Emmanuel Giah

ECG180001

October 27, 2021

CS 3354

Tien Nguyen

REFACTORING TO DESIGN PATTERNS

In this homework, you are required to use the code for the original movie rental source code and add the following features:

1. You are required to modify the code to make the function of computing frequent rental points into a strategy design pattern (That is, you will have strategy classes to compute frequent rental points).

//compute frequent renter points

public int checkRenterPoints(int points, int threshold) {

RentalPointsContext bonus;

if ((\_movie.getPriceCode() == Movie.NEW\_RELEASE) &&

(\_daysRented > threshold)) {

bonus = new RentalPointsContext(new ComputeNewRentalPoints());

points = bonus.executeRentalPointsStrategy(points);

} else {

bonus = new RentalPointsContext(new ComputeOldRentalPoints());

points = bonus.executeRentalPointsStrategy(points);

}

return points;

}

1. You are required to modify the code to make the function of computing rental prices into a strategy design pattern (That is, you will have strategy classes to compute rental prices).

public double computeAmount(double thisAmount) {

RentalPricesContext price;

switch (\_movie.getPriceCode()) {

case Movie.REGULAR:

price = new RentalPricesContext(new ComputeRegularMoviePrice());

thisAmount = price.executeRentalPriceStrategy(thisAmount, \_daysRented);

break;

case Movie.NEW\_RELEASE:

price = new RentalPricesContext(new ComputeNewMoviePrice());

thisAmount = price.executeRentalPriceStrategy(thisAmount, \_daysRented);

break;

case Movie.CHILDRENS:

price = new RentalPricesContext(new ComputeKidMoviePrice());

thisAmount = price.executeRentalPriceStrategy(thisAmount, \_daysRented);

break;

}

return thisAmount;

}