

Lab 06

Nhập xuất

Lập trình hướng đối tượng

Mục tiêu
<ol style="list-style-type: none">1. Nhập các thông tin của đối tượng từ bàn phím2. Xuất đối tượng với đầy đủ thông tin ra màn hình

1 Hướng dẫn khởi đầu

Mô tả bài tập

Cho trước thiết kế lớp **Điểm** trong không gian hai chiều với 2 thuộc tính **x** và **y**.

Hãy cài đặt cụ thể lớp này với các thành phần:

- + Thuộc tính private
- + Các hàm getter setter tương ứng.
- + Hàm tạo và hàm hủy
- + Hàm tạo có đối số
- + Hàm CalcDistanceTo để tính khoảng cách đến điểm khác

Point
- _x: float
- _y: float
+ InstanceCount: int
+ X(): const float
+ Y(): const float
+ SetX(const float)
+ SetY(const float)
+ Point()
+ Point(const float, const float)
+ Point(const Point*)
~Point()
+ CalcDistanceTo(const Point*)
+ CalcDistance(const Point*, const Point*)
+ Point(const Point&)
+ ToString(): string
+ operator>>(istream&, const Point*): istream&
+ operator<<(ostream&, Point*): ostream&

- + Thành phần tính InstanceCount đếm số lượng thể hiện đã tạo ra của lớp Điểm
- + Hàm tính CalcDistance để tính khoảng cách giữa hai điểm
- + Hàm tạo sao chép để khởi tạo thông tin từ một điểm khác
- + Hàm ToString() để chuyển một đối tượng thành chuỗi và xuất ra màn hình
- + **Nạp chồng toán tử nhập (operator>>) để nhập liệu tạo ra đối tượng**
- + **Nạp chồng toán tử xuất (operator<<) để xuất đối tượng ra màn hình**

Hướng dẫn cài đặt

Bước 1: Tạo định nghĩa lớp trong file Point.h (CPoint.h)

- Thêm một tập tin header bằng cách nhấn phải vào project, chọn **Add > New Item...**
- Chọn loại tập tin là **Header File (.h)**, đặt tên là **Point.h**
- Tạo ra định nghĩa lớp như sau:

```
#include <math.h>
#include <iostream>
#include <string>
#include <sstream>
using namespace std;

class Point {
public:
    static int InstanceCount;
private:
    float _x;
    float _y;
public:
    const float X() { return _x; }
    const float Y() { return _y; }
    void SetX(const float value) { _x = value; }
    void SetY(const float value) { _y = value; }
public:
    Point();
    Point(const float, const float);
    ~Point();
    Point(const Point&);
public:
    float CalcDistanceTo(const Point* other) const;
    string ToString() const;
public:
    static float CalcDistance(const Point* a, const Point* b);
    friend istream& operator>>(istream& in, Point* p);
    friend ostream& operator<<(ostream& out, const Point* p);
};
```

Bước 2: Cài đặt thêm hàm nhập và xuất trong file Point.cpp

```
string Point::ToString() const{
    stringstream writer;
    writer << "(" << _x << ", " << _y << ")";
    return writer.str();
}

istream& operator>>(istream& reader, Point* p)
{
    cout << "Nhap x:";
    reader >> p->_x;
    cout << "Nhap y:";
    reader >> p->_y;
    return reader;
}

ostream& operator<<(ostream& writer, const Point* p)
{
    writer << p->ToString();
    return writer;
}
```

Bước 3: Cài đặt hàm main để test việc cài đặt của lớp Point (CPoint)

```
int Point::InstanceCount = 0;

int main()
{
    Point* start = new Point();
    cin >> start;
    cout << "Diem bat dau: " << start << endl;
    delete start;
}
```

Chạy lên và thấy kết quả như sau:

```
Nhap x:3
Nhap y:7
Diem bat dau: (3, 7)
```

2 Bài tập vận dụng

Yêu cầu

- Thực hiện định nghĩa lớp theo thiết kế cho trước vào tập tin .h.
- Thực hiện cài đặt lớp trong tập tin .cpp cho lớp tương ứng.
 - + Bổ sung cài đặt toán tử nhập (operator>>)
 - + Bổ sung cài đặt toán tử xuất (operator<<)
- Viết các đoạn mã nguồn kiểm tra việc định nghĩa lớp trong hàm main.

Danh sách các lớp cần cài tiến cụ thể

- Lớp **Đường thẳng** có hai thành phần **Điểm**: **Bắt đầu** và **Kết thúc**.
- Lớp **Hình chữ nhật** có hai thành phần **Điểm**: **Trái trên** và **Phải Dưới**
- Lớp **Hình tam giác** có ba thành phần **Điểm** ứng với 3 đỉnh : **a**, **b**, **c**.
- Lớp **Hình tròn** có 2 thành phần: **tâm** (Lớp **Điểm**) và **bán kính** (số thực).
- Lớp **Phân số** có 2 thành phần: **tử** (số nguyên) và **mẫu** (số nguyên)
- Lớp **Sinh viên** có 3 thành phần: **họ** (chuỗi), **tên lót** (chuỗi) và **tên** (chuỗi).
- Lớp **Mảng động** (**DynamicArray**)
- Lớp **Thời gian** (**Time**) với 3 thành phần **giờ** (số nguyên), **phút** (số nguyên), **giây** (số nguyên)
 - + Hàm tĩnh: **IsValid(int, int, int)** cho biết 3 biến truyền vào có tạo thành thời gian hợp lệ hay không
- Lớp **Ngày Tháng** (**Date**) với 3 thành phần **ngày** (số nguyên), **tháng** (số nguyên), **năm** (số nguyên)
 - + Hàm tạo không đối: tự lấy ngày hiện tại để khởi tạo
 - + Hàm tạo có đối: khởi tạo từ ngày tháng năm truyền vào.
 - + Hàm tĩnh: **IsLeapYear(int)** cho biết năm có phải năm nhuận hay không
 - + Hàm tĩnh: **IsValid(int, int, int)** cho biết 3 biến ngày tháng năm truyền vào có tạo thành ngày tháng hợp lệ hay không

3 Hướng dẫn nộp bài

- + Mỗi bài tương ứng với 1 folder. Chỉ chứa file .h và .cpp và Makefile
- + Tổng hợp tất cả folder vào folder MSSV.
- + Nén lại tất cả thành một tập tin duy nhất MSSV.zip.

Để nộp bài, nén tất cả lại và đặt tên với định dạng MSSV.zip

-- HẾT --