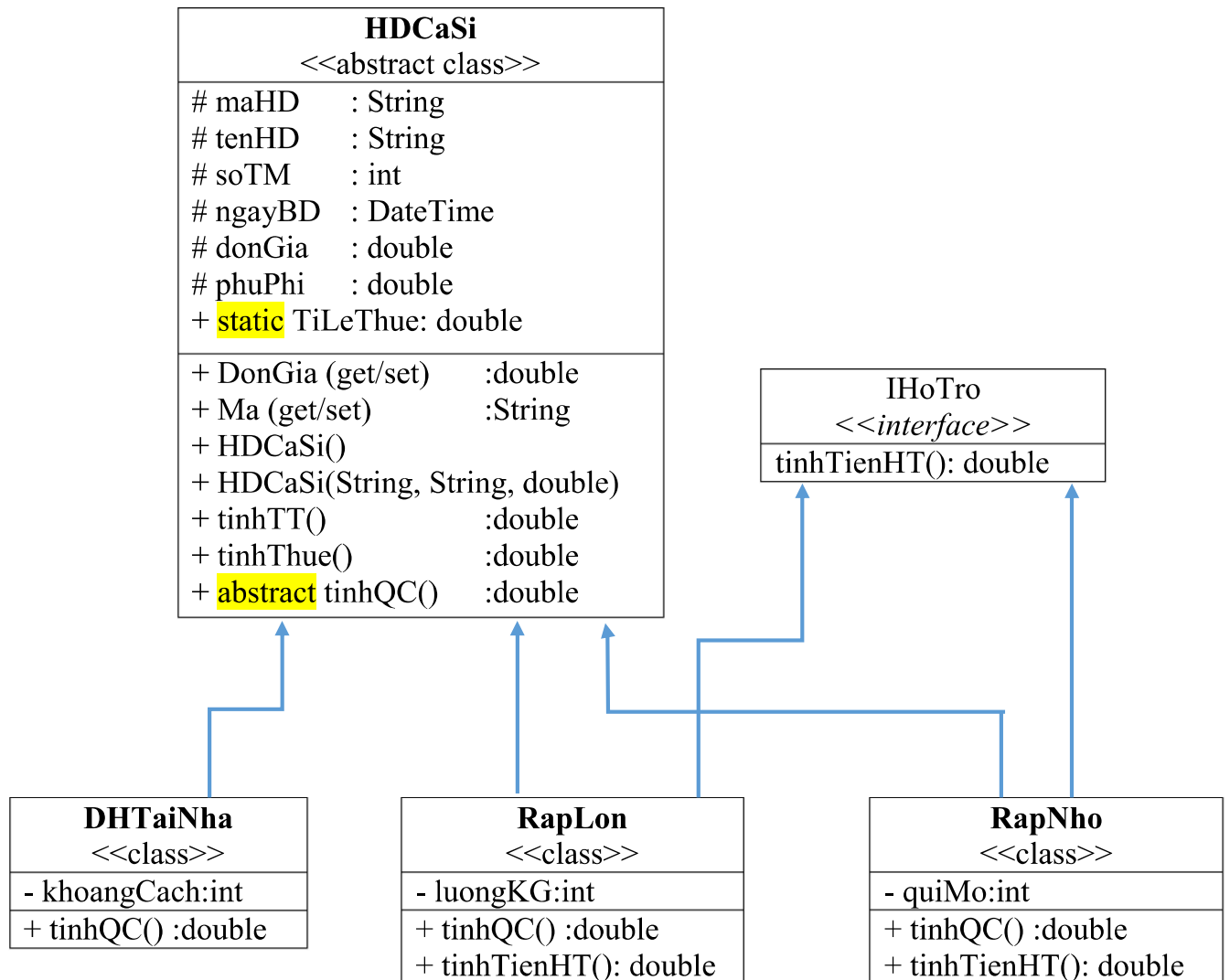


# ĐÁP ÁN ĐỀ ÔN TẬP PHẦN BÀI TẬP

## 1. Vẽ sơ đồ lớp



### Chú ý:

- Trước thuộc tính/phương thức static phải ghi thêm từ khóa static
- Dùng ký hiệu +, -, # để thay thế cho public, private, protected
- Dùng từ khóa abstract để mô tả phương thức trừu tượng
- Phương thức get/set được giải thích bằng (get/set) sau tên phương thức
- Sử dụng các ký hiệu <<abstract class>>, <<class>>, <<interface>> để biểu thị các lớp trừu tượng, lớp thông thường và interface

## 2. Cài đặt và thiết kế lớp

### Lớp HDCaSi

```
abstract public class HDCaSi
{
    protected String ma, ten;

    public String Ma
    {
        get { return ma; }
        set
        {
            ma = value;
            if (!(ma.Length == 6 && ma.StartsWith("HD") == true
&& ma.Substring(2).All(char.IsDigit)==true))
                throw new Exception("Loi roi");
        }
    }
    protected int soTM;
    protected DateTime ngayBD;
    protected double donGia, phuPhi;

    public double DonGia
    {
        get { return donGia; }
        set {
            donGia = value;
            if (donGia <= 0)
                throw new Exception("loi don gia");
        }
    }
    public static double TiLeThue = 0.15;
    public HDCaSi()
    {
        ma="HD3418";
        ten="Ly Khong Hay";
        soTM=3;
        ngayBD = new DateTime(2020,10,24);
        donGia = 4000000;
        phuPhi = 2000000;
    }
    public HDCaSi(String ma, String ten, double dg)
    {
        this.ma=ma;
        this.ten=ten;
        this.donGia=dg;
    }
}
```

```

    public HDCaSi(String ma, String ten, double dg)
    {
        this.ma=ma;
        this.ten=ten;
        this.donGia=dg;
        this.soTM = 2;
        phuPhi=0;
        ngayBD=new DateTime(2020,12,24);
    }

    public double tinhTT()
    {
        return soTM * donGia + phuPhi + tinhQC();
    }
    public abstract double tinhQC();
    public double tinhThue()
    {
        return tinhTT() * TiLeThue;
    }
}

```

## Interface

```

public interface IHoTro
{
    double tinhTienHT();
}

```

## Lớp HDTaiNha

```

public class HDTaiNha:HDCaSi
{
    int khoangCach;
    public override double tinhQC()
    {
        if (khoangCach <= 20)
            return 400000 * khoangCach;
        else
        {
            return Math.Min(khoangCach * 300000, 15000000);
        }
    }
}

```

## Lop **RapNho**

```
public class RapNho:HDCaSi,IHoTro
{
    int quiMo;
    public override double tinhQC()
    {
        return 10000000 + 2000000 * soTM;
    }

    public double tinhTienHT()
    {
        if (quiMo >= 1 && quiMo <= 3)
            return 1000000;
        return 0;
    }
}
```

## Lop **RapLon**

```
public class RapLon:HDCaSi,IHoTro
{
    int luongKG;
    public override double tinhQC()
    {
        if (luongKG < 1000000)
            return 30000000;
        else
            return 50000000;
    }

    public double tinhTienHT()
    {
        return Math.Min(luongKG / 100000 * 1000000, 200000000);
    }
}
```