

**ĐẠI HỌC QUỐC GIA  
THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA: CNTT**



**BÁO CAO  
ĐỀ THI THỰC HÀNH CUỐI KỲ  
CSC10003-PHƯƠNG PHÁP LẬP TRÌNH  
HƯỚNG ĐỐI TƯỢNG**

Ngày 25 tháng 12 năm 2021

# I. Tổng quan

Tên: Nguyễn Hoài Mẫn

Lớp: 20CLC05

## II. Báo cáo

### 1. Mô tả hệ thống

\* Trò chơi có một class CongTrinh với mục đích tạo ra công trình với các loại binh lính

```
class CongTrinh
{
private:
    int X;
    int Y;
public:
    CongTrinh(const CongTrinh& ct);
    CongTrinh();
    ~CongTrinh();
    Int getX()
    Int getY()
    virtual CongTrinh* Clone() = 0; //Phương thức nhân bản cài đặt đa hình
    virtual string LayTenCongTrinh() = 0;
    virtual void NhapCongTrinh(); //nhập thông tin của công trình
    virtual void XuatCongTrinh(); //xuat thông tin của công trình
```

- Các thuộc tính:
  - # int X: tọa độ của công trình
  - # int X: tọa độ của công trình
- Các phương thức:
  - + Int getX, int getY: các phương thức get thuộc tính của class.
  - + virtual CongTrinh\* Clone() = 0: Phương thức nhân bản cài đặt đa hình
  - + virtual string LayTenCongTrinh() = 0: lấy tên lớp đối tượng tương ứng

+ virtual void NhapCongTrinh(): phương thức thuần ảo đa hình nhập thông tin của công trình

+ virtual void XuatCongTrinh(): phương thức thuần ảo đa hình xuất thông tin của công trình

\* Class ThapPhapSu kế thừa class CongTrinh, quản lý các đặc điểm của công trình.

```
class ThapPhapSu : public CongTrinh
{
private:
    int mX;
    int mY;
    string Ten;
    int Level;
    int DEF;
    int ATK;
    int HP;
    int Mana;

public:
    ThapPhapSu();
    ~ThapPhapSu();
    string LayTenCongTrinh();
    CongTrinh* Clone();//Trả về đối tượng cần nhân bản theo cơ chế đa hình
    int getHP();
    int getMana();
    int getATK();
    int getDEF();
    void NhapCongTrinh();//Nhập thông tin của đối tượng
    void XuatCongTrinh();//xuất thông tin của đối tượng};
```

- Các thuộc tính:
  - # int mX: tọa độ của pháp sư
  - # int mY: tọa độ của pháp sư
  - # string Ten: Tên gọi
  - # int Level: cấp độ
  - # int DEF: sức phòng thủ
  - # int ATK: sức tấn công
  - # int HP: máu
  - # int Mana: mana
- Các phương thức:
  - + int getHP(), int getMana(), int getATK(), int getDEF(): các phương thức get thuộc tính của class.
  - + CongTrinh\* Clone(): Trả về đối tượng cần nhân bản theo cơ chế đa hình
  - + string LayTenCongTrinh(): lấy tên lớp đối tượng tương ứng
  - + void NhapCongTrinh(): Nhập thông tin của công trình
  - + void XuatCongTrinh(): Xuất thông tin của công trình

\* Class TraiBoBinh kế thừa class CongTrinh, quản lý các đặc điểm của công trình.

```
class TraiBoBinh :public CongTrinh
{
private:
    int mX;
    int mY;
    string Ten;
    int Level;
    int HP;
    int DEF;
    int ATK;

public:
    TraiBoBinh();
    ~TraiBoBinh();
    string LayTenCongTrinh();
    CongTrinh* Clone();//Trả về đối tượng cần nhân bản theo cơ chế đa hình
    int getHP();
    int getATK();
    int getDEF();
    void NhapCongTrinh();//Nhập thông tin của doi tuong
    void XuatCongTrinh();//Xuất thông tin doi tuong
};
```

- Các thuộc tính:

- # int mX: tọa độ của pháp sư
- # int mY: tọa độ của pháp sư
- # string Ten: Tên gọi
- # int Level: cấp độ

# int DEF: sức phòng thủ

# int ATK: sức tấn công

# int HP: máu

- Các phương thức:

+ int getHP(), int getATK(), int getDEF(): các phương thức get thuộc tính của class.

+ CongTrinh\* Clone(): Trả về đối tượng cần nhân bản theo cơ chế đa hình

+ string LayTenCongTrinh(): lấy tên lớp đối tượng tương ứng

+ void NhapCongTrinh(): Nhập thông tin của công trình

+ void XuatCongTrinh(): Xuất thông tin của công trình

\* class QuaiThu với mục đích tạo ra công trình với các loại quái thú

```
class QuaiVat
{
private:
    int X;
    int Y;
public:
    QuaiVat();
    ~QuaiVat();
    virtual QuaiVat* Clone() = 0; //Phương thức nhân bản cài đặt đa hình
    virtual string LayTenQuaiVat() = 0;
    virtual void NhapQuaiVat(); //Nhập thông tin quái thú
    virtual void XuatQuaiVat(); //Xuất thông tin quái thú
};
```

- Các thuộc tính:

# int X: tọa độ của quái thú

# int Y: tọa độ của quái thú

- Các phương thức:

+ virtual QuaiThu\* Clone() = 0: Phương thức nhân bản cài đặt đa hình

+ virtual string LayTenQuaiThu() = 0: lấy tên lớp đối tượng tương ứng  
+ virtual void NhapQuaiThu(): phương thức thuần ảo đa hình nhập thông tin của quái thú

+ virtual void XuatQuaiThu(): phương thức thuần ảo đa hình xuất thông tin của quái thú

\* Class Ma kế thừa class QuaiThu, quản lý các đặc điểm của quái thú

```
class Ma:public QuaiVat
{private:
    string TenQuaiVat;
    int Level;
    int DEF;
    int ATK;
    int HP;
    int DiemNo;
public:
    Ma();
    ~Ma();
    string LayTenQuaiVat();
    QuaiVat* Clone();//Trả về đối tượng cần nhân bản theo cơ chế đa hình
    int getDiemNo();
    int getDEF();
    int getATK();
    int getHP();
    void NhapQuaiVat();//Nhap thông tin của quái thú
    void XuatQuaiVat();//Xuat thông tin của quái thú
};
```

- Các thuộc tính:

```
# string TenQuaiVat: Tên gọi
# int Level: cấp độ

# int DEF: sức phòng thủ
# int ATK: sức tấn công
# int HP: máu
# int DiemNo: Điểm nô
```

- Các phương thức:
  - + int getHP(), int getATK(), int getDEF(), int getDiemNo(): các phương thức get thuộc tính của class.
  - + QuaiVat\* Clone(): Trả về đối tượng cần nhân bản theo cơ chế đa hình
  - + string LayTenQuaiVat(): lấy tên lớp đối tượng tương ứng
  - + void NhapQuaiVat(): Nhập thông tin của quái vật
  - + void XuatQuaiVat(): Xuất thông tin của quái vật

\* Class QuySoi kế thừa class QuaiThu, quản lý các đặc điểm của quái thú



```

class QuySoi:public QuaiVat
{
private:
    string TenQuaiVat;
    int Level;
    int DEF;
    int ATK;
    int HP;
    int DiemNo;
public:
    QuySoi();
    ~QuySoi();
    string LayTenQuaiVat();
    QuaiVat* Clone();//Trả về đối tượng cần nhân bản theo cơ chế đa hình
    int getDiemNo();
    int getDEF();
    int getATK();
    int getHP();
    void NhapQuaiVat();//Nhap thông tin của quái thủ
    void XuatQuaiVat();//Xuat thông tin của quái thủ
};

```

- Các thuộc tính:
  - # string TenQuaiVat: Tên gọi
  - # int Level: cấp độ
  - # int DEF: sức phòng thủ
  - # int ATK: sức tấn công
  - # int HP: máu
  - # int DiemNo: Điểm nộ

- Các phương thức:
  - + int getHP(), int getATK(), int getDEF(), int getDiemNo(): các phương thức get thuộc tính của class.
  - + QuaiVat\* Clone(): Trả về đối tượng cần nhân bản theo cơ chế đa hình
  - + string LayTenQuaiVat(): lấy tên lớp đối tượng tương ứng
  - + void NhapQuaiVat(): Nhập thông tin của quái vật
  - + void XuatQuaiVat(): Xuất thông tin của quái vật

\* Class Player quản lý hoạt động của người chơi

```
class Player
{
private:
    int X, Y;
public:
    Player();
    Player(int mX, int mY);
    ~Player();
    bool up();//Xu ly tang toa do Y + 1
    bool down();//Xu ly tang toa do Y - 1
    bool left();//Xu ly tang toa do X + 1
    bool right();//Xu ly tang toa do X - 1
    bool Win();//Xu ly thang
    bool Lose();//Xu ly thua
}
```

- Các thuộc tính:
  - # int X, Y: Tọa độ của người chơi
- Các phương thức:
  - + bool up(), bool down(), bool left(), bool right(): di chuyển của người chơi
  - + bool Win(), bool Lose(): xử lý thắng thua

\* Class Game quản lý toàn bộ các tiến trình game.

```
class Game
{
private:
    static vector<CongTrinh*>ListCongTrinh;//vector cac doi tuong ban mau
    static vector<QuaiVat*>ListQuaiVat;//vector cac doi tuong ban mau
public:
    Game();
    ~Game();
    voidNhapCongTrinh();//Nhap thong tin cua cong trinh
    voidXuatCongTrinh();//Xuat thong tin cua cong trinh
    voidNhapQuaiVat();//Nhap thong tin cua quai thu
    voidXuatQuaiVat();//Xuat thong tin cua quai thu
    static voidThemDoiTuongMau(CongTrinh* ns);//Method them doi tuong
    ban mau vao danh sach ban mau
    static CongTrinh* TaoDoiTuongCongTrinh(string newTen);//Method ho
    tro tao doi tuong dua theo ten
    static voidThemDoiTuongMau(QuaiVat* ct);
    static QuaiVat* TaoDoiTuongQuaiVat(string newTen);
};
```

- Các thuộc tính:
  - # static vector<CongTrinh\*>ListCongTrinh: vector cac doi tuong ban mau
  - # static vector<QuaiVat\*>ListQuaiVat;//vector cac doi tuong ban mau
- Các phương thức:
  - + voidNhapCongTrinh();//Nhap thong tin cua cong trinh
  - + voidXuatCongTrinh();//Xuat thong tin cua cong trinh
  - + voidNhapQuaiVat();//Nhap thong tin cua quai thu

- + void XuatQuaiVat();//Xuat thong tin cua quai thu
- + static void ThemDoiTuongMau(CongTrinh\* ns): Method them doi tuong ban mau vao danh sach ban mau
- + static CongTrinh\* TaoDoiTuongCongTrinh(string newTen): Method ho tro tao doi tuong dua theo ten
- + static void ThemDoiTuongMau(QuaiVat\* ct): Method them doi tuong ban mau vao danh sach ban mau
- + static QuaiVat\* TaoDoiTuongQuaiVat(string newTen): Method ho tro tao doi tuong dua theo ten

## **2. Khả năng mở rộng**

- Chương trình sử dụng mẫu thiết kế Prototype ở các class CongTrinh, QuaiThu nên có thể dễ dàng mở rộng ra thêm các loại công trình và quái thú khác