

Câu 1.

a. Phân biệt các kiểu kế thừa private, protected, public (1 điểm)

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.

b. Trình bày các đặc điểm quan trọng của lập trình hướng đối tượng (1 điểm)

- Trừu tượng hóa – Abstraction Cách nhìn khái quát hóa về một tập các đối tượng có chung các đặc điểm được quan tâm (và bỏ qua những chi tiết không cần thiết).
- Đóng gói – Encapsulation Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến. Vd: các hàm/ thủ tục đóng gói các câu lệnh, các đối tượng đóng gói dữ liệu của chúng và các thủ tục có liên quan.
- Thừa kế - Inheritance cho phép một lớp D có được các thuộc tính và thao tác của lớp C, như thể các thuộc tính và thao tác đó đã được định nghĩa tại lớp D.

Cho phép cài đặt nhiều quan hệ giữa các đối tượng: Đặc biệt hóa – Tổng quát hóa

- Đa hình – Polymorphism Là cơ chế cho phép một tên thao tác hoặc thuộc tính có thể được định nghĩa tại nhiều lớp và có thể có nhiều cài đặt khác nhau tại mỗi lớp trong các lớp đó.

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 3 yêu cầu của đề thi được xem là đạt yêu cầu.

-Vẽ sơ đồ lớp đối tượng (1.5đ)

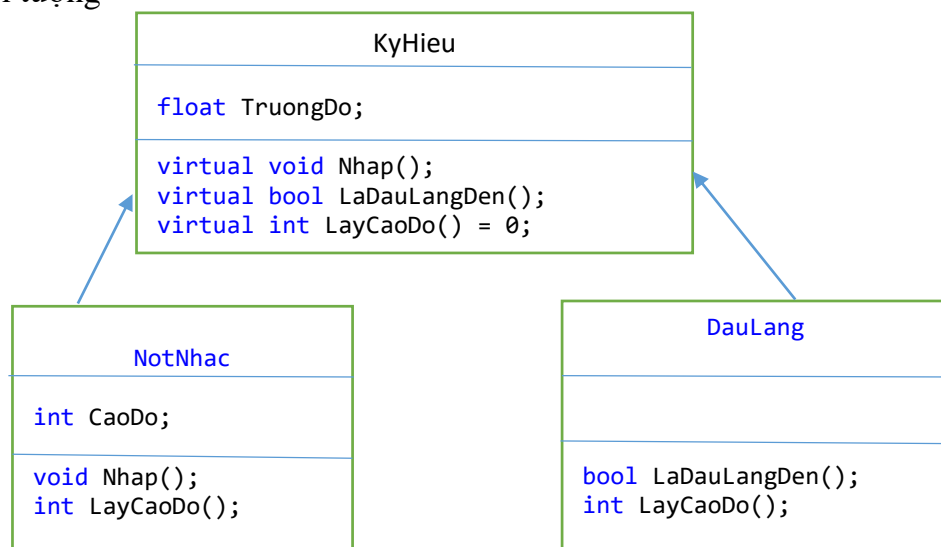
-Soạn bản nhạc (1.5đ)

-Đếm số dấu lặng đen (1đ)

-Nốt nhạc có cao độ cao nhất (1đ)

Dưới đây là một cách giải cho câu 3:

Sơ đồ lớp đối tượng



Chương trình:

```
#include <iostream>
using namespace std;
class KyHieu
{
    protected:
        float TruongDo;
    public:
        virtual void Nhap();
        virtual bool LaDauLangDen();
        virtual int LayCaoDo() = 0;
};
void KyHieu::Nhap()
{
    int t;
    cout << "Nhap gia tri truong do:";
    cout << "1.Tron 2.Trang 3.Den 4.Moc don";
    cout << "5.Moc kep 6.Moc tam 7.Moc tu";
    cin >> t;
    switch(t)
    {
        case 1: TruongDo = 4;
                break;
        case 2: TruongDo = 2;
                break;
        case 3: TruongDo = 1;
                break;
        case 4: TruongDo = 0.5;
                break;
        case 5: TruongDo = 0.25;
                break;
        case 6: TruongDo = 0.125;
                break;
        case 7: TruongDo = 0.0625;
                break;
    }
}
bool KyHieu::LaDauLangDen()
{
    return false;
}
class NotNhac :public KyHieu
{
    private:
        int CaoDo;
    public:
        void Nhap();
        int LayCaoDo();
};
void NotNhac::Nhap()
{
    //Nhap cao do
    int t;
    cout << "Nhap gia tri cao do:";
    cout << "1.Do(C) 2.Re(D) 3.Mi(E) 4.Fa(F)";
    cout << "5.Sol(G) 6.La(A) 7.Si(B)";
    cin >> t;
    CaoDo = t;
    //Nhap truong do
    KyHieu::Nhap();
}
```

```

}
int NotNhac::LayCaoDo()
{
    return CaoDo;
}
class DauLang :public KyHieu
{
    public:
        bool LaDauLangDen();
        int LayCaoDo();
};
bool DauLang::LaDauLangDen()
{
    if (TruongDo == 1)
        return true;
    return false;
}
int DauLang::LayCaoDo()
{
    return 0;
}
void main()
{
    KyHieu *BanNhac[50];
    int n;
    //cau 1. Soan ban nhac
    cout << "Nhap vao so luong cac ky hieu am nhac";
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        int t;
        cout << "Chon 1 de soan not nhac";
        cout << " va 2 de soan dau lang";
        cin >> t;
        switch (t)
        {
            case 1: BanNhac[i] = new NotNhac();
                    break;
            case 2: BanNhac[i] = new DauLang();
                    break;
        }
        BanNhac[i]->Nhap();
    }

    //cau 2. Dem dau lang den
    int count = 0;
    for (int i = 0; i < n; i++)
        if (BanNhac[i]->LaDauLangDen() == true)
            count++;
    cout << "So dau lang den la" << count;

    //cau 3.Tim not nhac co cao do cao nhat
    int max = BanNhac[0]->LayCaoDo();
    int vt = 0;
    for (int i = 1; i < n; i++)
        if (BanNhac[i]->LayCaoDo()>max)
        {
            max = BanNhac[i]->LayCaoDo();
            vt = i;
        }
    cout << "Vi tri not nhac co cao do cao nhat" << vt;
}

```