**Bài tập tổng hợp**

1. **Lý thuyết**
   1. **So sánh đặc điểm của lập trình hướng đối tượng và lập trình hướng cấu trúc ?**

|  |  |  |
| --- | --- | --- |
| Đặc điểm | LT hướng cấu trúc | LT hướng đối tượng |
| Đối tượng làm việc | Các hàm (function) | Các đối tượng (class) |
| Tập trung chủ yếu vào | Sử dụng giải thuật | Sử dụng đối tượng (các thuộc tính và phương thức) |
| Tính chất | Thực hiện tuần tự theo thứ tự được lập trình sẵn | Tính kế thừa, tính đóng gói, tính đa hình, trừu tượng |
| Mở rộng chương trình | Khó | Dễ dàng nhờ các tính chất của OOP |
| Bảo trì chương trình | Khó | Dễ dàng nhờ các đối tượng được trừu tượng hóa. |
| Độ phức tạp | Thấp | Cao |
| Hiệu suất | Cao đối với các chương trình đơn giản không quá phức tạp | Cao với các chương trình phức tạp |

*Bảng so sánh lập trình hướng đối tượng và lập trình hướng cấu trúc*

* 1. **Hãy so sánh khái niệm Struct trong C và khái niệm Class (lớp) trong C++**

|  |  |  |
| --- | --- | --- |
| Đặc điểm | Struct | Class |
| Mức độ truy cập | Public | Có thể public, private tùy vào định nghĩa của class |
| Tính kế thừa | Có thể có | Có đầy đủ |
| Tính đa hình | Phải sử dụng các kỹ thuật nâng cao như con trỏ hàm | Hỗ trợ thông qua cơ chế hàm ảo, hàm thuần ảo |
| Tính đóng gói | Không | Các thông tin nội bộ của đối tượng không thể truy cập từ bên ngoài class nếu khai báo ở mức private |
| Độ bảo mật | Thấp | Cao |
| Ý nghĩa | Thường dùng để nhóm các dữ liệu đơn giản lại với nhau | Dùng để mô hình hóa đối tượng phức tạp |

*Bảng so sánh giữa struct và class*

1. **Trắc nghiệm**

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | C | **5** | B |
| **2** | A | **6** | C |
| **3** | D | **7** | B |
| **4** | B | **8** | A |

1. **Tự luận**

**Câu 1:**

#include <iostream>

#include <math.h>

using namespace std;

class SoPhuc{

private:

int a;

int b;

public:

SoPhuc();

SoPhuc(int a, int b);

SoPhuc(const SoPhuc &sp);

void setSoPhuc(int a, int b);

int getReal();

int getImag();

float module();

friend istream& operator >>(istream &is, SoPhuc &p);

friend ostream& operator <<(ostream &os, SoPhuc &p);

SoPhuc operator+(SoPhuc x);

int operator < (SoPhuc x);

friend int tong(SoPhuc x);

};

SoPhuc::SoPhuc(){

a = b = 1;

}

SoPhuc::SoPhuc(int a, int b){

this->a = a;

this->b = b;

}

SoPhuc::SoPhuc(const SoPhuc &sp){

a = sp.a;

b = sp.b;

}

int main(){

SoPhuc n, n2;

int x, y;

cout << "Nhap a, b cua so phuc a + bi: "; cin >> n;

cout << "\nn1 = "; cout << n;

cout << "\nNhap lai he so a, b: "; cin >> x >> y;

n.setSoPhuc(x, y);

cout << "\nSo phuc sau thay doi: "; cout << n;

cout << "\nNhap a, b cua so phuc thu hai: "; cin >> n2;

cout << "\nn2 = "; cout << n2;

SoPhuc tmp;

tmp = n + n2;

cout << "\nTong la: "; cout << tmp;

cout << '\n';

switch (n < n2)

{

case -1:

cout << "n1 nho hon n2";

break;

case 0:

cout << "n1 bang n2";

break;

case 1:

cout << "n1 lon hon n2";

break;

}

return 0;

}

istream& operator >>(istream &is, SoPhuc &p){

cout << "\nNhap phan thuc, phan ao: ";

is >> p.a >> p.b;

return is;

}

ostream& operator <<(ostream &os, SoPhuc &p){

if (p.a != 0){

os << p.a;

if (p.b > 0) os << " + ";

}

if (p.b < 0) os << " - ";

if (p.b != 0){

if (p.b != 1) os << p.b;

os << 'i';

}

return os;

}

void SoPhuc::setSoPhuc(int a, int b){

this->a = a;

this->b = b;

}

int SoPhuc::getReal(){

return a;

}

int SoPhuc::getImag(){

return b;

}

float SoPhuc::module(){

return sqrt((double) (a\*a + b\*b));

}

SoPhuc SoPhuc:: operator +(SoPhuc x){

SoPhuc n;

n.a = a + x.a;

n.b = b + x.b;

return n;

}

int SoPhuc::operator<(SoPhuc x){

if (this->module() < x.module())

return 1;

else if (this->module() > x.module())

return -1;

else

return 0;

}

**Câu 2:**

#include <iostream>

#include <string>

using namespace std;

class SV{

public:

SV();

SV(string hoTen, float diemVan, float diemToan);

SV(const SV& Sv);

~SV();

friend istream& operator >>(istream& is, SV &Sv);

friend istream& operator <<(ostream& os, SV &Sv);

friend void inserction\_sort(SV Sv[], int n);

string getName();

void setMa();

bool operator >(SV Sv);

bool operator >=(SV Sv);

bool operator <(SV Sv);

bool operator <=(SV Sv);

bool operator ==(SV Sv);

bool operator !=(SV Sv);

void operator =(SV Sv);

float getDTB();

private:

int maSv;

string hoTen;

float diemVan;

float diemToan;

};

SV::SV(){

hoTen = "Nguyen Thanh Tri";

diemToan = 10;

diemVan = 6;

setMa();

}

SV::SV(string hoTen, float diemVan, float diemToan){

this->hoTen = hoTen;

this->diemToan = diemToan;

this->diemVan = diemVan;

setMa();

}

SV::SV(const SV& Sv){

hoTen = Sv.hoTen;

diemToan = Sv.diemToan;

diemVan = Sv.diemVan;

setMa();

}

SV::~SV(){

}

string SV::getName(){

return hoTen;

}

void SV::setMa(){

static int id = 1;

this->maSv = id++;

}

istream &operator >>(istream& is, SV &Sv){

cout << "Nhap thong tin hoc sinh:\n";

cout << "Nhap ten hoc sinh: ";

cin.ignore();

getline(is, Sv.hoTen);

cout << "Nhap diem toan: "; is >> Sv.diemToan;

cout << "Nhap diem van: "; is >> Sv.diemVan;

}

istream &operator <<(ostream& os, SV &Sv){

os << "\n\nMa hoc sinh: " << Sv.maSv;

os << "\nHo va ten: " << Sv.hoTen;

os << "\nDiem toan: " << Sv.diemToan;

os << "\nDiem van: " << Sv.diemVan;

os << "\nDiem trung binh: " << (Sv.diemToan + Sv.diemVan)/2;

}

bool SV:: operator >(SV Sv){

float dtb1 = (diemToan + diemVan)/2;

float dtb2 = (Sv.diemToan + Sv.diemVan)/2;

return (dtb1 > dtb2);

}

bool SV:: operator >=(SV Sv){

float dtb1 = (diemToan + diemVan)/2;

float dtb2 = (Sv.diemToan + Sv.diemVan)/2;

return (dtb1 >= dtb2);

}

bool SV:: operator <(SV Sv){

float dtb1 = (diemToan + diemVan)/2;

float dtb2 = (Sv.diemToan + Sv.diemVan)/2;

return (dtb1 < dtb2);

}

bool SV:: operator <=(SV Sv){

float dtb1 = (diemToan + diemVan)/2;

float dtb2 = (Sv.diemToan + Sv.diemVan)/2;

return (dtb1 <= dtb2);

}

bool SV:: operator ==(SV Sv){

float dtb1 = (diemToan + diemVan)/2;

float dtb2 = (Sv.diemToan + Sv.diemVan)/2;

return (dtb1 == dtb2);

}

bool SV:: operator !=(SV Sv){

float dtb1 = (diemToan + diemVan)/2;

float dtb2 = (Sv.diemToan + Sv.diemVan)/2;

return (dtb1 != dtb2);

}

void SV::operator =(SV Sv){

maSv = Sv.maSv;

hoTen = Sv.hoTen;

diemVan = Sv.diemVan;

diemToan = Sv.diemToan;

}

float SV::getDTB(){

return (diemToan + diemVan) /2;

}

void inputSv(SV Sv[], int n){

for (int i = 0; i < n; ++i){

cin >> Sv[i];

}

}

void outputSv(SV Sv[], int n){

for (int i = 0; i < n; ++i){

cout << Sv[i];

}

cout << '\n';

}

void inserction\_sort(SV Sv[], int n){

for (int i = 1; i < n; ++i){

SV key = Sv[i];

int j = i - 1;

while(j >= 0 && key < Sv[i]){

Sv[j+1] = Sv[j];

j--;

}

Sv[j+1] = key;

}

}

int main(){

SV Sv[100]; int n;

cout << "So luong hoc sinh: "; cin >> n;

inputSv(Sv, n);

outputSv(Sv, n);

inserction\_sort(Sv, n);

cout << "\n====Sau khi sort===\n";

outputSv(Sv, n);

return 0;

}