**Họ và tên: Trần Đăng Hiếu**

**Lớp: 64CNTT3**

**Mã SV: 2251061779**

**Bài 2: QUẢN LÝ ĐĂNG KÝ HỌC THEO TÍN CHỈ**

Khai báo lớp ***Sinh viên(Masv, Hoten, Diachi, SoDT, lop)***, mã sinh viên là một số nguyên có 5 chữ số, tự động tăng.

Khai báo lớp ***Môn học(Mamh, Tenmon, Tongsotiet, Loaimonhoc)*** – Trong đó loại môn học có thể là: Đại cương, Cơ sở ngành, Chuyên ngành bắt buộc, Chuyên ngành tự chọn. Mã môn học là số nguyên có 3 chữ số, tự động tăng.

Khai báo lớp Bảng Đăng ký có quan hệ kế thừa 2 lớp Sinh Viên và Môn học. Biết trong đó 1 sinh viên có thể được đăng ký không quá 8 môn học một kỳ. Ghi rõ thời gian đăng ký của sinh viên.

Viết chương trình bằng ngôn ngữ C++:

1. Nhập danh sách môn học mới. In ra danh sách các môn học đã có

2. Nhập danh sách sinh viên. In ra danh sách sinh viên đã có

3. Lập bảng đăng ký cho từng sinh viên bằng cách nhập các môn học cho sinh viên và thời gian đăng ký( lấy từ thời gian hệ thống), và in danh sách ra màn hình (chú ý: cùng 1 sinh viên với 1 môn học thì không thể xuất hiện 2 lần trong bảng này)

4. sắp xếp danh sách Bảng đăng ký

a. Theo tên sinh viên

b. Theo thời gian đăng ký

5. Lập danh sách lớp học theo danh sách đăng ký. Mỗi lớp không quá 30 sinh viên và lấy theo thứ tự thời gian đăng ký.

Source Code

1. #include <iostream>
2. #include <string.h>
3. #include <vector>
4. #include <iomanip>
5. #include <ctime>
6. #include <algorithm>
8. **using** **namespace** std;
10. //Khai bao class mon hoc
11. **class** MonHoc {
12. **private**:
13. **static** **int** IDMonHoc;
14. **public**:
15. **int** MaMonHoc;
16. string TenMonHoc;
17. **int** TongSoTiet;
18. string LoaiMonHoc;
20. MonHoc(string TenMonHoc, **int** TongSoTiet, string LoaiMonHoc) {
21. **this**->MaMonHoc = IDMonHoc++;
22. **this**->TenMonHoc = TenMonHoc;
23. **this**->TongSoTiet = TongSoTiet;
24. **this**->LoaiMonHoc = LoaiMonHoc;
25. }
26. };
28. **int** MonHoc::IDMonHoc = 100;
30. // Khai bao class sinh vien
31. **class** SinhVien {
32. **private**:
33. **static** **int** IDSinhVien;
34. **public**:
35. **int** MaSinhVien;
36. string HoTen;
37. string DiaChi;
38. string SoDT;
39. string Lop;
41. SinhVien(string HoTen, string DiaChi, string SoDT, string Lop) {
42. **this**->MaSinhVien = IDSinhVien++;
43. **this**->HoTen = HoTen;
44. **this**->DiaChi = DiaChi;
45. **this**->SoDT = SoDT;
46. **this**->Lop = Lop;
47. }
48. };
50. **int** SinhVien::IDSinhVien = 10000;
52. //Khai bao class bang dang ky
53. **class** BangDangKy {
54. **public**:
55. SinhVien\* sinhVien;
56. MonHoc\* monHoc;
57. **time\_t** thoiGianDangKy;
59. BangDangKy(SinhVien\* sinhVien, MonHoc\* monHoc) {
60. **this**->sinhVien = sinhVien;
61. **this**->monHoc = monHoc;
62. **this**->thoiGianDangKy = time(0);
63. }
64. };
66. //In danh sach mon hoc
67. **void** InDanhSachMonHoc(vector<MonHoc\*> danhSachMonHoc) {
68. cout << endl;
69. cout << setw(60) << "Danh sach cac mon hoc" << endl << endl;
70. //ke bang
71. cout << setfill('-') << setw(25) << "" << setw(25) << "" << setw(25) << "" << setw(20) << "" << endl;
72. cout << setfill(' ');
73. cout << left << setw(25) << "Ma Mon Hoc" << setw(25) << "Ten Mon Hoc" << setw(25) << "Tong So Tiet" << setw(20) << "Loai Mon Hoc" << endl;
74. **for** (**int** i = 0; i < danhSachMonHoc.size(); i++) {
75. cout << setfill('-') << setw(25) << "" << setw(25) << "" << setw(25) << "" << setw(20) << "" << endl;
76. cout << setfill(' ');
77. MonHoc\* monHoc = danhSachMonHoc[i];
78. cout << setw(25) << monHoc->MaMonHoc << setw(25) << monHoc->TenMonHoc << setw(25) << monHoc->TongSoTiet << setw(20) << monHoc->LoaiMonHoc << endl;
79. }
80. cout << endl;
81. }

84. //in danh sach sinh vien
85. **void** InDanhSachSinhVien(vector<SinhVien\*> danhSachSinhVien) {
86. cout << endl;
87. cout << right << setw(65) << "Danh sach sinh vien " << endl << endl;
88. //ke bang
89. cout << setfill('-') << setw(20) << "" << setw(20) << "" << setw(30) << "" << setw(30) << "" << setw(15) << "" << setw(15) << endl;
90. cout << setfill(' ');
91. cout << left << setw(20) << "Ma Sinh Vien" << setw(20) << "Ho Ten" << setw(30) << "Dia Chi" << setw(30) << "So DT" << setw(15) << "Lop" << endl;
92. **for** (**int** i = 0; i < danhSachSinhVien.size(); i++) {
93. cout << setfill('-') << setw(20) << "" << setw(20) << "" << setw(30) << "" << setw(30) << "" << setw(15) << "" << setw(15) <<  endl;
94. cout << setfill(' ');
95. SinhVien\* sinhVien = danhSachSinhVien[i];
96. cout << setw(20) << sinhVien->MaSinhVien << setw(20) << sinhVien->HoTen << setw(30) << sinhVien->DiaChi << setw(30) << sinhVien->SoDT << setw(15) << sinhVien->Lop << endl;
97. }
98. cout << endl << endl;
99. }
101. //In bang dang ky
102. **void** InBangDangKy(vector<BangDangKy\*> danhSachBangDangKy, SinhVien\* sinhVien) {
103. cout << endl << endl;
104. cout << right << setw(60) << "Bang Dang Ky cua sinh vien " << sinhVien->MaSinhVien << ": " << endl;
105. //ke bang
106. cout << setfill('-') << setw(15) << "" << setw(15) << "" << setw(15) << "" << setw(15) << "" << setw(25) << "" << setw(15) << endl;
107. cout << setfill(' ');
108. cout << left << setw(15) << "Ma Mon Hoc" << setw(15) << "Ten Mon Hoc" << setw(15) << "Tong So Tiet"  << setw(15) << "Loai Mon Hoc" << setw(25) << "Thoi Gian Dang Ky" << endl;
109. **for** (**int** i = 0; i < danhSachBangDangKy.size(); i++) {
110. BangDangKy\* bangDangKy = danhSachBangDangKy[i];
111. **if** (bangDangKy->sinhVien == sinhVien) {
112. cout << setfill('-') << setw(15) << "" << setw(15) << "" << setw(15) << "" << setw(15) << "" << setw(25) << "" << setw(15) << endl;
113. cout << setfill(' ');
114. cout << setw(15) << bangDangKy->monHoc->MaMonHoc << setw(15) << bangDangKy->monHoc->TenMonHoc  << setw(15) << bangDangKy->monHoc->TongSoTiet << setw(15) << bangDangKy->monHoc->LoaiMonHoc << setw(25) << ctime(&bangDangKy->thoiGianDangKy);
115. }
116. }
117. cout << endl << endl;
118. }
120. //Sap xep theo ten sinh vien
121. **bool** sapXepTheoTenSinhVien(BangDangKy\* a, BangDangKy\* b) {
122. **return** a->sinhVien->HoTen < b->sinhVien->HoTen;
123. }
125. //Sap xep theo thoi gian dang ky
126. **bool** sapXepTheoThoiGianDangKy(BangDangKy\* a, BangDangKy\* b) {
127. **return** a->thoiGianDangKy < b->thoiGianDangKy;
128. }

131. **int** main() {
132. vector<MonHoc\*> danhSachMonHoc;
133. vector<SinhVien\*> danhSachSinhVien;
134. vector<BangDangKy\*> danhSachBangDangKy;
136. //Nhap danh sach mon hoc
137. **int** soMonHoc;
138. cout << "Nhap so mon hoc: ";
139. cin >> soMonHoc;
140. cout<<endl;
141. cin.ignore();
143. **for** (**int** i = 0; i < soMonHoc; i++) {
144. string TenMonHoc, LoaiMonHoc;
145. **int** TongSoTiet;
147. cout << "Nhap ten mon hoc thu " << i+1 << ": ";
148. getline(cin, TenMonHoc);
150. cout << "Nhap tong so tiet cua mon hoc thu " << i+1 << ": ";
151. cin >> TongSoTiet;
152. cin.ignore();
154. cout << "Nhap loai mon hoc thu " << i+1 << ": ";
155. getline(cin, LoaiMonHoc);
157. MonHoc\* monHoc = **new** MonHoc(TenMonHoc, TongSoTiet, LoaiMonHoc);
158. danhSachMonHoc.push\_back(monHoc);
159. cout<<endl;
160. }
162. //in danh sach mon hoc
163. InDanhSachMonHoc(danhSachMonHoc);
165. // Nhap danh sach sinh vien
166. **int** soSinhVien;
167. cout << endl<< "Nhap so sinh vien: ";
168. cin >> soSinhVien;
169. cin.ignore();
170. cout << endl;
171. **for** (**int** i = 0; i < soSinhVien; i++) {
172. string HoTen, DiaChi, SoDT, Lop;
174. cout << "Nhap ho ten sinh vien thu " << i+1 << ": ";
175. getline(cin, HoTen);
177. cout << "Nhap dia chi sinh vien thu " << i+1 << ": ";
178. getline(cin, DiaChi);
180. cout << "Nhap so dien thoai sinh vien thu " << i+1 << ": ";
181. getline(cin, SoDT);
183. cout << "Nhap Lop cua sinh vien thu " << i+1 << ": ";
184. getline(cin, Lop);
186. SinhVien\* sinhVien = **new** SinhVien(HoTen, DiaChi, SoDT, Lop);
187. danhSachSinhVien.push\_back(sinhVien);
188. cout << endl;
189. }
191. // In danh sach sinh vien
192. InDanhSachSinhVien(danhSachSinhVien);
194. //Lap bang dang ky
195. **for** (**int** i = 0; i < danhSachSinhVien.size(); i++) {
196. SinhVien\* sinhVien = danhSachSinhVien[i];
198. **for** (**int** j = 0; j < danhSachMonHoc.size(); j++) {
199. MonHoc\* monHoc = danhSachMonHoc[j];
200. string luaChon;
201. **do**{
202. cout << "Ban co muon dang ky mon hoc " << monHoc->TenMonHoc << " cho sinh vien " << sinhVien->HoTen << "? (Y/N): ";
203. getline(cin, luaChon);
204. **if** (luaChon == "Y" || luaChon == "y") {
205. cout << "Dang ky mon hoc thanh cong!" <<endl;
206. BangDangKy\* bangDangKy = **new** BangDangKy(sinhVien, monHoc);
207. danhSachBangDangKy.push\_back(bangDangKy);
208. **break**;
209. }**else** **if**(luaChon == "N" || luaChon == "n"){
210. cout << "Dang ky that bai!" << endl;
211. **break**;
212. }**else**{
213. cout << "Nhap thong tin sai! Vui long nhap lai!" <<endl;
214. }
215. }**while**(luaChon != "Y" || luaChon != "y" || luaChon != "N" || luaChon != "n" );
217. }
218. }
220. //in bang dang ky
221. **for** (**int** i = 0; i < danhSachSinhVien.size(); i++) {
222. SinhVien\* sinhVien = danhSachSinhVien[i];
223. InBangDangKy(danhSachBangDangKy, sinhVien);
224. }
226. // sap xep
227. // Sap xem theo ten sinh vien
228. sort(danhSachBangDangKy.begin(), danhSachBangDangKy.end(), sapXepTheoTenSinhVien);
229. cout << endl << "Danh sach dang ky theo ten sinh vien:" ;
230. **for** (**int** i = 0; i < danhSachSinhVien.size(); i++) {
231. SinhVien\* sinhVien = danhSachSinhVien[i];
232. InBangDangKy(danhSachBangDangKy, sinhVien);
233. }
235. // Sap xep theo thoi gian dang ky
236. sort(danhSachBangDangKy.begin(), danhSachBangDangKy.end(), sapXepTheoThoiGianDangKy);
237. cout << endl << "Danh sach dang ky theo thoi gian dang ky:";
238. **for** (**int** i = 0; i < danhSachSinhVien.size(); i++) {
239. SinhVien\* sinhVien = danhSachSinhVien[i];
240. InBangDangKy(danhSachBangDangKy, sinhVien);
241. }
243. **return** 0;
244. }

Result

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated