSize;i++)
152. cout<<i+1<<"\t"<<score[i]<<endl;
153. cout<<"\n";
154. return ;
155. }
156. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
157. 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，
158. 否则，显示没有找到。
159. 参数：
160. 参数1：score[]
161. 类型：int []
162. 说明：学生成绩数组
163. 参数2：length
164. 类型：int
165. 说明：要输入的学生成绩数组的长度
166. 参数2：xScore
167. 类型：int
168. 说明：要查找的学生成绩
169. 返回值：k
170. 类型：int
171. 说明：
172. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
173. int Score::queryScore(int xScore)
174. {
175. auto iter = std::find(score.begin(), score.end(), xScore);
176. if (iter != score.end()) {
177. return iter - score.begin();
178. }
179. return -1;
180. }
181. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
182. 功能:删除学生的成绩
183. 参数：
184. 参数1：pos
185. 类型：int
186. 说明：学生成绩下标
187. 返回值：无
188. 类型：void
189. 说明：
190. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
191. void Score::deleteScore(int pos){
192. pos --;
193. if( pos >= dataSize ||  pos <0){
194. cout<<" 删除失败，无此下标!\n";
195. return ;
196. }
197. score.erase(score.begin()+pos);
198. --dataSize;
199. cout<< " 删除下标为："<<pos+1<<" 的成绩成功！\n";
200. }
201. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
202. 功能:排序学生的成绩，按照从小到大顺序排序。
203. 参数：
204. 参数1：score[]
205. 类型：int []
206. 说明：学生成绩数组
207. 参数2：length
208. 类型：int
209. 说明：要输入的学生成绩数组的长度
210. 返回值：无
211. 类型：void
212. 说明：
213. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
214. void Score::sortScore()
215. {
216. sort(score.begin(), score.end());
217. }
218. Course Score::switchCourse(){
219. cout << "请输入课程编号（0=Chinese, 1=Math, 2=English）: ";
220. int input;
221. while(1){
222. cin>>input;
223. if (input >= 0 && input <= 2) {
224. Course course = static\_cast<Course>(input);
225. clearScreen();
226. switch (course) {
227. case Course::Chinese:
228. std::cout << " 已切换到课程：语文" << std::endl;
229. break;
230. case Course::Math:
231. std::cout << " 已切换到课程：数学" << std::endl;
232. break;
233. case Course::English:
234. std::cout << " 已切换到课程：英语" << std::endl;
235. break;
236. }
237. break;
238. } else {
239. std::cerr << " 输入无效！请重新输入：";
240. }
241. }
243. return static\_cast<Course>(input);
244. }
245. void Score::clearScreen() {
246. #ifdef \_WIN32
247. system("cls");   *// Windows*
248. #else
249. system("clear"); *// Linux/macOS*
250. #endif
251. }
252. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
253. 功能:主函数
254. 参数：无
255. 返回值：无
256. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
257. int main()
258. {
259. int choice = 0; */\*代表用户选择的操作数字 \*/*
260. int s=0,datalen=0,x,find;
261. *// 课程成绩数组*
262. Score course[COURSEMAX];
264. */\*====验证用户的口令====\*/*
265. if (Score::login( ) == 0)
266. {
267. cout<<" 口令错误，不允许进入本系统!\n";
268. exit( 0 );
269. }
271. */\*====根据用户的选择，执行相应的操作.====\*/*
272. Course input = Course::Chinese;
273. while(1)
274. {
275. Score::displayMenu();
276. cout << "\n 当前课程: ";
277. switch (input) {
278. case Course::Chinese:
279. std::cout << "语文" << std::endl;
280. break;
281. case Course::Math:
282. std::cout << "数学" << std::endl;
283. break;
284. case Course::English:
285. std::cout << "英语" << std::endl;
286. break;
287. }
289. cout<<"\n 请选择您的操作(0,1,2,3,4,5,6)：\n";
290. cin>>choice;
291. int i = static\_cast<int>(input);
292. switch(choice)
293. {
294. case 1:
295. datalen=course[i].inputScore();
296. break;
297. case 2:
298. Score::clearScreen();
299. course[i].outputScore();
300. break;
301. case 3:
302. cout<<"\n 请输入要查找的成绩：";
303. cin>>x;
304. find=course[i].queryScore(x);
305. if(find>=0)
306. cout<<"要查成绩在数组中的下标为:"<<find<<endl;
307. else
308. cout<<"没有要查找的成绩:\n";
309. break;
310. case 4:
311. course[i].sortScore();
312. course[i].outputScore();
313. break;
314. case 5:
315. input = Score::switchCourse();
316. break;
317. case 6:
318. int pos;
319. cout<<" 请输入要删除的成绩所在下标: ";
320. cin>>pos;
321. course[i].deleteScore(pos);
322. break;
323. case 0:
324. cout<<" 正在退出成绩管理系统,请稍后..."<<endl;
325. Sleep(500);
326. exit(0);
327. default:
328. printf(" 选择错误，请重新选择!\n");
329. }
330. }
331. return 0;
332. }