```
public class ChildrensPrice extends Price
{
    public ChildrensPrice() {
        super(3, 1.5, 1.5);
    }

    public int getPriceCode()
    {
        return Movie.CHILDRENS;
    }
}
```



```
import java.awt.*;
import java.awt.event.*;
import java.lang.*;
import java.util.*;

public class Customers
{
    private String _name;
    private Vector _rentals = new Vector();
    public Customers(String name)
    {
        _name=name;
    }
    public void addRental(Rental rental)
    {
        _rentals.addElement(rental);
    }
    public String getName()
    {
        return _name;
    }

    public String statement()
    {
        Enumeration rentals=_rentals.elements();
        String result = "Rental Record for "+getName()+"\n";
        while (rentals.hasMoreElements())
            {
                Rental each=(Rental) rentals.nextElement();
                result +="\t" + each.getMovie().getTitle()+"\t"+
                    String.valueOf(each.getCharge()) +" \n";
            }
        result += "Amount owned is " + String.valueOf(getTotalCharge()) +
            "\n";
        result += "You earned " + String.valueOf(getTotalFrequentRenterPoints()) +
            " frequent renter points";
        return result;
    }

    public String htmlStatement()
    {
        Enumeration rentals=_rentals.elements();
        String result = "<H1>Rental Record for "+getName()+"</EM></H1><P>\n";
        while (rentals.hasMoreElements())
            {
                Rental each=(Rental) rentals.nextElement();
                result +=each.getMovie().getTitle()+": "+
                    String.valueOf(each.getCharge()) +"<BR>\n";
            }
        result += "<P>Amount owned is <EM>" + String.valueOf(getTotalCharge()) +
            "</EM><P>\n";
        result += "You earned <EM>" + String.valueOf(getTotalFrequentRenterPoints(
)) +
            " </EM>frequent renter points<P>";
        return result;
    }


    private double getTotalCharge()
    {
        double result=0;
        Enumeration rentals=_rentals.elements();
```

```
        while (rentals.hasMoreElements())
            {
                Rental each=(Rental) rentals.nextElement();
                result+=each.getCharge();
            }
        return result;
    }


    private int getTotalFrequentRenterPoints()
    {
        int result=0;
        Enumeration rentals=_rentals.elements();
        while (rentals.hasMoreElements())
            {
                Rental each=(Rental) rentals.nextElement();
                result+=each.getFrequentRenterPoints();
            }
        return result;
    }
}
```



```
public class Movie
{
    public static final int CHILDRENS = 2;
    public static final int REGULAR = 0;
    public static final int NEW_RELEASE = 1;

    private String _title;
    private Price _price;

    public Movie(String title,int priceCode)
    {
        _title=title;
        setPriceCode(priceCode);
    }
    public int getPriceCode()
    {
        return _price.getPriceCode();
    }
    public void setPriceCode(int priceCode)
    {
        switch (priceCode)
            {
            case Movie.REGULAR:
                _price=new RegularPrice();
                break;
            case Movie.NEW_RELEASE:
                _price=new NewReleasePrice();
                break;
            case Movie.CHILDRENS:
                _price=new ChildrensPrice();
                break;
            default:
                throw new IllegalArgumentException("Incorrect Price Code");
            }
    }
    public String getTitle()
    {
        return _title;
    }

    public int getFrequentRenterPoints(int daysRented)
    {
        return _price.getFrequentRenterPoints(daysRented);

    }

    public double getCharge(int daysRented)
    {
        return _price.getCharge(daysRented);
    }



}
```

```
public class NewReleasePrice extends Price
{
    public NewReleasePrice() {
        super(0, 0., 3.);
    }

    public int getPriceCode()
    {
        return Movie.NEW_RELEASE;
    }

    public int getFrequentRenterPoints(int daysRented)
    {
        return (daysRented > 1) ? 2 : 1;
    }
}
```



```
abstract class Price
{
    int nDays;
    double minPrice;
    double dayPrice;

    Price(int nDays, double minPrice, double dayPrice) {
        this.nDays = nDays;
        this.minPrice = minPrice;
        this.dayPrice = dayPrice;
    }

    abstract int getPriceCode();
    public double getCharge(int daysRented) {
        return minPrice
            + (daysRented > nDays ? (daysRented - nDays) * dayPrice : 0.);
    }

    public int getFrequentRenterPoints(int daysRented)
    {
        return 1;
    }
}
```

Oct 07, 04 14:59 **RegularPrice.java** Page 1/1

```
public class RegularPrice extends Price
{
    public RegularPrice() {
        super(2, 2., 1.5);
    }

    public int getPriceCode()
    {
        return Movie.REGULAR;
    }
}
```

Oct 13, 03 15:59 **Rental.java** Page 1/1

```
public class Rental
{
    private Movie _movie;
    private int _daysRented;

    public Rental(Movie movie,int daysRented)
    {
        _movie=movie;
        _daysRented=daysRented;
    }
    public int getDaysRented()
    {
        return _daysRented;
    }
    public Movie getMovie()
    {
        return _movie;
    }
    public double getCharge()
    {
        return _movie.getCharge(_daysRented);
    }

    public int getFrequentRenterPoints()
    {
        return _movie.getFrequentRenterPoints(_daysRented);

    }
}
```

```
import java.lang.*;
import java.util.*;

public class TestProg
{
    public static void main( String[] argv)
    {
        Movie m1=new Movie(new String("Xmen    "),Movie.NEW_RELEASE);
        Movie m2=new Movie(new String("Gladiators"),Movie.REGULAR);
        Movie m3=new Movie(new String("Mulan   "),Movie.CHILDRENS);

        Customers c1=new Customers(new String("Rémi"));
        Customers c2=new Customers(new String("Sebastien"));
        Customers c3=new Customers(new String("Laurent"));

        c1.addRental(new Rental(m1,4));
        c1.addRental(new Rental(m2,1));
        c1.addRental(new Rental(m3,7));

        c2.addRental(new Rental(m1,1));
        c2.addRental(new Rental(m2,2));

        c3.addRental(new Rental(m1,8));
        c3.addRental(new Rental(m2,10));
        c3.addRental(new Rental(m3,1));

        System.out.println(c1.statement());
        System.out.println(c2.statement());
        System.out.println(c3.statement());
    }
}
```