**The exercise is customized from the Video Store Example  
Source: Advanced Programming Concepts 2014**

**Set up:**

Video Store is a program that calculates and prints a statement of a customer’s charges at a video store. The program is told which movies a customer rented and for how long. It then calculates the charges, which depend on how long the movie is rented, and identifies the types of movies. There are three kinds of movies: regular, children’s, and new releases. In addition to calculating charges, the statement also computes frequent renter points, which vary depending on whether the film is a new release.

You do not have to develop from scratch. An existing codebase is made for the program (in this folder). Below is the class diagram and sequence diagram of the program:

Diagram

Description automatically generated

Diagram

Description automatically generated

**Task 1: implement a new function htmlstatement()**

The current statement() method prints out a plain string. Now, we want you to implement a **htmlstatement()** method to print in JSON format (<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON#:~:text=JavaScript%20Object%20Notation%20(JSON)%20is,page%2C%20or%20vice%20versa>), so that the statement can be sent further to an external API (micro service)

*Tips: Consider the impact this change would have. As you look at the code you can see that it is impossible to reuse any of the behavior of the current statement method for an HTML statement. Now, you can just copy the statement method and make whatever changes you need.*

*But what happens when the charging rules change? You have to fix both statement and htmlStatement and ensure the fixes are consistent…*

Task 2: Decompose a long method

Long methods (A method contains too many lines of code) is a code smell. When we notice a long method like that, we want to decompose it into smaller pieces. Smaller pieces of code tend to make things more manageable. Now we want you to rewrite statement() methods (including your **htmlstatement())** and create an additional method called amountFor(), as shown in the UML diagram below:

Diagram

Description automatically generated

**Task 3: Renaming variables**

Find variable, methods that are difficult to understand or confusing. And change the variable names!

**Task 4: Moving methods**

amountFor uses information from the rental, but it does not use information from the customer. This is a smell of Feature Envy (A method accesses the data of another object more than its own data). We would like you to move amountFor() from Customer to Rental class, rename it to getCharge() and do any updating which is necessary. The UML diagram of the program should look like this for Task 4:

Diagram

Description automatically generated

**References:**

1. <https://www.ge.infn.it/geant4/training/APC2014/exercise1.html>
2. M. Fowler, Refactoring: Improving the Design of Existing Code, Addison-Wesley Professional, 1999
3. M. Feathers, Working Effectively with Legacy Code, Prentice Hall, 2004
4. S. Demeyer, S. Ducasse, Oscar Nierstrasz, Object-Oriented Reengineering Patterns, 2009
5. Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008