

ĐÁP ÁN – THANG ĐIỂM

Câu 1.

a. Phân biệt khái niệm overload (tải chồng) và override (ghi đè) (1đ)

	Override	Overload
Khái niệm	Là một tính năng cho phép một lớp con hoặc lớp con cung cấp một triển khai cụ thể của một phương thức đã được cung cấp bởi một trong các lớp siêu hoặc các lớp cha của nó. Nói cách khác, nếu lớp con cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi một trong các lớp cha của nó, thì đó là ghi đè phương thức.	Nạp chồng phương thức đơn giản là có vài phương thức trùng tên nhưng khác nhau về đối số. Cài chồng phương thức cho phép ta tạo nhiều phiên bản của một phương thức, mỗi phiên bản chấp nhận một danh sách đối số khác nhau, nhằm tạo thuận lợi cho việc gọi phương thức.
Hành vi	Thay đổi hành vi hiện tại của phương thức.	Thêm hoặc mở rộng cho hành vi của phương thức.
Đa hình	Thể hiện tính đa hình tại run time.	Thể hiện tính đa hình tại compile time.
Danh sách tham số	Danh sách tham số phải giống nhau.	Danh sách tham số khác nhau (số lượng, thứ tự, kiểu dữ liệu)
Giá trị trả về	Kiểu trả về bắt buộc phải giống nhau.	Kiểu trả về có thể khác nhau.
Phạm vi	Xảy ra giữa 2 class có quan hệ kế thừa	Xảy ra trong phạm vi cùng 1 class.

b. Phân biệt các kiểu kế thừa private, protected và public. (1đ)

Từ khóa dẫn xuất Phạm vi truy cập	Private	Protected	public
Private	X	X	X

Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con
- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, các thành phần public của lớp cha trở thành public của lớp con.
- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con.
- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con.

Câu 2.

Định nghĩa lớp Thời gian (giờ, phút giây) thuộc tính và phương thức (1đ).

Định nghĩa các phép toán ++ để tăng thời gian thêm 1 giây (1đ)

Định nghĩa toán tử >> (0.5đ) và << (0.5đ)

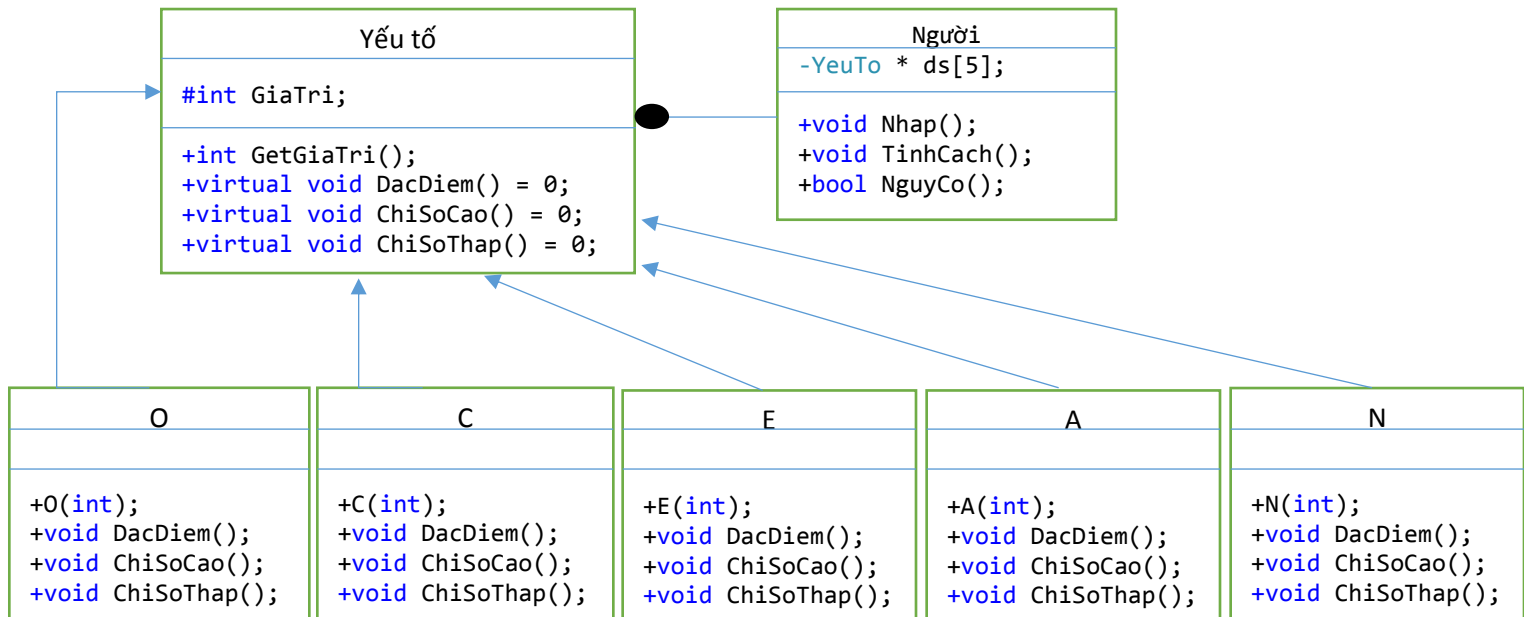
Câu 3.

Có nhiều cách để thiết kế và viết chương trình cho câu 3. Sv có sử dụng kế thừa, đa hình và thực hiện các yêu cầu của đề thi được xem là đạt yêu cầu.

- Nhập vào thông tin kết quả đánh giá tâm lý của một người. (1đ)
- Nhập vào kết quả đánh giá tâm lý của n người trong danh sách. (0.5đ)
- Chọn một người trong danh sách và cho biết các thông tin về tính cách của người đó (1đ)
- Cho biết những người có nguy cơ cao mà nhà tuyển dụng/ tổ chức/ doanh nghiệp cần lưu ý (1đ)

Dưới đây là một cách giải cho câu 3:

Sơ đồ lớp đối tượng



Chương trình

```

class YeuTo
{
protected:
    int GiaTri;
public:
    YeuTo();
    ~YeuTo();
    int GetGiaTri();
    virtual void DacDiem() = 0;
    virtual void ChiSoCao() = 0;
    virtual void ChiSoThap() = 0;
};

```

```

YeuTo::YeuTo()
{
}

```

```

YeuTo::~~YeuTo()
{
}

```

```

int YeuTo::GetGiaTri()
{
    return GiaTri;
}

```

```

class O :
    public YeuTo

```

```

{
public:
    O();
    ~O();
    O(int);
    void DacDiem();
    void ChiSoCao();
    void ChiSoThap();
};
O::O()
{
}
O::O(int nn)
{
    GiaTri = nn;
}

void O::DacDiem()
{
    cout << "San sang trai nghiem (0)";
}

O::~~O()
{
}
void O::ChiSoCao()
{
    cout << "Nguoi co chi so cao o yeu to nay thuong la nguoi thich nhung y tuong moi
...";
}

void O::ChiSoThap()
{
    cout << "Nguoi co chi so thaps o yeu to nay thuong la nguoi kha bao thu ...";
}
class C :
    public YeuTo
{
public:
    C();
    ~C();
    C(int);
    void DacDiem();
    void ChiSoCao();
    void ChiSoThap();
};
C::C()
{
}
C::C(int nn)
{
    GiaTri = nn;
}

void C::DacDiem()
{
    cout << "Tu chu tan tam (C)";
}

```

```

C::~C()
{
}

void C::ChiSoCao()
{
    cout << "Nguoi co chi so cao o yeu to nay thuong la nguoi cham chi ...";
}

void C::ChiSoThap()
{
    cout << "Nguoi co chi so thap o yeu to nay thuong la nguoi de bo cuoc ...";
}

class E :
    public YeuTo
{
public:
    E();
    ~E();
    E(int);
    void DacDiem();
    void ChiSoCao();
    void ChiSoThap();
};

E::E()
{
}

E::E(int nn)
{
    GiaTri = nn;
}

void E::DacDiem()
{
    cout << "Huong ngoai (E)";
}

E::~~E()
{
}

void E::ChiSoCao()
{
    cout << "Nguoi co chi so cao o yeu to nay thuong la nguoi nhiet tinh, nang dong
...";
}

void E::ChiSoThap()
{
    cout << "Nguoi co chi so thap o yeu to nay thuong la ngai giao tieps ...";
}

class A :
    public YeuTo
{
public:
    A();
    ~A();
    A(int);
}

```

```

        void DacDiem();
        void ChiSoCao();
        void ChiSoThap();
};
A::A()
{
}
A::A(int nn)
{
    GiaTri = nn;
}

void A::DacDiem()
{
    cout << "Hoa dong de chiu (A)";
}

A::~~A()
{
}

void A::ChiSoCao()
{
    cout << "Nguoi co chi so cao o yeu to nay thuong than thien coi mo ...";
}

void A::ChiSoThap()
{
    cout << "Nguoi co chi so thap o yeu to nay thuong dat loi ich cua ban than len
tren ...";
}
class N :
    public YeuTo
{
public:
    N();
    ~N();
    N(int);
    void DacDiem();
    void ChiSoCao();
    void ChiSoThap();
};

N::N()
{
}

N::N(int nn)
{
    GiaTri = nn;
}

void N::DacDiem()
{
    cout << "Bat on cam xuc (N)";
}

N::~~N()

```

```

{
}

void N::ChiSoCao()
{
    cout << "Nguoi co chi so cao o yeu to nay thuong co cac cam xuc tieu cuc ...";
}

void N::ChiSoThap()
{
    cout << "Nguoi co chi so thap o yeu to nay thuong kiem soat duoc cam xuc ...";
}
class Nguoi
{
private:
    YeuTo * ds[5];
public:
    Nguoi();
    ~Nguoi();
    void Nhap();
    void TinhCach();
    bool NguyCo();
};

Nguoi::Nguoi()
{
}

Nguoi::~~Nguoi()
{
}

void Nguoi::Nhap()
{
    int temp;
    cout << "Nhap vao gia tri yeu to 0";
    cin >> temp;
    ds[0] = new O(temp);
    cout << "Nhap vao gia tri yeu to C";
    cin >> temp;
    ds[1] = new C(temp);
    cout << "Nhap vao gia tri yeu to E";
    cin >> temp;
    ds[2] = new E(temp);
    cout << "Nhap vao gia tri yeu to A";
    cin >> temp;
    ds[3] = new A(temp);
    cout << "Nhap vao gia tri yeu to N";
    cin >> temp;
    ds[4] = new N(temp);
}

void Nguoi::TinhCach()
{
    for (int i = 0; i < 5; i++)
    {
        ds[i]->DacDiem();
        if (ds[i]->GetGiaTri() >= 70)
    }
}

```

```

        ds[i]->ChiSoCao();
    else
        if (ds[i]->GetGiaTri() <= 30)
            ds[i]->ChiSoThap();
        else
            cout << "Khong xac dinh ro";
    }
}

bool Nguoi::NguyCo()
{
    if ((ds[2]->GetGiaTri() <= 30) && (ds[3]->GetGiaTri() >= 70))
        return true;
    if (ds[3]->GetGiaTri() >= 70)
        return true;
    if (ds[1]->GetGiaTri() <= 30)
        return true;
    return false;
}

void main()
{
    Nguoi dsn[50];
    int n;
    cout << "Nhap so luong nguoi";
    cin >> n;
    for (int i = 0; i < n; i++)
        dsn[i].Nhap();

    int x;
    cout << "Chon nguoi thu ";
    cin >> x;
    dsn[x].TinhCach();

    for (int i = 0; i < n; i++)
        if (dsn[i].NguyCo() == true)
            cout << i << " ";
    system("pause");
}

```