

Bài thực hành số 1

Mục tiêu:

- Khởi động chương trình DevC++
- Tạo, biên dịch và thực hiện chương trình C++
- Định nghĩa lớp và khai báo đối tượng
- Cách thức truy cập và thao tác trên các hàm thành viên và các biến thành viên của lớp

1. Khởi động chương trình DevC++



2. Tạo, biên dịch và thực hiện chương trình C++

- Trong DevC, gõ Ctrl-N để mở một file mới
- Trong cửa sổ soạn thảo bạn gõ vào đoạn chương trình sau:

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello the World!";
}
```

- Lưu file vừa soạn thảo.
- Biên dịch và chạy chương trình (F11).

3. Định nghĩa lớp và khai báo đối tượng

- Tạo một file mới.
- Trong cửa sổ soạn thảo gõ vào đoạn chương trình sau:

```
#include <iostream.h>
#include <conio.h>
class MyFirstClass //định nghĩa lớp
{
public:
    int dataMember;

}; // kết thúc định nghĩa lớp.
void main()
{
    MyFirstClass aObject; // khai báo đối tượng
    aObject.dataMember = 50;
    cout << "Du lieu cua doi tuong la: " <<aObject.dataMember;
    getch();
}
```

- Lưu thành file FirstClass.CPP

- Biên dịch và chạy chương trình.

Chú ý: Một số chương trình trong tập này được soạn để thực hiện trên Turbo C++ hoặc Borland C++. Khi thực hiện chúng trên Dev C++ cần có hai thay đổi:

1. Thay cách viết các thư viện kiểu cũ (có .h) thành kiểu mới (không có .h), chẳng hạn:

```
#include <iostream.h>
```

cần viết thành:

```
#include <iostream>
using namespace std;
```

2. Thay cách viết cũ **void main()**

thành:

```
int main()
```

4. Cách thức truy cập, thao tác trên các hàm, biến thành viên của lớp

Ví dụ: Xây dựng lớp Phân số (PS) có

- Các thành phần dữ liệu (private): ts, ms
- Các phương thức (public): nhap(), xuat() và nhan()

Viết hàm main() để sử dụng lớp PS trên: nhập vào hai phân số, thực hiện nhân hai phân số đó, sau đó in phân số tích ra màn hình.

```
#include <iostream>
using namespace std;
class PS {
    int ts, ms;
public:
    void nhap();
    void xuat();
    PS nhan(PS p);
    PS cong(PS p);
};
void PS::xuat() {
    cout<<ts<<"/"<<ms;
}
void PS::nhap() {
    cout<<"\nTu so, mau so: ";
    cin>>ts>>ms;
}
PS PS::nhan(PS p) {
    PS t;
    t.ms= ms * p.ms;
    t.ts= ts * p.ts;
    return t;
}
PS PS::cong(PS p) {
    PS t;
    t.ms= ms * p.ms;
    t.ts= ts*p.ms + ms*p.ts;
    return t;
```

```

    }

    int main() {
        PS p1, p2, p3, p4;
        p1.nhap();
        p2.nhap();
        p3= p1.nhan(p2);
        cout<<"\nTich hai phan so: ";
        p3.xuat();
        p4= p1.cong(p2);
        cout<<"\nTong hai phan so: ";
        p4.xuat();
    }

```

Bài tập làm thêm:

TH1.1:

- a) Xây dựng lớp hình chữ nhật Rect, gồm có các thuộc tính (private): nWidth và nHeight, các phương thức (public):
- Input() cho phép người dùng nhập các chiều của hình chữ nhật từ bàn phím.
 - Init() khởi tạo các chiều của hình chữ nhật với các giá trị w và h.
 - Area() trả về diện tích của hình vuông.
 - IsSquare() sẽ trả về 1 nếu hình chữ nhật là hình vuông, nếu không thì trả về 0.
- b) Viết chương trình tính diện tích của hình chữ nhật và kiểm tra xem hình chữ nhật đó có phải là hình vuông hay không.
- c) Nhập một dãy N hình chữ nhật. Xác định xem trong số đó có bao nhiêu hình vuông và tính diện tích hình vuông lớn nhất.

```

class Rect // lop hình chu nhật
{
    private:

        int  nWidth; //chieu rong
        int  nHeight; //chieu dai

    public: // khai bao ham thanh vien

        void Input(); // nhap cac chieu tu ban phim
        void Init(int w,int h);// khoi tao gia tri
        int  Area() ; // tinh dien tich
        int  IsSquare();// Kiem tra xem co phai la hình vuông hay không

}; // ket thuc dinh nghĩa lop.

```

TH 1.2: Xây dựng lớp Điểm có các thuộc tính (private): hoành độ, tung độ, và các phương thức (public):

- nhap(), xuất() theo dạng (*hoành độ, tung độ*)
- kc() tính khoảng cách từ một điểm đến gốc tọa độ
- kc(Diem d) để tính khoảng cách giữa hai điểm

- Hàm bạn kc(Diem d1, Diem d2) để tính khoảng cách giữa hai điểm d1 và d2

Xây dựng hàm main() để sử dụng lớp trên.

TH 1.3: (*) Xây dựng lớp ngày (Date) cho phép nhập vào một ngày, khởi tạo một ngày, kiểm tra ngày hợp lệ, tính ngày tiếp theo, và in ngày ra màn hình. Viết hàm main() nhập hai mốc thời gian (gồm ngày, tháng, năm), sau đó tính số ngày giữa hai mốc thời gian đó.

Hướng dẫn: Định nghĩa lớp ngày như sau:

```
class Date // lop ngay
{
    private:
        int  nYear; //nam
        int  nMonth; //thang
        int  nDate; //ngay

    public: // khai bao ham thanh vien
        void Input(); // nhap ngay tu ban phim
        void Init(int yy, int mm, int dd); // khoi tao gia tri
        int  IsValid() ; // kiem tra ngay hop le
        void NextDay(Date& next); // tinh ngay tiep theo va luu vao next
        void Print();
};
```

Bài thực hành số 1 (tiếp)

Mục tiêu:

- Lớp (class) và đối tượng (object).
- Các thuộc tính và phương thức “public” và “private”.
- Mảng các đối tượng

1. Lớp, đối tượng và các phương thức

- Tạo file mới
- Gõ đoạn chương trình sau:


```
#include<iostream>
using namespace std;

class Date
{
    public:
        int day;
        int month;
        int year;
};
// lớp như trên chỉ có tác dụng như một cấu trúc !

void main()
{
    Date today;
    today.day = 25;
    today.month = 12;
    today.year = 2001;
    cout << "\n Ngày la " << today.day<< "/" ;
    cout << today.month << "/" << today.year << endl;
}
```

- Lưu file với tên “date211.cpp”
- Dịch và chạy chương trình

Kết quả trên màn hình:



Ngày la 25/12/2001

- Sửa lại file “date211.cpp”

```
#include<iostream>
using namespace std;
class date {
    int day;
    int month;
    int year;
};
void main( )
```

```

{
    date today;
    today.day = 25;
    today.month = 12;
    today.year = 2001;
    cout << "\n Ngay la " << today.day << "/" ;
    cout << today.month << "/" << today.year << endl;
}

```

- Lưu file (save as) với tên “date212.cpp”
- Dịch chương trình trên được kết quả như sau:

```

date:: day is not accessible
date:: month is not accessible
date:: year is not accessible

```

*** Sử dụng các hàm thành viên để truy cập vào các thuộc tính private:**

- Tạo file mới
- Gõ chương trình sau sau:

```

#include<iostream>
using namespace std;
class date {
    int day;
    int month;
    int year;
public:
    void getdate ( ) {
        cout << "\n Nhập vào ngày (dd): ";
        cin >> day;
        cout << "\n Nhập vào tháng (mm): ";
        cin >> month;
        cout << "\n Nhập vào năm (yy): ";
        cin >> year;

    } //ket thuc ham getdate
    void display ( ) {
        cout << "\n Ngay la " << day << "/";
        cout << month << "/" << year << endl;

    } // ket thuc ham display
};
int main( ){
    date today;
    today.getdate();
    today.display();
}

```

- Lưu file với tên “date213.cpp”
- Dịch và chạy chương trình

Kết quả:

Nhap vao ngay (dd): 25
Nhap vao thang (mm): 12
Nhap vao nam (yy): 01
Ngay la 25/12/01

*** Truyền tham số đối tượng cho các hàm thành viên**

- Tạo file mới
- Gõ đoạn chương trình sau:

```
#include<iostream>
using namespace std;
class Time {
    int hours, minutes, seconds;
public:
    void getinfo (void) {
        cin >> hours >> minutes >> seconds;
    }
    void display (void) {
        cout<<hours<<\":\"<<minutes<<\":\"<<seconds<< endl;
    }
    void addit( Time aa, Time bb) {
        hours = aa.hours + bb.hours;
        minutes = aa.minutes + bb.minutes;
        seconds = aa.seconds + bb.seconds;
        //kiem tra seconds < 60
        if( seconds >= 60)
        {
            seconds -= 60; // seconds = seconds - 60
            minutes++; //minutes = minutes + 1
        }
        //kiem tra minutes < 60
        if (minutes >= 60)
        {
            minutes -=60;
            hours++;
        }
    }
};
int main( ) {
    Time one, two, three;
    cout << "\\n Nhap vao thoi gian mot (gio phut giay): ";
    one.getinfo();

    cout << "\\n Nhap vao thoi gian hai (gio phut giay): ";
    two.getinfo();

    three.addit(one, two);
    cout << "Ket qua la ";
    three.display();
}
```

- Lưu file với tên “time214.cpp”. Dịch và chạy chương trình
- Kết quả:

```
Nhap vao thoi gian mot (gio phut giay): 12 30 53
Nhap vao thoi gian hai (gio phut giay): 4 15 28
Ket qua la 16:46:21
Ngay la 25/12/01
```

2. Mảng các đối tượng:

- Tạo file mới
- Gõ đoạn chương trình sau:

```
#include<iostream>
using namespace std;
class Student {
    int rollno;
    int marks;
public:
    void getinfo () {
        cout << "Nhap vao ma so sinh vien: ";
        cin >> rollno ;
        cout << "Nhap vao diem: ";
        cin >> marks ;
    }

    void display () { cout<<rollno<<'\t'<<marks<< endl;}
    int getmarks(){ return marks;    }

};// ket thuc lop Student
int main( )
{
    Student stulist[100];
    float total, average;
    int no, i;
    total = 0.0;
    cout << "\n Nhap vao so luong sinh vien: ";
    cin >> no ;
    for (i=0; i< no ; i++) {
        stulist[i].getinfo();
        total = total + stulist[i].getmarks();
    }
    cout << "Ma so Diem"<< endl;
    for (i=0; i< no; i++) {
        stulist[i].display();
    }
    average = total / no;
    cout << "Diem trung binh cac sinh vien = " << average;
}
```

- Lưu file với tên “stu22.cpp”. Dịch và chạy chương trình.

- Kết quả:

```
Nhap vao so luong sinh vien: 3
Nhap vao ma so sinh vien: 1
Nhap vao diem: 13
Nhap vao ma so sinh vien: 2
Nhap vao diem: 15
Nhap vao ma so sinh vien: 3
Nhap vao diem: 19
Ma so Diem
1 13
2 15
3 19
Diem trung binh cac sinh vien = 15.666667
```

Bài tập làm thêm:

TH 1.4. Xây dựng lớp Điểm, gồm:

Dữ liệu (private): hoành độ, tung độ

Các phương thức (public):

- Nhập điểm, Xuất điểm ra màn hình theo dạng (*hoành độ, tung độ*)
- Tính khoảng cách giữa hai điểm

Xây dựng hàm main():

- Nhập tọa độ ba điểm và tính diện tích của tam giác có các đỉnh là ba điểm trên.
- Nhập tọa độ của n điểm và tính độ dài đường gấp khúc lần lượt đi qua các điểm 1, 2, 3, ..., n-1, n.

TH 1.5. Xây dựng lớp DT (Đa thức), trong đó:

- Các thuộc tính:
 - int n; // là bậc của đa thức
 - float *a; // là con trỏ xác định vùng bộ nhớ chứa các hệ số
- Phương thức nhap() để nhập các hệ số của đa thức
- Phương thức xuat() để in các hệ số của đa thức ra màn hình
- Phương thức gia_tri(t) để tính giá trị của đa thức tại $x = t$.

TH 1.6. Viết chương trình quản lý học sinh với các chức năng sau:

- Thêm một học sinh
- Tìm thông tin của học sinh khi biết tên.
- Tìm học sinh có điểm trung bình cao nhất
- Xóa một học sinh ra khỏi danh sách.
- In danh sách học sinh.

