Họ và tên: Trần Đăng Hiếu

Lớp: 64CNTT3

MSV: 2251061779

Câu 1: Xây dựng lớp cơ sở người (NGUOI) gồm:

Biến thành viên trong phạm vi protected: họ tên kiểu kí tự, năm sinh kiểu số nguyên

Phương thức thành viên với phạm vi truy cập công khai (public):

 Hàm tạo mặc định và hàm tạo 2 tham số

 Các hàm truy cập và biến đổi giá trị các biến thành viên (get/set)

 Phương thức Nhập thông tin

 Phương thức Xuất thông tin

Câu 2: Xây dựng lớp bệnh nhân (BENH\_NHAN) kế thừa từ lớp NGUOI bổ sung thêm:

Biến thành viên trong khu vực riêng tư gồm: tên bệnh án kiểu kí tự, số ngày nằm

viện kiểu số nguyên, giá viện phí/ngày

Phương thức thành viên với phạm vi truy cập công khai:

 Hàm tạo mặc định và hàm tạo 5 tham số

 Phương thức: Nhập, Xuất thông tin, Tính tuổi hiện nay, Tiền viện phí = số

ngày nằm viện \* giá viện phí/ngày.

Câu 3: Xây dựng lớp bác sĩ (BAC\_SI) kế thừa từ lớp NGUOI bổ sung thêm:

Biến thành viên trong khu vực riêng tư gồm: MaBS kiểu kí tự, số ngày làm việc kiểu

số thực, tiền công/ngày

Phương thức thành viên với phạm vi truy cập công khai:

 Hàm tạo mặc định và hàm tạo 5 tham số

 Phương thức: Nhập, Xuất thông tin, Tiền công tháng = số ngày làm việc \*

tiền công/ngày.

Câu 4: Xây dựng chương trình chính để thực hiện các công việc sau:

 Nhập danh sách n (0<n<=100) NGUOI. Khi nhập thông tin, chương trình hỏi

người dùng muốn nhập cho BENH\_NHAN hay BAC\_SI và sau đó nhập

đúng thông tin cho đối tượng của lớp đó.

 Hiển thị ra màn hình danh sách NGUOI nhập vào.

 Hiện ra màn hình danh sách các BENH\_NHAN có tuổi <= 10.

 Hiện ra thông tin các BAC\_SI có số ngày làm việc nhiều nhất.

Source Code

1. #include<iostream>
2. #include<string>
3. #include<vector>
4. #include<algorithm>
5. #include<iomanip>
6. #include<cstring>
8. **using** **namespace** std;
10. //class nguoi
11. **class** NGUOI{
12. **private**:
13. string HoTen;
14. **int** Namsinh;
15. **public**:
16. NGUOI(): HoTen(""), Namsinh(0){}
17. NGUOI(string HoTen, **int** Namsinh):HoTen(HoTen), Namsinh(Namsinh){}
18. **int** getNamsinh(){**return** Namsinh;}
19. string getHoTen(){
20. **return** HoTen;
21. }
22. **void** Nhap();
23. **virtual** **void** Xuat();
24. };
26. **void** NGUOI::Nhap(){
27. cout<<"Nhap ten: "; getline(cin, HoTen);
28. cout<<"Nhap Nam sinh: "; cin>>Namsinh; cin.ignore();
29. }
31. **void** NGUOI::Xuat(){
32. cout << left << setw(20) << HoTen << setw(10) << Namsinh << endl;
33. }
35. // class bac si ke thua lop nguoi
36. **class** BACSI : **public** NGUOI{
37. **private**:
38. **char** MaBS[50];
39. **float** SoNgayLamViec;
40. **float** TienCong;
41. **public**:
42. BACSI(){};
43. BACSI(**char** MaBS[50], **float** SoNgayLamViec, **float** TienCong)
44. : SoNgayLamViec(SoNgayLamViec), TienCong(TienCong) {
45. strcpy(**this**->MaBS, MaBS);
46. }
47. **void** Nhap();
48. **double** TinhLuong();
49. **void** Xuat();
50. **float** getSoNgayLamViec();
51. };
53. **void** BACSI::Nhap(){
54. NGUOI::Nhap();
55. cout<<"Nhap ma bac si: "; cin>>MaBS;
56. cout<<"Nhap so ngay lam viec: "; cin>>SoNgayLamViec;
57. cout<<"Nhap tien luong: "; cin>>TienCong;
58. }
60. **double** BACSI::TinhLuong(){
61. **return** (TienCong\*SoNgayLamViec)/30;
62. }
64. **void** BACSI::Xuat(){
65. cout << left << setw(20) << getHoTen() << setw(15) << getNamsinh() << setw(20) << MaBS << setw(40) << SoNgayLamViec << setw(20) << fixed << setprecision(0) << TienCong << setw(20) << TinhLuong() << endl;
66. }
68. **float** BACSI::getSoNgayLamViec(){
69. **return** SoNgayLamViec;
70. }
72. //class benh nhan ke thua lop nguoi
73. **class** BENHNHAN : **public** NGUOI{
74. **private**:
75. **char** Tenbenhan[50];
76. **int** Songaynam;
77. **double** Giavienphi;
78. **public**:
79. BENHNHAN(){};
80. BENHNHAN(**char** Tenbenhnhan[50], **int** Songaynam, **double** Giavienphi)
81. : Songaynam(Songaynam), Giavienphi(Giavienphi) {
82. strcpy(**this**->Tenbenhan, Tenbenhnhan);
83. }
84. **int** Tinhtuoi();
85. **double** TinhTien();
86. **void** Nhap();
87. **void** Xuat();
88. };
90. **int** BENHNHAN::Tinhtuoi(){
91. **return** 2024 - getNamsinh();
92. }
94. **double** BENHNHAN::TinhTien(){
95. **return** Giavienphi\*Songaynam;
96. }
98. **void** BENHNHAN::Nhap(){
99. NGUOI::Nhap();
100. cout<<"Nhap ten benh an: "; cin>>Tenbenhan;
101. cout<<"Nhap so ngay dieu tri: "; cin>>Songaynam;
102. cout<<"Nhap gia vien phi: "; cin>>Giavienphi;
103. }
105. **void** BENHNHAN::Xuat(){
106. cout << left << setw(20) << getHoTen() << setw(15) << getNamsinh() << setw(10) << Tinhtuoi() << setw(20) << Tenbenhan << setw(20) << Songaynam << setw(20) << fixed << setprecision(0) << Giavienphi << setw(20) << TinhTien() << endl;
107. }
109. // hàm main
110. **int** main(){
111. vector<NGUOI\*> dsNguoi;
112. vector<BACSI\*> dsBacSi;
113. vector<BENHNHAN\*> dsBenhNhan;
114. **int** n;
115. cout << "Nhap so nguoi trong danh sach: ";
116. cin >> n;
117. cin.ignore();
118. cout<<endl;
119. **char** check;
120. //Nhap thong tin
121. **for**(**int** i=1; i<=n; i++){
122. cout<<"Nhap thong tin nguoi thu "<<i<<endl;
123. cout<<"Ban muon nhap thong tin bac si hay benh nhan? Nhan phim 1 de nhap thong tin bac si, phim 2 de nhap thong tin benh nhan. "<<endl;
124. **do**{
125. cout<<"Lua chon cua ban: ";
126. cin>>check;
127. cin.ignore();
128. **if**(check=='1'){
129. BACSI\* BacSi = **new** BACSI();
130. BacSi->Nhap();
131. dsBacSi.push\_back(BacSi);
132. dsNguoi.push\_back(BacSi);
133. }**else** **if**(check=='2'){
134. BENHNHAN\* BenhNhan = **new** BENHNHAN();
135. BenhNhan->Nhap();
136. dsBenhNhan.push\_back(BenhNhan);
137. dsNguoi.push\_back(BenhNhan);
138. }**else**{
139. cout<<"Nhap sai, moi nhap lai: ";
140. }
141. }**while**(check!='1' && check!='2');
142. }
144. //in thong tin vua nhap
146. cout << endl << right << setw(70) << "Danh sach bac si" << endl;
147. cout << endl << left << setw(20) << "Ho Ten" << setw(15) << "Nam Sinh" <<setw(20) << "Ma Bac Si" << setw(40) << "So ngay lam viec"<< setw(20) << "Tien cong " << setw(20) << "Luong" << endl;
148. **for**(**int** i=0; i<dsBacSi.size(); i++){
149. dsBacSi[i]->Xuat();
150. }
152. cout << endl << endl << right << setw(70) << "Danh sach benh nhan" << endl;
153. cout << endl << left << setw(20) << "Ho Ten" << setw(15) << "Nam Sinh" << setw(10) << "Tuoi" << setw(20) << "Ten benh an" << setw(20) << "So ngay nam" << setw(20) << "Gia vien phi" << setw(20) << "Tong tien"<< endl;
154. **for**(**int** i=0; i<dsBenhNhan.size(); i++){
155. dsBenhNhan[i]->Xuat();
156. }
157. cout << endl << endl << right << setw(65) << "Danh sach benh nhan co tuoi <= 10" << endl;
158. cout << endl << left << setw(20) << "Ho Ten" << setw(15) << "Nam Sinh" << setw(10) << "Tuoi" << setw(20) << "Ten benh an" << setw(20) << "So ngay nam" << setw(20) << "Gia vien phi" << setw(20) << "Tong tien"<< endl;
159. **for**(**int** i=0; i<dsBenhNhan.size(); i++){
160. **if**(dsBenhNhan[i]->Tinhtuoi() <= 10){
161. dsBenhNhan[i]->Xuat();
162. }
163. }
164. cout << endl << endl << right << setw(60) << "Danh sach bac si co so ngay lam nhieu nhat" << endl;
165. cout << endl << left << setw(20) << "Ho Ten" << setw(15) << "Nam Sinh" <<setw(20) << "Ma Bac Si" << setw(40) << "So ngay lam viec"<< setw(20) << "Tien cong " << setw(20) << "Luong" << endl;
166. **int** maxNgaylamviec=dsBacSi[0]->getSoNgayLamViec();
167. **for**(**int** i=0; i<dsBacSi.size(); i++){
168. **if**(maxNgaylamviec < dsBacSi[i]->getSoNgayLamViec()){
169. maxNgaylamviec = dsBacSi[i]->getSoNgayLamViec();
170. }
171. }
172. **for**(**int** i=0; i<dsBacSi.size(); i++){
173. **if**(maxNgaylamviec == dsBacSi[i]->getSoNgayLamViec()){
174. dsBacSi[i]->Xuat();
175. }
176. }
177. **return** 0;
178. }

Ket qua

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated