```
public class ChildrensPrice extends Price
{
    public int getPriceCode()
    {
        return Movie.CHILDRENS;
    }
    public double getCharge(int daysRented)
    {
        double result = 1.5;
        if(daysRented > 3)
            result+= (daysRented-3)*1.5;
        return result;
    }
}
```



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

public class Customers
{
    private StatementBuilder statementBuilder = new TextBuilder();
    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 void setStatementBuilder(StatementBuilder b)
    {
        statementBuilder = b;
    }

    public String statement()
    {
        Enumeration rentals=_rentals.elements();

        statementBuilder.reset();
        statementBuilder.setTitle( "Rental Record for "+getName());
        while (rentals.hasMoreElements())
            {
                Rental each=(Rental) rentals.nextElement();
                statementBuilder.setTabulation();
                statementBuilder.addText(each.getMovie().getTitle());
                statementBuilder.setTabulation();
                statementBuilder.addLine(String.valueOf(each.getCharge()));
            }
        statementBuilder.beginParagraph("Amount owned is " + String.valueOf(getTota
lCharge()));
        statementBuilder.beginParagraph("You earned " + String.valueOf(getTotalFre
quentRenterPoints()) +
            " frequent renter points");
        statementBuilder.endParagraph();
        return statementBuilder.getText();
    }

    private double getTotalCharge()
    {
        double result=0;
        Enumeration rentals=_rentals.elements();
        while (rentals.hasMoreElements())
            {
                Rental each=(Rental) rentals.nextElement();
                result+=each.getCharge();
            }
        return result;
    }
```

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



```java
public class HtmlBuilder implements StatementBuilder
{
    private String allText = "";

    public void reset()
    {
        allText = "";
    }
    public void setTitle(String s)
    {
        allText += "<H1>" +  s +  "</H1><P>\n" ;
    }
    public void beginParagraph(String s)
    {
        allText += "<P>" + s + "\n";

    }
    public void beginParagraph()
    {
        allText +=  "<P>";
    }
    public void endParagraph()
    {
        allText +=  "\n";
    }

    public void setTabulation()
    {
        allText += "\t";
    }
    public void addLine(String s)
    {
        allText += s +   "\n";
    }
    public void addText(String s)
    {
        allText += s;
    }

    public String getText()
    {
        return allText;
    }

}
```

```java
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);
    }



}
```



```java
public class NewReleasePrice extends Price
{
    public int getPriceCode()
    {
        return Movie.NEW_RELEASE;
    }
    public double getCharge(int daysRented)
    {
        return daysRented*3;

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

```java
abstract class Price
{
    abstract int getPriceCode();
    abstract double getCharge(int daysRented);
    public int getFrequentRenterPoints(int daysRented)
    {
        return 1;
    }
}
```



```java
public class RegularPrice extends Price
{
    public int getPriceCode()
    {
        return Movie.REGULAR;
    }
    public double getCharge(int daysRented)
    {
        double result=2;
        if (daysRented>2)
            {
                result+=(daysRented-2)*1.5;
            }
        return result;
    }
}
```

```java
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);

    }
}
```



```java
public interface StatementBuilder
{

    public void reset();
    public void setTitle(String s);
    public void beginParagraph(String s);
    public void beginParagraph();
    public void endParagraph();
    public void setTabulation();
    public void addLine(String s);
    public void addText(String s);
    public String getText();


}
```

```java
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());

        HtmlBuilder  htmlBuilder = new HtmlBuilder();

        c1.setStatementBuilder(htmlBuilder);
        c2.setStatementBuilder(htmlBuilder);
        c3.setStatementBuilder(htmlBuilder);

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

    }
}
```



```java
public class TextBuilder implements StatementBuilder
{
    private String allText = "";

    public void reset()
    {
        allText = "";
    }
    public void setTitle(String s)
    {
        allText += s +  "\n" ;
    }
    public void beginParagraph(String s)
    {
        allText += "\n" + s;

    }
    public void beginParagraph()
    {
        allText +=  "\n";
    }
    public void endParagraph()
    {
        allText +=  "\n";
    }

    public void setTabulation()
    {
        allText += "\t";
    }
    public void addLine(String s)
    {
        allText += s +   "\n";
    }
    public void addText(String s)
    {
        allText += s;
    }

    public String getText()
    {
        return allText;
    }

}
```