Họ và tên: Trần Đăng Hiếu

Lớp: 64CNTT3

MSV: 2251061779

Đề bài

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

• Sắp xếp và in thông tin theo MSV + Tên

Source code

Khai báo thư viện:

1. #include <iostream>
2. #include <string>
3. #include <vector>
4. #include <algorithm>
5. #include <cstring>
6. #include <iomanip>
7. #include <utility>
8. //thu vien utility de su dung pair
10. **using** **namespace** std;

Khai báo lớp người:

1. **class** NGUOI {
2. **protected**:
3. **char** hoTen[50];
4. **int** namSinh;
5. **public**:
6. NGUOI(){};
7. NGUOI(**char** \*hoTen, **int** namSinh){
8. strcpy(**this**->hoTen, hoTen);
9. **this**->namSinh = namSinh;
10. }
11. // get hoTen
12. **char**\* getHoTen(){
13. **return** hoTen;
14. }
16. // set hoTen
17. **void** setHoTen( **char** \*newHoTen){
18. strcpy(**this**->hoTen, newHoTen);
19. }
21. // get namSinh
22. **int** getNamSinh(){
23. **return** namSinh;
24. }
26. // set namSinh
27. **void** setNamSinh(**int** newNamSinh){
28. namSinh = newNamSinh;
29. }
31. // nhap thong tin
32. **void** nhap(){
33. string hoten;
34. cout << endl << "Nhap ho ten: ";
35. cin.ignore();
36. getline(cin, hoten);
37. strcpy(**this**->hoTen, hoten.c\_str());
38. //c\_str() chuyen string thanh chuoi ki tu
39. cout << "Nhap nam sinh: ";
40. cin >> namSinh;
41. }
43. // xuat thong tin
44. **void** xuat(){
45. cout << left << setw(40) << hoTen << setw(20) << namSinh << endl;
46. }
47. };

Khai báo lớp sinh viên kế thừa lớp người

1. **class** SINHVIEN : **public** NGUOI {
2. **private**:
3. **double** MaSV;
4. **public**:
5. SINHVIEN(){};
6. SINHVIEN(**char** \*hoTen, **int** namSinh, **double** MaSV){
7. strcpy(**this**->hoTen, hoTen);
8. **this**->namSinh = namSinh;
9. **this**->MaSV = MaSV;
10. }
11. // get MaSV
12. **double** getMaSV(){
13. **return** MaSV;
14. }
15. **void** Xuat(){
16. cout << left << setw(40) << hoTen << setw(20) << fixed << setprecision(0) << MaSV << endl;
17. }
18. };

Hàm sắp xếp

1. **void** sapxep(vector<SINHVIEN\*> &dsSinhVien){
2. **for**(**int** i = 0; i < dsSinhVien.size() - 1; i++){
3. **for**(**int** j = i + 1; j < dsSinhVien.size(); j++){
4. **if**(dsSinhVien[i]->getMaSV() > dsSinhVien[j]->getMaSV()){
5. SINHVIEN \*temp = dsSinhVien[i];
6. dsSinhVien[i] = dsSinhVien[j];
7. dsSinhVien[j] = temp;
8. }
9. }
10. }
11. }

Hàm main

1. **int** main(){
2. vector<NGUOI\*> dsNguoi;
3. vector<SINHVIEN\*> dsSinhVien;
5. //Nhap thong tin nguoi
6. **int** n;
7. cout << "Nhap so luong nguoi: ";
8. cin >> n;
9. **for**(**int** i = 0; i < n; i++){
10. cout << endl << "Nhap thong tin nguoi thu " << i + 1 << endl;
11. NGUOI \*nguoi = **new** NGUOI();
12. nguoi->nhap();
13. dsNguoi.push\_back(nguoi);
14. }
16. //Xuat thong tin nguoi
17. cout << endl << endl;
18. cout << right << setw(40) << "Danh sach nguoi vua nhap: " << endl << endl;
19. cout << setfill('-') << setw(60) << "-" << setfill(' ') << endl;
20. cout << left << setw(40) << "Ho ten" << setw(20) << "Nam sinh" << endl;
21. cout << setfill('-') << setw(60) << "-" << setfill(' ') << endl;
22. **for**(**int** i = 0; i < n; i++){
23. dsNguoi[i]->xuat();
24. }
25. cout << endl << endl;
27. //Nhap thong tin sinh vien
28. **for**(**int** i=0; i < n; i++){
29. cout << endl << "Nhap MSV cho sinh vien " << dsNguoi[i]->getHoTen() << ": ";
30. **double** MaSV;
31. cin >> MaSV;
32. cin.ignore();
33. pair<**double**, NGUOI\*> sinhvien(MaSV, dsNguoi[i]);
34. SINHVIEN \*sinhVien = **new** SINHVIEN(sinhvien.second->getHoTen(), sinhvien.second->getNamSinh(), sinhvien.first);
35. dsSinhVien.push\_back(sinhVien);
36. }
38. //Sap xep va xuat thong tin sinh vien
39. cout << endl << endl;
40. sapxep(dsSinhVien);
41. cout << right << setw(45) << "Danh sach sinh vien sau khi sap xep: " << endl << endl;
42. cout << setfill('-') << setw(60) << "-" << setfill(' ') << endl;
43. cout << left << setw(40) << "Ho ten" << setw(20) << "Ma sinh vien" << endl;
44. cout << setfill('-') << setw(60) << "-" << setfill(' ') << endl;
45. **for**(**int** i = 0; i < n; i++){
46. dsSinhVien[i]->Xuat();
47. }
48. **return** 0;
49. }

Kết quả

Hàm tạo mặc định và hàm tạo 2 tham số

1. NGUOI(){};
2. NGUOI(**char** \*hoTen, **int** namSinh){
3. strcpy(**this**->hoTen, hoTen);
4. **this**->namSinh = namSinh;
5. }

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

1. // get hoTen
2. **char**\* getHoTen(){
3. **return** hoTen;
4. }
6. // set hoTen
7. **void** setHoTen( **char** \*newHoTen){
8. strcpy(**this**->hoTen, newHoTen);
9. }
11. // get namSinh
12. **int** getNamSinh(){
13. **return** namSinh;
14. }
16. // set namSinh
17. **void** setNamSinh(**int** newNamSinh){
18. namSinh = newNamSinh;
19. }

Phương thức Nhập thông tin

1. // nhap thong tin
2. **void** nhap(){
3. string hoten;
4. cout << endl << "Nhap ho ten: ";
5. cin.ignore();
6. getline(cin, hoten);
7. strcpy(**this**->hoTen, hoten.c\_str());
8. //c\_str() chuyen string thanh chuoi ki tu
9. cout << "Nhap nam sinh: ";
10. cin >> namSinh;
11. }

Phương thức Xuất thông tin

1. **void** xuat(){
2. cout << left << setw(40) << hoTen << setw(20) << namSinh << endl;
3. }
4. };

Sắp xếp và in thông tin theo MSV + Tên:

1. //Sap xep va xuat thong tin sinh vien
2. cout << endl << endl;
3. sapxep(dsSinhVien);
4. cout << right << setw(45) << "Danh sach sinh vien sau khi sap xep: " << endl << endl;
5. cout << setfill('-') << setw(60) << "-" << setfill(' ') << endl;
6. cout << left << setw(40) << "Ho ten" << setw(20) << "Ma sinh vien" << endl;
7. cout << setfill('-') << setw(60) << "-" << setfill(' ') << endl;
8. **for**(**int** i = 0; i < n; i++){
9. dsSinhVien[i]->Xuat();

    }

Ảnh chụp kết quả

A screenshot of a computer

Description automatically generated

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