

```

using System;
using System.Collections.Generic;

namespace Refactoring
{
    /// <summary>
    /// Класс, представляющий клиента магазина.
    /// </summary>
    class Customer
    {
        // Поля

        private string name = null;
        private List<Rental> rentals = new List<Rental>();

        // Методы

        public Customer(string name)
        {
            this.name = name;
        }

        // TODO: Метод Statement() - слишком громоздкий.

        /// <summary>
        /// Метод создающий отчет.
        /// </summary>
        /// <returns></returns>
        public string Statement()
        {
            string result = string.Format("Учет аренды для {0}: ", name);

            foreach (Rental rental in rentals)
            {
                result += "\t" + rental.Movie.Title + "\t" + rental.GetThisAmount() + "\n";
            }

            result += "Сумма задолженности составляет " + GetTotalAmount() + "\n";
            result += "Вы заработали " + GetFrequentRenterPoints() + " очков за активность";
            return result;
        }

        double GetTotalAmount()
        {
            double res = 0;

            foreach (Rental rental in rentals)
            {
                res += rental.GetThisAmount();
            }

            return res;
        }

        int GetFrequentRenterPoints()
        {
            int res = 0;

            foreach (Rental rental in rentals)
            {
                res += rental.GetFrequentRenterPoints();
            }
        }
    }
}

```

```

        }

        return res;
    }

    public Rental Rentals
    {
        set { this.rentals.Add(value); }
    }

    public string Name
    {
        get { return this.name; }
    }
}
}

```

```

using System;
using Ref3;

namespace Refactoring
{
    /// <summary>
    /// Класс, который предоставляет данные о фильме.
    /// </summary>
    class Movie
    {
        private string title = null;
        Price price;

        public Movie(string title, Price price)
        {
            this.title = title;
            this.price = price;
        }

        public Price Price
        {
            get { return price; }
            set { price = value; }
        }

        public string Title
        {
            get { return this.title; }
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Ref3
{
    public abstract class Price
    {

```

```

        public abstract double GetThisAmount(int daysRented);
        public abstract int GetFrequentRenterPoints(int daysRented);
    }

    public class PriceRegular : Price
    {
        public override double GetThisAmount(int daysRented)
        {
            double result = 2;
            if (daysRented > 2)
                result += (daysRented - 2) * 1.5;

            return result;
        }

        public override int GetFrequentRenterPoints(int daysRented)
        {
            int res = 1;

            return res;
        }
    }

    public class PriceNewRelease : Price
    {
        public override double GetThisAmount(int daysRented)
        {
            double result = daysRented * 3;

            return result;
        }

        public override int GetFrequentRenterPoints(int daysRented)
        {
            int res = 1;

            if (daysRented > 1)
                res++;

            return res;
        }
    }

    public class PriceChildrens : Price
    {
        public override double GetThisAmount(int daysRented)
        {
            double result = 1.5;

            if (daysRented > 3)
                result += (daysRented - 3) * 1.5;

            return result;
        }

        public override int GetFrequentRenterPoints(int daysRented)
        {
            int res = 1;

            return res;
        }
    }

```

```
    }  
}
```

```
using System;  
using Ref3;
```

```
namespace Refactoring  
{
```

```
    /// <summary>  
    /// Класс, представляющий данные о прокате фильма.  
    /// </summary>
```

```
    class Rental  
    {
```

```
        // Поля
```

```
        private Movie movie = null;  
        private int daysRented = 0;
```

```
        // Методы
```

```
        public Rental(Movie movie, int daysRented)  
        {  
            this.movie = movie;  
            this.daysRented = daysRented;  
        }
```

```
        // Свойства
```

```
        public Movie Movie  
        {  
            get { return this.movie; }  
        }
```

```
        public int DaysRented  
        {  
            get { return this.daysRented; }  
        }
```

```
        public double GetThisAmount()  
        {  
            return Movie.Price.GetThisAmount(DaysRented);  
        }
```

```
        public int GetFrequentRenterPoints()  
        {  
            return Movie.Price.GetFrequentRenterPoints(DaysRented);  
        }
```

```
    }  
}
```

```

using System;
using Ref3;

// TODO: Слабо спроектированная программа. (Не объектно-ориентированная)

// Программа рассчитывает и выводит отчет об оплате клиентом услуг в магазине видеопроката.
// Программе сообщается, какие фильмы брал в прокате клиент и на какой срок.
// После этого программа рассчитывает сумму платежа исходя из продолжительности проката и
// типа фильма.
// Фильмы бывают трех типов: обычные, детские и новинки. Помимо расчета суммы оплаты
// начисляются бонусы
// в зависимости от того, является ли фильм новым.

namespace Refactoring
{
    class Program
    {
        static void Main()
        {
            Customer customer = new Customer("Alex");
            Movie movie = new Movie("Matrix", new PriceNewRelease());
            Rental rental = new Rental(movie, 2);

            customer.Rentals = rental;

            Movie movie2 = new Movie("Shrek", new PriceChildrens());
            Rental rental2 = new Rental(movie2, 2);

            customer.Rentals = rental2;

            Console.WriteLine(customer.Statement());

            // Delay.
            Console.ReadKey();
        }
    }
}

```