**Bài thực hành số 5 (Ôn tập)**

**Ví dụ.** Xây dựng lớp Nguoi:

* Thuộc tính (private): họ tên, tuổi
* Phương thức: nhập, xuất, các phương thức khác

Xây dựng lớp Cauthu dẫn xuất của lớp Nguoi, trong đó bổ sung thêm các thuộc tính sbt (số bàn thắng), sptd (số phút thi đấu).

Viết hàm main() thực hiện:

* Nhập n cầu thủ.
* Tìm cầu thủ lớn tuổi nhất.
* Xác định số tiền thưởng của từng cầu thủ biết rằng cầu thủ thi đấu tổng cộng trên 300 phút thì được thưởng 10 triệu, và cứ ghi được 1 bàn thắng thì được thưởng thêm 5 triệu đồng.

#include<iostream>

using namespace std; class Cauthu: public Nguoi {

int bt, ptd; class Nguoi { public: string ht; void nhap();

int tuoi;

public: float tinhtien() { void nhap(); float t=0; int laytuoi() { if(ptd>300) return tuoi; t += 10;

} t += 5\*bt;

string layht() { return t;

return ht; }

} };

};

void Cauthu::nhap() {

void Nguoi::nhap() { Nguoi::nhap(); cin.ignore(1); cout<<"So ban thang: "; cout<<"Ho ten: "; cin>>bt; getline(cin, ht); cout<<"So phut dau: ";

cout<<"Tuoi: "; cin>>ptd;

cin>>tuoi; }

}

int main()

{

Cauthu ct[100]; int n, i; cout<<"So cau thu: "; cin>>n;

*// Nhap danh sach cau thu*  for(i=1; i<=n; ++i) ct[i].nhap();

cout<<endl<<"Danh sach cau thu:"<<endl; for(i=1; i<=n; ++i)

cout<<ct[i].layht()<<" "<<ct[i].laytuoi()<<endl;

*// Tim cau thu nhieu tuoi nhat*  int max= 0; for(i=1; i<=n; ++i) if(ct[i].laytuoi()>max) max= ct[i].laytuoi();

cout<<endl<<"Cau thu nhieu tuoi nhat:"<<endl;

for(i=1; i<=n; ++i) if(ct[i].laytuoi()==max)

cout<<ct[i].layht()<<" "<<ct[i].laytuoi()<<endl;

*// In danh sach cau thu va tien thuong*  cout<<endl<<"Danh sach cau thu cung tien thuong:"<<endl; for(i=1; i<=n; ++i)

{

cout<<ct[i].layht()<<" "<<ct[i].laytuoi()<<" "; cout<< ct[i].tinhtien();

}

}

# Đề 1

**Câu 1.** Xây dựng lớp Diem (Điểm), trong đó có các thuộc tính (private) gồm hoành độ và tung độ; các phương thức hoặc hàm bạn: >> để nhập tọa độ của một điểm, << để in một điểm lên màn hình theo dạng (hoành độ, tung độ), tính khoảng cách từ một điểm đến gốc tọa độ.

Viết hàm main() để:

* Nhập tọa độ n điểm trên mặt phẳng.
* Tính tổng khoảng cách từ tất cả các điểm trên đến gốc tọa độ.
* Tìm một điểm ở gần gốc tọa độ nhất.
* #include<bits/stdc++.h>
* #include<string.h>
* using namespace std;
* class *diem* {
* private:
* float x,y;
* public:
* friend *istream*& operator >> (*istream*& , *diem* &);
* friend *ostream*& operator << (*ostream*& , *diem* &);
* float khoangcach(){
* return sqrt(x\*x+y\*y);
* }
* };
* *istream*& operator >> (*istream*& *in*, *diem* &*a*) {
* *in*>>*a*.x>>*a*.y;
* return *in*;
* }
* *ostream*& operator << (*ostream*& *out*, *diem* &*a*) {
* *out* << "("<<*a*.x <<","<<*a*.y<<")"<<endl;
* return *out*;
* }
* int main(){
* int n; cout <<"nhap so diem : ";cin>>n;
* *diem* a[n];
* for(int i=0;i<n;i++){
* cin>>a[i];
* }
* double t=0;
* for(int i=0;i<n;i++){
* t+=a[i].khoangcach();
* }
* cout<<"tong khoang cach la : "<<t<<endl;
* //tim diem gan goc tọa do nhat
* double min=a[0].khoangcach();
* int vt;
* for(int i=0;i<n;i++){
* if(a[i].khoangcach() <min) vt=i;
* }
* cout <<"diem dan gox toa do nhat la ";
* for(int i=0;i<n;i++){
* if(vt==i) cout<<a[i];
* }
* }

**Câu 2.** Xây dựng lớp Toado (tọa độ), gồm các thuộc tính (private): hoành độ, tung độ; các phương thức: nhập, xuất tọa độ và tính khoảng cách đến gốc tọa độ.

Xây dựng lớp DiemMau (điểm màu) dẫn xuất từ lớp Toado, bổ sung thêm thuộc tính mau (là mã mầu có giá trị 1, 2, 3, với 1 là Đỏ, 2 là Vàng, 3 là Xanh).

Viết hàm main() để

* Nhập thông tin cho n điểm.
* Thống kê số điểm mỗi màu.
* Tìm các điểm có khoảng cách lớn nhất đến gốc tọa độ.
* #include<bits/stdc++.h>
* #include<string.h>
* using namespace std;
* class *toado* {
* private:
* float x,y;
* public:
* //friend istream& operator >> (istream& , diem &);
* //friend ostream& operator << (ostream& , diem &);
* void nhap();
* void xuat();
* float khoangcach(){
* return sqrt(x\*x+y\*y);
* }
* };
* void *toado*::nhap() {
* cin>>x>>y;
* }
* void *toado*::xuat() {
* cout << "("<<x <<","<<y<<")"<<endl;
* }
* class *diemmau* : public *toado* {
* private:
* string mau;
* public:
* void nhap();
* void xuat();
* *string* getmau() {return *this* ->mau;}
* };
* void *diemmau*::nhap() {
* *toado*::nhap();
* cout<<"ma mau : ";
* cin>>mau;
* }
* void *diemmau*::xuat() {
* *toado*::xuat();
* cout<<"ma mau : "<<mau;
* }
* int main(){
* int n; cout <<"nhap so diem : ";cin>>n;
* *diemmau* a[n];
* for(int i=0;i<n;i++){
* cout <<"diem thu "<<i<<" : " <<endl;
* a[i].nhap();
* }
* int d1=0,d2=0,d3=0;
* for(int i=0;i<n;i++){
* if(a[i].getmau() == "1") d1++;
* if(a[i].getmau() == "2") d2++;
* if(a[i].getmau() == "3") d3++;
* }
* cout <<"co "<<d1<<" diem mau do !"<<endl;
* cout <<"co "<<d2<<" diem mau vang !"<<endl;
* cout <<"co "<<d3<<" diem mau xanh !"<<endl;
* double max=a[0].khoangcach();
* for(int i=0;i<n;i++){
* if(a[i].khoangcach() >max)  max=a[i].khoangcach();
* }
* cout <<"diem dan gox toa do nhat la ";
* for(int i=0;i<n;i++){
* if(a[i].khoangcach() == max ) a[i].xuat();
* }
* return 0;
* }

# Đề 2

**Câu 1.** Xây dựng lớp DaThuc (Đa thức), trong đó có các thuộc tính (private) gồm: bậc của đa thức và các hệ số xác định đa thức; các phương thức (public): nhập đa thức, tính giá trị đạo hàm của đa thức.

Viết hàm main() để:

* Nhập các hệ số của một đa thức.
* Nhập một số thực x.
* Tính giá trị đạo hàm của đa thức tại x.
* #include<iostream>
* #include<math.h>
* using namespace std;
* class *DT*{
* int n;
* float \*a;
* public:
* DT(){   };
* DT(int *x*){
* n=*x*;
* };
* void nhap(){
* a=**new** float[n+1];
* cout<<"nhap cac bac cua da thuc:";
* for(int i=0;i<=n;i++){
* cout<<"a["<<i<<"]=";
* cin>>a[i];
* }
* }
* void xuat(){
* cout<<"he so cua da thuc la:";
* for(int i=0;i<=n;i++){
* cout<<"a["<<i<<"]="<<a[i]<<" ";
* }
* }
* float giatri(float *x*){
* float s;
* for(int i=0;i<=n;i++){
* s=s+a[i]\*pow(*x*,i);
* }
* return s;
* }
* };
* int main(){
* int n;
* cout<<"nhap he so cua da thuc la:";
* cin>>n;
* *DT* d(n);
* d.nhap();
* d.xuat();
* cout<<endl;
* float x;
* cout<<"nhap gia tri cua x la:";
* cin>>x;
* cout<<"gia tri cua da thuc tai"<<x<<"la:"<<d.giatri(x)<<endl;
* }

**Câu 2.** Xây dựng lớp QLH (quản lý hàng), gồm các thuộc tính (private): mã hàng, tên hàng, xuất xứ, loại hàng(1, 2, 3), số lượng; các phương thức (public): nhập, xuất, và các phương thức khác (nếu cần).

Xây dựng lớp DIENTU (điện tử) dẫn xuất từ lớp QLH bổ sung thêm thuộc tính tgbh (thời gian bảo hành, theo tháng).

Viết hàm main() để:

* Nhập thông tin cho n mặt hàng điện tử.
* Thống kê tổng số hàng của từng loại hàng mà có thời gian bảo hành từ 12 tháng trở lên.
* Tìm một mặt hàng có tên cho trước.

♣   #include<iostream>

♣   #include<string>

♣

♣   using namespace std;

♣   class qlh {

♣       //friend class dientu;

♣       string ma,ten,xuatxu,loai;

♣       public:

♣           friend class dientu;

♣           void nhap();

♣           void xuat();

♣           string getloai(){return *this*->loai; }

♣           string getten(){return *this*->ten;   }

♣   };

♣

♣   class dientu : public *qlh*{

♣       int tgbh;

♣       public:

♣           void nhap();

♣           void xuat();

♣           int gettgbh(){ return *this*->tgbh;}

♣   };

♣   void qlh::nhap (){

♣       cout <<"nhap ma hang : ";cin.ignore();getline(cin,ma);

♣       cout <<"nhap ten hang : ";cin.ignore();getline(cin,ten);

♣       cout <<"nhap xuat xu hang : ";cin.ignore();getline(cin,xuatxu);

♣       cout <<"nhap loai hang : ";cin.ignore();getline(cin,loai);

♣   }

♣   void dientu::nhap() {

♣       qlh::nhap() ;

♣       cout<<"nhap tgbh : ";cin>>tgbh;

♣   }

♣   void qlh::xuat() {

♣       cout <<"ma hang : "<<ma<<endl;

♣       cout <<" ten hang : "<<ten<<endl;

♣       cout <<"xuat xu hang : "<<xuatxu<<endl;

♣       cout <<"loai hang : "<<loai<<endl;

♣   }

♣   int main(){

♣       int n; cout <<"nhap so luong hang : ";cin>>n;

♣       dientu a[n];

♣       for (int i=0;i<n;i++){

♣           a[i].nhap();

♣       }

♣

♣       //thong ke

♣       int d1=0,d2=0,d3=0;

♣       for (int i=0;i<n;i++){

♣           if (a[i].getloai()=="1" && a[i].gettgbh() >=12) d1++;

♣           if (a[i].getloai()=="2" && a[i].gettgbh() >=12) d2++;

♣           if (a[i].getloai()=="3" && a[i].gettgbh() >=12) d3++;

♣       }

♣       cout <<"loai hang 1 co "<<d1<<" mat hang co bao hanh tren 12 thang!";

♣       cout <<"loai hang 2 co "<<d2<<" mat hang co bao hanh tren 12 thang!";

♣       cout <<"loai hang 3 co "<<d3<<" mat hang co bao hanh tren 12 thang!";

♣

♣       //tim hang

♣       string tentim;

♣       cout<<"tim hang : \n";

♣       cout <<"ten hang can nhap : ";cin.ignore();getline(cin,tentim);

♣       for (int i=0;i<n;i++){

♣           if(a[i].getten()==tentim ) a[i].xuat();

♣       }

♣   }

# Đề 3

**Câu 1.** Xây dựng lớp HCN (hình chữ nhật) có các thuộc tính (private) gồm độ dài cạnh dài và cạnh rộng; các phương thức (public): nhập, xuất một hình chữ nhật, tính diện tích hình chữ nhật.

Viết hàm main() để:

* Nhập kích thước cho n hình chữ nhật.
* Tính diện tích trung bình của các hình chữ nhật trên.
* Tìm những hình có diện tích nhỏ nhất

♣   class hcn {

♣       double d,r;

♣       public:

♣           hcn(){}

♣           friend *istream* &operator >> (*istream* &*is* , *hcn* &*p*);

♣           friend *ostream* &operator << (*ostream* &*os* , *hcn* *p*);

♣           double area() { return d\*r;}

♣   };

♣   *istream* &operator >> (*istream* &*is* , *hcn* &*p*) {

♣       is>>p.d>>p.r;

♣       return is;

♣   }

♣   *ostream* &operator << (*ostream* &*os* , *hcn* *p*) {

♣       os <<p.d<<" "<<p.r;

♣       return os;

♣   }

♣   int main () {

♣       int n; cin>>n;

♣       hcn a[n];

♣       for (int i=0;i<n;i++) {

♣           cout<<"hinh cn thu "<<i<<" : ";

♣           cin>>a[i];

♣       }

♣       // dien tich trung binh

♣       int t= 0 ;

♣       for (int i=0;i<n;i++) {

♣           t+=a[i].area();

♣       }

♣       cout <<"dien tich trung binh la : "<< t/n<<endl;

♣

♣       // tim s max;

♣       double min = a[1].area();

♣       for (int i=0;i<n;i++) {

♣           if (a[i].area() < min)  min=a[i].area();

♣       }

♣       cout << "nhung hinh chu nhat co dt min : "<<min<<endl;;

♣       for (int i=0;i<n;i++) {

♣           if(a[i].area() == min) cout << "hinh thu "<<i;

♣       }

♣   }

**Câu 2.** Xây dựng lớp Thisinh (thí sinh), có các thuộc tính gồm: hoten (họ tên), diem (Số điểm); các phương thức (public): nhập, xuất, đọc số điểm của thí sinh.

Xây dựng lớp Olympic dẫn xuất từ lớp Thisinh bổ sung thuộc tính: truong (trường), với giả thiết chỉ có 3 trường đại học tham gia là A, B, C; các phương thức (public): nhập, xuất...

Viết hàm main() để:

* Nhập thông tin cho n thí sinh.
* Tính tổng số điểm của mỗi trường và xác định trường có tổng điểm cao nhất.
* Thống kê số lượng sinh viên tham dự của mỗi trường.

class *thisinh* {

    string ht;

    double diem;

    public:

        void nhap () {

            cout << "nhap ten : ";cin.ignore();

            getline(cin,ht);

            cout <<"nhap diem : ";cin>>diem;

        }

        void xuat();

        double getd() {

            return *this*->diem;

        }

};

class *olim* : public *thisinh* {

    string truong;

    public:

        void nhap() {

*thisinh* :: nhap();

            cout <<"nhap ten truong : ";cin.ignore();

            getline(cin,truong);

        }

*string* gettr() {return *this*->truong;}

        void xuat();

};

int main() {

    int n; cin >>n;

*olim* a[n];

    for (int i = 0;i<n;i++) {

         a[i].nhap();

    }

    double t1=0,d1=0,t2=0,d2=0,t3=0,d3=0;

    for (int i= 0;i<n;i++) {

        if (a[i].gettr() == "A") {

            t1+=a[i].getd(); d1++;

        }

        if (a[i].gettr() == "B") {

            t2+=a[i].getd(); d2++;

        }

        if (a[i].gettr() == "C") {

            t3+=a[i].getd(); d3++;

        }

    }

    // diem ca nhat

    if(t1>=t2 && t1>=t3) cout <<"truong A co diem cao nhat = "<<t1<<endl;

    if(t1<=t2 && t2>=t3) cout <<"truong B co diem cao nhat = "<<t2<<endl;

    if(t3>=t2 && t1<=t3) cout <<"truong C co diem cao nhat = "<<t3<<endl;

    cout <<"sinh vien truong A co : "<<d1<<" thi sinh"<<endl;

    cout <<"sinh vien truong B co : "<<d2<<" thi sinh"<<endl;

    cout <<"sinh vien truong C co : "<<d3<<" thi sinh"<<endl;

}

# Đề 4

**Câu 1.** Xây dựng lớp Dayso (dãy số nguyên) có các thuộc tính (private) gồm: số tự nhiên n để lưu trữ số phần tử và một con trỏ nguyên để trỏ tới nơi lưu trữ các phần tử của dãy; các phương thức hoặc hàm tạo (public): các toán từ nhập, xuất (>> và <<) một dãy, tính trung bình cộng các phần tử của dãy dãy.

Viết hàm main() để nhập vào một dãy số nguyên rồi tính tổng dãy.

**Câu 2.** Xây dựng lớp MatHang (Mặt hàng) có các thuộc tính (private):Mã hàng (kiểu chuỗi), Đơn giá (kiểu nguyên dài); các phương thức (public): nhập và các các phương thức khác (nếu cần).

Xây dựng lớp HoaDon (Hoá đơn bán hàng) có các thuộc tính (private): Mã hàng (kiểu chuỗi), số lượng (kiểu nguyên), ngày bán (kiểu chuỗi); các phương thức (public): nhập và các phương thức khác (nếu cần).

Viết hàm main() thực hiện các yêu cầu:

* Nhập m mặt hàng
* Nhập n hoá đơn (giả thiết mã hàng của mỗi hoá đơn đều có trong m mặt hàng đã nhập).
* Đếm số hoá đơn có số lượng lớn hơn 10.
* Tính số tiền bán được trong một ngày cho trước.