

Họ và tên: Tô Đức Thành

MSV: 2251061887

Lớp: 64CNTT3

BÀI 2: QUẢN LÝ ĐĂNG KÝ HỌC THEO TÍN CHỈ

Khai báo lớp **Sinh viên** (mã sinh viên, Họ tên, Địa chỉ, Số ĐT, lớp), mã sinh viên là một số nguyên có 5 chữ số, tự động tăng.

Khai báo lớp **Môn học** gồm các thuộc tính (mã môn học, Tên môn, Tổng số tiết, Loại môn học) – 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 một số nguyên có 3 chữ số, tự động tăng.

Khai báo lớp **Bảng Đăng ký** có quan hệ kết hợp (association) với lớp Sinh viên và lớp **Môn học** trong đó một sinh viên sẽ được phép đăng ký không quá 8 môn học một học kỳ. Ghi rõ thời gian đăng ký của sinh viên. Viết chương trình trong ngôn ngữ JAVA thực hiện các yêu cầu sau:

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 đó và thời gian đăng ký (lấy từ thời gian hệ thống), và in danh sách ra màn hình (chú ý: cùng một sinh viên với một 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.

Code:

```

1. #include <iostream>
2. #include <string>
3. #include <vector>
4. #include <iomanip>
5. #include <ctime>
6. #include <algorithm>
7.
8. using namespace std;
9.
10. //Khai bao lop mon hoc
11. class MonHoc {
12. private:
13.     static int nextId;
14. public:
15.     int maMonHoc;
16.     string tenMonHoc;
17.     int tongSoTiet;
18.     string loaiMonHoc;
19.
20.     MonHoc(string tenMonHoc, int tongSoTiet, string loaiMonHoc) {
21.         this->maMonHoc = nextId++;
22.         this->tenMonHoc = tenMonHoc;
23.         this->tongSoTiet = tongSoTiet;
24.         this->loaiMonHoc = loaiMonHoc;
25.     }
26. };
27.
28. int MonHoc::nextId = 100;
29.
30. // Khai bao lop sinh vien
31. class SinhVien {
32. private:
33.     static int nextId;
34. public:
35.     int maSinhVien;
36.     string hoTen;
37.     string diaChi;
38.     string soDT;
39.     string lop;
40.
41.     SinhVien(string hoTen, string diaChi, string soDT, string lop) {
42.         this->maSinhVien = nextId++;
43.         this->hoTen = hoTen;
44.         this->diaChi = diaChi;
45.         this->soDT = soDT;
46.         this->lop = lop;
47.     }
48. };
49.
50. int SinhVien::nextId = 10000;
51.
52. //Khai bao bang dang ky
53. class BangDangKy {
54. public:
55.     SinhVien* sinhVien;
56.     MonHoc* monHoc;
57.     time_t thoiGianDangKy;
58.
59.     BangDangKy(SinhVien* sinhVien, MonHoc* monHoc) {
60.         this->sinhVien = sinhVien;
61.         this->monHoc = monHoc;

```

```

62.         this->thoiGianDangKy = time(0);
63.     }
64. };
65.
66. //In danh sach mon hoc
67. void inDanhSachMonHoc(vector<MonHoc*> danhSachMonHoc) {
68.     cout << "-----Danh sach cac mon hoc da co-----" << endl;
69.     cout << left << setw(15) << "Ma Mon Hoc" << setw(15) << "Ten Mon Hoc" << set
w(15) << "Tong So Tiet" << setw(20) << "Loai Mon Hoc" << endl;
70.     for (int i = 0; i < danhSachMonHoc.size(); i++) {
71.         MonHoc* monHoc = danhSachMonHoc[i];
72.         cout << setw(15) << monHoc->maMonHoc << setw(15) << monHoc-
>tenMonHoc << setw(15) << monHoc->tongSoTiet << setw(20) << monHoc-
>loaiMonHoc << endl;
73.     }
74.     cout << endl;
75. }
76.
77. //in danh sach sinh vien
78. void inDanhSachSinhVien(vector<SinhVien*> danhSachSinhVien) {
79.     cout << "-----Danh sach sinh vien da co-----" << endl;
80.     cout << left << setw(15) << "Ma Sinh Vien" << setw(30) << "Ho Ten" << setw(3
0) << "Dia Chi" << setw(30) << "So DT" << setw(15) << "Lop" << endl;
81.     for (int i = 0; i < danhSachSinhVien.size(); i++) {
82.         SinhVien* sinhVien = danhSachSinhVien[i];
83.         cout << setw(15) << sinhVien->maSinhVien << setw(30) << sinhVien-
>hoTen << setw(30) << sinhVien->diaChi << setw(30) << sinhVien-
>soDT << setw(15) << sinhVien->lop << endl;
84.     }
85.     cout << endl;
86. }
87.
88. //In bang dang ky
89. void inBangDangKy(vector<BangDangKy*> danhSachBangDangKy, SinhVien* sinhVien) {
90.     cout << "Bang Dang Ky cua sinh vien " << sinhVien-
>maSinhVien << ": " << endl;
91.     cout << 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;
92.     for (int i = 0; i < danhSachBangDangKy.size(); i++) {
93.         BangDangKy* bangDangKy = danhSachBangDangKy[i];
94.         if (bangDangKy->sinhVien == sinhVien) {
95.             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);
96.         }
97.     }
98.     cout << endl;
99. }
100.
101. bool sapXepTheoTenSinhVien(BangDangKy* a, BangDangKy* b) {
102.     return a->sinhVien->hoTen < b->sinhVien->hoTen;
103. }
104.
105. bool sapXepTheoThoiGianDangKy(BangDangKy* a, BangDangKy* b) {
106.     return a->thoiGianDangKy < b->thoiGianDangKy;
107. }
108.

```

```

109.
110. int main() {
111.     vector<MonHoc*> danhSachMonHoc;
112.     vector<SinhVien*> danhSachSinhVien;
113.     vector<BangDangKy*> danhSachBangDangKy;
114.
115. //Nhap danh sach mon hoc
116.     int soMonHoc;
117.     cout << "Nhap so mon hoc: ";
118.     cin >> soMonHoc;
119.     cin.ignore();
120.
121.     for (int i = 0; i < soMonHoc; i++) {
122.         string tenMonHoc, loaiMonHoc;
123.         int tongSoTiet;
124.
125.         cout << "Nhap ten mon hoc thu " << i+1 << ": ";
126.         getline(cin, tenMonHoc);
127.
128.         cout << "Nhap tong so tiet cua mon hoc thu " << i+1 << ": ";
129.         cin >> tongSoTiet;
130.         cin.ignore();
131.
132.         cout << "Nhap loai mon hoc thu " << i+1 << ": ";
133.         getline(cin, loaiMonHoc);
134.
135.         MonHoc* monHoc = new MonHoc(tenMonHoc, tongSoTiet, loaiMonHoc);
136.         danhSachMonHoc.push_back(monHoc);
137.     }
138.
139. //in danh sach mon hoc
140.     inDanhSachMonHoc(danhSachMonHoc);
141.
142. // Nhap danh sach sinh vien
143.     int soSinhVien;
144.     cout << "Nhap so sinh vien: ";
145.     cin >> soSinhVien;
146.     cin.ignore();
147.
148.     for (int i = 0; i < soSinhVien; i++) {
149.         string hoTen, diaChi, soDT, lop;
150.
151.         cout << "Nhap ho ten sinh vien thu " << i+1 << ": ";
152.         getline(cin, hoTen);
153.
154.         cout << "Nhap dia chi sinh vien thu " << i+1 << ": ";
155.         getline(cin, diaChi);
156.
157.         cout << "Nhap so dien thoai sinh vien thu " << i+1 << ": ";
158.         getline(cin, soDT);
159.
160.         cout << "Nhap lop cua sinh vien thu " << i+1 << ": ";
161.         getline(cin, lop);
162.
163.         SinhVien* sinhVien = new SinhVien(hoTen, diaChi, soDT, lop);
164.         danhSachSinhVien.push_back(sinhVien);
165.     }
166.
167. // In danh sach sinh vien
168.     inDanhSachSinhVien(danhSachSinhVien);
169.

```

```

170. //Lap bang dang ky
171.     for (int i = 0; i < danhSachSinhVien.size(); i++) {
172.         SinhVien* sinhVien = danhSachSinhVien[i];
173.
174.         for (int j = 0; j < danhSachMonHoc.size(); j++) {
175.             MonHoc* monHoc = danhSachMonHoc[j];
176.
177.             string luaChon;
178.             cout << "Ban co muon dang ky mon hoc " << monHoc->
tenMonHoc << " cho sinh vien " << sinhVien->hoTen << "? (Y/N): ";
179.             getline(cin, luaChon);
180.
181.             if (luaChon == "Y" || luaChon == "y") {
182.                 BangDangKy* bangDangKy = new BangDangKy(sinhVien, monHoc);
183.                 danhSachBangDangKy.push_back(bangDangKy);
184.             }
185.         }
186.     }
187.
188. //in bang dang ky
189.     for (int i = 0; i < danhSachSinhVien.size(); i++) {
190.         SinhVien* sinhVien = danhSachSinhVien[i];
191.         inBangDangKy(danhSachBangDangKy, sinhVien);
192.     }
193. // sap xep
194.
195. // Sap xem theo ten sinh vien
196. sort(danhSachBangDangKy.begin(), danhSachBangDangKy.end(), sapXepTheoTenSinhVien
);
197. cout << "Danh sach Bang Dang Ky theo ten sinh vien:" << endl;
198. for (int i = 0; i < danhSachSinhVien.size(); i++) {
199.     SinhVien* sinhVien = danhSachSinhVien[i];
200.     inBangDangKy(danhSachBangDangKy, sinhVien);
201. }
202.
203. // Sap xep theo thoi gian dang ky
204. sort(danhSachBangDangKy.begin(), danhSachBangDangKy.end(), sapXepTheoThoiGianDan
gKy);
205. cout << "Danh sach Bang Dang Ky theo thoi gian dang ky:" << endl;
206. for (int i = 0; i < danhSachSinhVien.size(); i++) {
207.     SinhVien* sinhVien = danhSachSinhVien[i];
208.     inBangDangKy(danhSachBangDangKy, sinhVien);
209. }
210.
211.     return 0;
212. }

```

Kết quả:

```
F:\OOP\toducthanh.exe
Nhap so mon hoc: 2
Nhap ten mon hoc thu 1: 3
Nhap tong so tiet cua mon hoc thu 1: 3
Nhap loai mon hoc thu 1: 2
Nhap ten mon hoc thu 2: 1
Nhap tong so tiet cua mon hoc thu 2: 23
Nhap loai mon hoc thu 2: 3
-----Danh sach cac mon hoc da co-----
Ma Mon Hoc    Ten Mon Hoc    Tong So Tiet    Loai Mon Hoc
100            3              3              2
101            1              23             3

Nhap so sinh vien: _
```

Compile Log | Debug | Find Results | Close