

**ĐÁP ÁN THI THỬ**

**Câu 1:**

1) Khái niệm Kế thừa:

- Kế thừa là một đặc điểm của ngôn ngữ dùng để biểu diễn mối quan hệ đặc biệt hóa – tổng quát hóa giữa các lớp. Các lớp được trừu tượng hóa và được tổ chức thành một sơ đồ phân cấp lớp.
- Sự kế thừa là một mức cao hơn của trừu tượng hóa, cung cấp một cơ chế gom chung các lớp có liên quan với nhau thành một mức khái quát hóa đặc trưng cho toàn bộ các lớp nói trên.

Ưu điểm của Kế thừa:

- Kế thừa cho phép xây dựng lớp mới từ lớp đã có.
- Kế thừa cho phép tổ chức các lớp chia sẻ mã chương trình chung, nhờ vậy có thể dễ dàng sửa chữa, nâng cấp hệ thống.
- Trong C++, kế thừa còn định nghĩa sự tương thích, nhờ đó ta có cơ chế chuyển kiểu tự động.
- Trong lớp ta có thể bổ sung thêm các thành phần hoặc định nghĩa lại các thành phần.

Ví dụ minh họa: Thí sinh tự cho ví dụ

2) Khái niệm Đa hình:

- Đa hình: Là hiện tượng các đối tượng thuộc các lớp khác nhau có khả năng hiểu cùng một thông điệp theo các cách khác nhau.

Ưu điểm của Đa hình:

- Không phải viết lại mã hoặc lớp đã có sẵn.
- Dùng một tên duy nhất để lưu trữ biến của nhiều kiểu dữ liệu khác nhau (float, double, long, int,...).
- Tạo ra sự linh hoạt trong việc sử dụng tính thừa kế để nâng cấp, phát triển chương trình.

Ví dụ minh họa: Thí sinh tự cho ví dụ

**Câu 2:**

```
class cSoPhuc {  
private:  
    float thuc, ao;  
public:  
    cSoPhuc();  
    cSoPhuc(float x, float y);  
};
```

```

    friend std::istream& operator>>(std::istream& is, cSoPhuc& a);
    friend std::ostream& operator<<(std::ostream& os, cSoPhuc a);

    friend cSoPhuc operator+(cSoPhuc a, cSoPhuc b);
    friend cSoPhuc operator+(float a, cSoPhuc b);

    cSoPhuc operator++();
    cSoPhuc operator++(int);
};

cSoPhuc::cSoPhuc()
{
    thuc = 0;
    ao = 0;
}
cSoPhuc::cSoPhuc(float a, float b) {
    this->thuc = a;
    this->ao = b;
}

cSoPhuc cSoPhuc::operator++() {
    this->thuc++;
    return *this;
}

cSoPhuc cSoPhuc::operator++(int) {
    cSoPhuc temp = *this;
    this->thuc++;
    return temp;
}

cSoPhuc operator+(cSoPhuc sp1, cSoPhuc sp2) {
    return cSoPhuc(sp1.thuc + sp2.thuc, sp1.ao + sp2.ao);
}

cSoPhuc operator+(float x, cSoPhuc sp) {
    return cSoPhuc(x + sp.thuc, sp.ao);
}

std::ostream& operator<<(std::ostream& os, cSoPhuc sp) {
    os << sp.thuc << " + " << sp.ao << "i";
    return os;
}

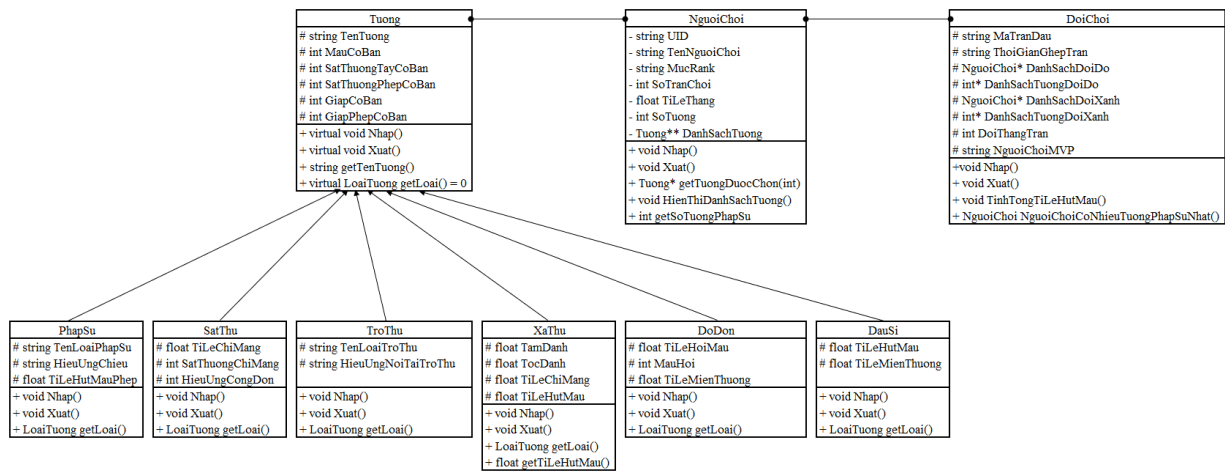
std::istream& operator>>(std::istream& is, cSoPhuc& sp) {
    is >> sp.thuc >> sp.ao;
    return is;
}

```

### Câu 3:

Đáp án tham khảo:

- Sơ đồ lớp:



- Chương trình:

## Tuong.h

```
#pragma once
#include <iostream>
#include <string>
using namespace std;
enum LoaiTuong {TuongPhapSu, TuongXaThu, TuongTroThu, TuongDoDon,
TuongDauSi, TuongSatThu};
class Tuong
{
protected:
    string TenTuong;
    int MauCoBan;
    int SatThuongTayCoBan;
    int SatThuongPhepCoBan;
    int GiapCoBan;
    int GiapPhepCoBan;
public:
    virtual void Nhap();
    virtual void Xuat();
    string getTenTuong();
    virtual LoaiTuong getLoai() = 0;
};
```

## Tuong.cpp

```
#include "Tuong.h"
void Tuong::Nhap()
{
    cin.ignore();
    cout << "Nhap ten tuong: ";
    getline(cin, TenTuong);
    cout << "Nhap mau co ban: ";
    cin >> MauCoBan;
    cout << "Nhap sat thuong tay co ban: ";
    cin >> SatThuongTayCoBan;
    cout << "Nhap sat thuong phep co ban: ";
    cin >> SatThuongPhepCoBan;
    cout << "Nhap giap co ban: ";
```

```

        cin >> GiapCoBan;
        cout << "Nhap giap phep co ban: ";
        cin >> GiapPhepCoBan;
    }
    string Tuong::getTenTuong()
    {
        return TenTuong;
    }
    void Tuong::Xuat()
    {
        cout << "Ten tuong: " << TenTuong << endl;
        cout << "Mau co ban: " << MauCoBan << endl;
        cout << "Sat thuong tay co ban: " << SatThuongTayCoBan << endl;
        cout << "Sat thuong phep co ban: " << SatThuongPhepCoBan << endl;
        cout << "Giap co ban: " << GiapCoBan << endl;
        cout << "Giap phep co ban: " << GiapPhepCoBan << endl;
    }
}

```

PhapSu.h

```

#pragma once
#include "Tuong.h"
class PhapSu :
    public Tuong
{
protected:
    string TenLoaiPhapSu;
    string HieuUngChieu;
    float TiLeHutMauPhep;
public:
    void Nhap();
    void Xuat();
    LoaiTuong getLoai();
};

```

PhapSu.cpp

```

#include "PhapSu.h"
void PhapSu::Nhap()
{
    Tuong::Nhap();
    cin.ignore();
    cout << "Nhap ten loai phap su: ";
    getline(cin, TenLoaiPhapSu);
    cout << "Nhap hieu ung chieu: ";
    getline(cin, HieuUngChieu);
    cout << "Nhap ti le hut mau phep: ";
    cin >> TiLeHutMauPhep;
}
void PhapSu::Xuat()
{
    Tuong::Xuat();
    cout << "Class: Phap su" << endl;
    cout << "Ten loai phap su: " << TenLoaiPhapSu << endl;
    cout << "Hieu ung chieu: " << HieuUngChieu << endl;
    cout << "Ti le hut mau phep: " << TiLeHutMauPhep << endl;
}

```

```

LoaiTuong PhapSu::getLoai()
{
    return LoaiTuong::TuongPhapSu;
}

```

SatThu.h

```

#pragma once
#include "Tuong.h"
class SatThu :
    public Tuong
{
protected:
    float TileChiMang;
    int SatThuongChiMang;
    int HieuUngCongDon;
public:
    void Nhap();
    void Xuat();
    LoaiTuong getLoai();
};

```

SatThu.cpp

```

#include "SatThu.h"
void SatThu::Nhap()
{
    Tuong::Nhap();
    cout << "Nhap ti le chi mang: ";
    cin >> TileChiMang;
    cout << "Nhap sat thuong chi mang: ";
    cin >> SatThuongChiMang;
    cout << "Nhap sat thuong cong don: ";
    cin >> HieuUngCongDon;
}
void SatThu::Xuat()
{
    Tuong::Xuat();
    cout << "Class: Sat thu" << endl;
    cout << "Ti le chi mang: " << TileChiMang << endl;
    cout << "Nhap sat thuong chi mang: " << SatThuongChiMang << endl;
    cout << "Nhap sat thuong cong don: " << HieuUngCongDon << endl;
}
LoaiTuong SatThu::getLoai()
{
    return LoaiTuong::TuongSatThu;
}

```

DauSi.h

```

#pragma once
#include "Tuong.h"
class DauSi :
    public Tuong
{
protected:
    float TileMienThuong;
    float TileHutMau;
}

```

```

public:
    void Nhap();
    void Xuat();
    LoaiTuong getLoai();
};

```

DauSi.cpp

```

#include "DauSi.h"
void DauSi::Nhap()
{
    Tuong::Nhap();
    cout << "Nhap ti le mien thuong: ";
    cin >> TileMienThuong;
    cout << "Nhap ti le hut mau: ";
    cin >> TileHutMau;
}
void DauSi::Xuat()
{
    Tuong::Xuat();
    cout << "Class: Dau si" << endl;
    cout << "Ti le mien thuong: " << TileMienThuong << endl;
    cout << "Ti le hut mau: " << TileHutMau << endl;
}
LoaiTuong DauSi::getLoai()
{
    return LoaiTuong::TuongDauSi;
}

```

XaThu.h

```

#pragma once
#include "Tuong.h"
class XaThu :
    public Tuong
{
protected:
    float TamDanh;
    float TocDanh;
    float TileChiMang;
    float TileHutMau;
public:
    void Nhap();
    void Xuat();
    LoaiTuong getLoai();
    float getTileHutMau();
};

```

XaThu.cpp

```

#include "XaThu.h"
void XaThu::Nhap()
{
    Tuong::Nhap();
    cout << "Nhap tam danh: ";
    cin >> TamDanh;
    cout << "Nhap toc danh: ";
    cin >> TocDanh;
}

```

```

        cout << "Nhap ti le chi mang: ";
        cin >> TileChiMang;
        cout << "Nhap ti le hut mau: ";
        cin >> TileHutMau;
    }
    void XaThu::Xuat()
    {
        Tuong::Xuat();
        cout << "Class: Xa thu" << endl;
        cout << "Tam danh: " << TamDanh << endl;
        cout << "Toc danh: " << TocDanh << endl;
        cout << "Ti le chi mang: " << TileChiMang << endl;
        cout << "Ti le hut mau: " << TileHutMau << endl;
    }
    LoaiTuong XaThu::getLoai()
    {
        return LoaiTuong::TuongXaThu;
    }
    float XaThu::getTileHutMau()
    {
        return TileHutMau;
    }
}

```

TroThu.h

```

#pragma once
#include "Tuong.h"
class TroThu :
    public Tuong
{
protected:
    string TenLoaiTroThu;
    string HieuUngNoiTaiTroThu;
public:
    void Nhap();
    void Xuat();
    LoaiTuong getLoai();
};

```

TroThu.cpp

```

#include "TroThu.h"
void TroThu::Nhap()
{
    Tuong::Nhap();
    cout << "Nhap loai tro thu: ";
    cin >> TenLoaiTroThu;
    cout << "Nhap hieu ung noi tai tro thu: ";
    cin >> HieuUngNoiTaiTroThu;
}
void TroThu::Xuat()
{
    Tuong::Xuat();
    cout << "Class: Tro thu" << endl;
    cout << "Loai tro thu: " + TenLoaiTroThu << endl;
    cout << "Hieu ung noi tai tro thu: " + HieuUngNoiTaiTroThu << endl;
}

```

```

LoaiTuong TroThu::getLoai()
{
    return LoaiTuong::TuongTroThu;
}

```

SatThu.h

```

#pragma once
#include "Tuong.h"
class SatThu :
    public Tuong
{
protected:
    float TileChiMang;
    int SatThuongChiMang;
    int HieuUngCongDon;
public:
    void Nhap();
    void Xuat();
    LoaiTuong getLoai();
};

```

SatThu.cpp

```

#include "SatThu.h"
void SatThu::Nhap()
{
    Tuong::Nhap();
    cout << "Nhap ti le chi mang: ";
    cin >> TileChiMang;
    cout << "Nhap sat thuong chi mang: ";
    cin >> SatThuongChiMang;
    cout << "Nhap sat thuong cong don: ";
    cin >> HieuUngCongDon;
}
void SatThu::Xuat()
{
    Tuong::Xuat();
    cout << "Class: Sat thu" << endl;
    cout << "Ti le chi mang: " << TileChiMang << endl;
    cout << "Nhap sat thuong chi mang: " << SatThuongChiMang << endl;
    cout << "Nhap sat thuong cong don: " << HieuUngCongDon << endl;
}
LoaiTuong SatThu::getLoai()
{
    return LoaiTuong::TuongSatThu;
}

```

NguoiChoi.h

```

#pragma once
#include "DauSi.h"
#include "DoDon.h"
#include "PhapSu.h"
#include "TroThu.h"
#include "XaThu.h"
#include "SatThu.h"
class NguoiChoi

```



```

{
private:
    string UID;
    string TenNguoiChoi;
    string MucRank;
    int SoTranChoi;
    float TiLeThang;
    int SoTuong;
    Tuong** DanhSachTuong;
public:
    void Nhap();
    void Xuat();
    Tuong* getTuongDuocChon(int);
    void HienThiDanhSachTuong();
    int getSoTuongPhapSu();
};

```

NguoiChoi.cpp

```

#include "NguoiChoi.h"
void NguoiChoi::HienThiDanhSachTuong()
{
    cout << "DANH SACH TUONG" << endl;
    for (int i = 0; i < SoTuong; i++)
    {
        cout << (i + 1) << ". " + DanhSachTuong[i]->getTenTuong() <<
endl;
    }
}
void NguoiChoi::Nhap()
{
    cin.ignore();
    cout << "Nhap ma UID: ";
    getline(cin, UID);
    cout << "Nhap ten nguoi choi: ";
    getline(cin, TenNguoiChoi);
    cout << "Nhap muc rank: ";
    getline(cin, MucRank);
    cout << "Nhap so tran choi: ";
    cin >> SoTranChoi;
    cout << "Nhap ti le thang: ";
    cin >> TiLeThang;
    cout << "Nhap so tuong so huu: ";
    cin >> SoTuong;
    DanhSachTuong = new Tuong * [SoTuong];
    for (int i = 0; i < SoTuong; i++)
    {
        cout << "Nhap vi tuong thu " << i + 1 << endl;
        cout << "Chon loai tuong muon nhap: " << endl;
        cout << "\t1. Phap su" << endl;
        cout << "\t2. Tro thu" << endl;
        cout << "\t3. Xa thu" << endl;
        cout << "\t4. Dau si" << endl;
        cout << "\t5. Do don" << endl;
        cout << "\t6. Sat thu" << endl;
        cout << "Moi nhap lua chon: ";
        int iluachon;
    }
}

```

```

        cin >> iluachon;
        switch (iluachon)
        {
            case 1:
                DanhSachTuong[i] = new PhapSu;
                break;
            case 2:
                DanhSachTuong[i] = new TroThu;
                break;
            case 3:
                DanhSachTuong[i] = new XaThu;
                break;
            case 4:
                DanhSachTuong[i] = new DauSi;
                break;
            case 5:
                DanhSachTuong[i] = new DoDon;
                break;
            case 6:
                DanhSachTuong[i] = new SatThu;
                break;
        }
        DanhSachTuong[i]->Nhap();
    }
}

int Nguoichi::getSoTuongPhapSu()
{
    int dem = 0;
    for (int i = 0; i < SoTuong; i++)
        if (DanhSachTuong[i]->getLoai() == LoaiTuong::TuongPhapSu)
            dem++;
    return dem;
}

void Nguoichi::Xuat()
{
    cout << "Ma UID: " << UID << endl;
    cout << "Ten nguoi choi: " << TenNguoiChoi << endl;
    cout << "Rank: " << MucRank << endl;
    cout << "So tran: " << SoTranChoi << endl;
    cout << "Ti le thang: " << TiLeThang << endl;
    cout << "So tuong dang co: " << SoTuong << endl;
}

Tuong* Nguoichi::getTuongDuocChon(int i)
{
    return DanhSachTuong[i];
}

```

DoiChoi.h

```

#pragma once
#include "NguoiChoi.h"
class DoiChoi
{
private:
    string MaTranDau;
    string ThoiGianGhepTran;
    Nguoichi* DanhSachDoiDo;

```

```

    int* DanhSachTuongDoiDo;
    NguoiChoi* DanhSachDoiXanh;
    int* DanhSachTuongDoiXanh;
    int DoiThangTran;
    string NguoiChoiMVP;
public:
    void Nhap();
    void Xuat();
    void TinhTongTiLeHutMau();
    NguoiChoi NguoiChoiCoNhiềuTuongPhapSuNhat();
};

```

DoiChoi.cpp

```

#include "DoiChoi.h"
void DoiChoi::Nhap()
{
    cout << "Nhap ma tran dau: ";
    cin >> MaTranDau;
    cout << "Nhap thoi gian ghep tran: ";
    cin >> ThoiGianGhepTran;
    DanhSachDoiDo = new NguoiChoi[5];
    DanhSachTuongDoiDo = new int[5];
    cout << "NHAP THONG TIN DOI DO" << endl;
    for (int i = 0; i < 5; i++)
    {
        cout << "Nhap thong tin nguoi choi thu " << i + 1 << endl;
        DanhSachDoiDo[i].Nhap();
        cout << "TIEN HANH CHON TUONG: " << endl;
        DanhSachDoiDo[i].HienThiDanhSachTuong();
        int iluachon;
        cout << "Moi nhap lua chon: ";
        cin >> iluachon;
        DanhSachTuongDoiDo[i] = iluachon - 1;
    }
    DanhSachDoiXanh = new NguoiChoi[5];
    DanhSachTuongDoiXanh = new int[5];
    cout << "NHAP THONG TIN DOI XANH" << endl;
    for (int i = 0; i < 5; i++)
    {
        cout << "Nhap thong tin nguoi choi thu " << i + 1 << endl;
        DanhSachDoiXanh[i].Nhap();
        cout << "TIEN HANH CHON TUONG: " << endl;
        DanhSachDoiXanh[i].HienThiDanhSachTuong();
        int iluachon;
        cout << "Moi nhap lua chon: ";
        cin >> iluachon;
        DanhSachTuongDoiXanh[i] = iluachon - 1;
    }
    cout << "Nhap doi thang tran (1: Do, 2: Xanh): ";
    cin >> DoiThangTran;
    cin.ignore();
    cout << "Nhap ten nguoi choi MVP: ";
    getline(cin, NguoiChoiMVP);
}
void DoiChoi::Xuat()
{

```

```

cout << "THONG TIN DOI DO" << endl;
for (int i = 0; i < 5; i++)
{
    cout << "Thong tin thanh vien thu " << i + 1 << endl;
    DanhSachDoiDo[i].Xuat();
    cout << "Thong tin tuong duoc chon: " << endl;
    DanhSachDoiDo[i].getTuongDuocChon(DanhSachTuongDoiDo[i]) -
>Xuat();
}
cout << "THONG TIN DOI XANH" << endl;
for (int i = 0; i < 5; i++)
{
    cout << "Thong tin thanh vien thu " << i + 1 << endl;
    DanhSachDoiXanh[i].Xuat();
    cout << "Thong tin tuong duoc chon: " << endl;
    DanhSachDoiXanh[i].getTuongDuocChon(DanhSachTuongDoiXanh[i]) -
>Xuat();
}
}
void DoiChoi::TinhTongTileHutMau()
{
    float sum1 = 0, sum2 = 0;
    for (int i = 0; i < 5; i++)
    {
        if (DanhSachDoiDo[i].getTuongDuocChon(DanhSachTuongDoiDo[i]) -
>getLoai() == LoaiTuong::TuongXaThu)
            sum1 = sum1 +
((XaThu*)(DanhSachDoiDo[i].getTuongDuocChon(DanhSachTuongDoiDo[i])))-
>getTileHutMau();
        if
(DanhSachDoiXanh[i].getTuongDuocChon(DanhSachTuongDoiXanh[i]) ->getLoai()
== LoaiTuong::TuongXaThu)
            sum2 = sum2 +
((XaThu*)(DanhSachDoiXanh[i].getTuongDuocChon(DanhSachTuongDoiXanh[i])))-
>getTileHutMau();
    }
    cout << "Ti le hut mau xa thu doi do: " << sum1 << "%" << endl;
    cout << "Ti le hut mau xa thu doi xanh: " << sum2 << "%" << endl;
    if (sum1 > 20 || sum2 > 20) cout << "Trang bi Sach truy hon duoc su
dung" << endl;
    else cout << "Trang bi sach truy hon khong duoc su dung" << endl;
}
NguoiChoi DoiChoi::NguoiChoiCoNhieuTuongPhapSuNhat()
{
    NguoiChoi NguoiChoiMax;
    int max = -1;
    for (int i = 0; i < 5; i++)
    {
        if (DanhSachDoiDo[i].getSoTuongPhapSu() > max)
        {
            max = DanhSachDoiDo[i].getSoTuongPhapSu();
            NguoiChoiMax = DanhSachDoiDo[i];
        }
        if (DanhSachDoiXanh[i].getSoTuongPhapSu() > max)
        {
            max = DanhSachDoiXanh[i].getSoTuongPhapSu();
            NguoiChoiMax = DanhSachDoiXanh[i];
        }
    }
}

```

```
        }  
    }  
    return NgnoiChoiMax;  
}
```