**Centennial College**

**COMP 228: Java Programming**

**LAB #3 – Using Inheritance and Polymorphism**

**Studen**t: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Due Date: **Week 7**.

Purpose: The purpose of this Lab assignment is to:

1. Practice the use of Inheritance
2. Practice the use of Polymorphism.

References: Learning materials for weeks 5 and 6, textbook, and other references (if any)

Be sure to read the following general instructions carefully:

This lab should be completed individually by all the students.

YOU NEED TO SUBMIT THE FOLLOWING 2 DOCUMENTS IN THE DROPBOX TITLED LAB3:

1. THE FIRST ONE IS A WORD DOCUMENT. USE THIS DOCUMENT AND ADD SCREEN SHOTS OF THE RUNNING STATE OF EACH EXERCISE (If there are more than 1 exercise). DO NOT DELETE THE QUESTIONS. THE SCREEN SHOTS SHOULD FOLLOW EACH QUESTION AND COVER ALL THE ASPECTS/FUNCTIONALITIES OF EACH EXERCISE. AFTER THE SCREEN SHOTS PLEASE COPY THE CODE FROM THE CODE WINDOW AND PASTE THE COMPLETE CODE. DO NOT GIVE ME SCREEN SHOTS OF THE CODE. DO NOT ZIP THIS FILE AND KEEP IT SEPARATE FROM YOUR ZIPPED PROGAM FILE.

2. SUBMIT ALSO ONE ZIPPED PROJECT FILE THAT CONTAINS ALL THE EXERISES SEPARATELY INTO THE SAME DROP BOX.

This material provides the necessary information you need to complete the exercises.

You must name your Eclipse project according to the following rule:

**YourFullName\_COMP228Labnumber**

Example: **JohSmith\_COMP228Lab3**

Each exercise should be placed in a separate package named *exercise1*, *exercise2*, etc.

Submit your assignment in a **zip file** that is named according to the following rule:

**YourLastName\_COMP228Labnumber.zip**

Example: **JohSmith\_COMP228Lab3.zip**

Apply the naming conventions for variables, methods, classes, and packages:

- *variable names* start with a *lowercase* character

- *classes* start with an *uppercase* character

- **packages** use only *lowercase* characters

- *methods* start with a *lowercase* character

## Exercise 1

Write a Java application that implements different types of insurance policies for employees of an organization.

Let **Insurance** be an abstract superclass and **Health** and **Life** two of its subclasses that describe respectively health insurance and life insurance.

The **Insurance** class defines an instance variable of type **String** to describe the **type of insurance** and an instance variable of type **double** to hold the **monthly cost** of that insurance.

Implement the **get** methods for both variables of class **Insurance**. Declare also two **abstract** methods named **setInsuranceCost()** and **displayInfo()** for this class.

The **Life** and **Health** class should implement **setInsuranceCost** and **display** methods by setting the appropriate monthly fee and display the information for each insurance type.

Write a driver class to test this hierarchy. This application should ask the user to enter the type of insurance and its monthly fee. Then, will create the appropriate object (Life or Health) and display the insurance information.

As you create each insurance object, place an **Insurance** reference to each new **Insurance** object into an array. Each class has its own **setInsuranceCost** method. Write **a polymorphic** screen manager that walks through the array sending **setInsuranceCost** messages to each object in the array and displaying this information on the screen.

(3 marks)

**Exercise #2:**

Create an abstract class called GameTester. The GameTester class includes a name for the game tester and a boolean value representing the status (full-time, part-time).

Include an abstract method to determine the salary, with full-time game testers getting a base salary of $3000 and part-time game testers getting $20 per hour.

Create two subclasses called FullTimeGameTester, PartTimeGameTester. Create a console application that demonstrates how to create objects of both subclasses. Allow the user to choose game tester type and enter the number of hours for the part-time testers.

(3 marks)

**Exercise #3:**

CityToronto bank provides mortgages for individuals and businesses up to $300,000. Write a Java application that keeps track of mortgages and computes the total amount owed at any time (mortgage amount + interest).

Design the following classes to implement your application:

**Mortgage** – an abstract class that implements the *MortgageConstants* interface. A Mortgage includes a mortgage number, customer name, amount of mortgage, interest rate, and term.

Don’t allow mortgage amounts over $300,000. Force any mortgage term that is not defined in the *MortgageConstants* interface to a short-term, one year loan. Create a *getMortgageInfo* method to display all the mortgage data.

**MortgageConstants** – includes constant values for *short-term* (one year), *medium-term* (three years) and *long-term* (5 years) mortgages. It also contains constants for bank name and the maximum mortgage amount.

**BusinessMortgage** – extends Mortgage. Its constructor sets the interest rate to 1% over the current prime rate.

**PersonalMortgage** - extends Mortgage. Its constructor sets the interest rate to 2% over the current prime rate.

**ProcessMortgage** – a main class that create an array of 3 mortgages. Prompt the user for the current interest rate. Then in a loop prompts the user for a mortgage type and all relevant information for that mortgage. Store the created Mortgage objects in the array. When data entry is complete, display all mortgages.

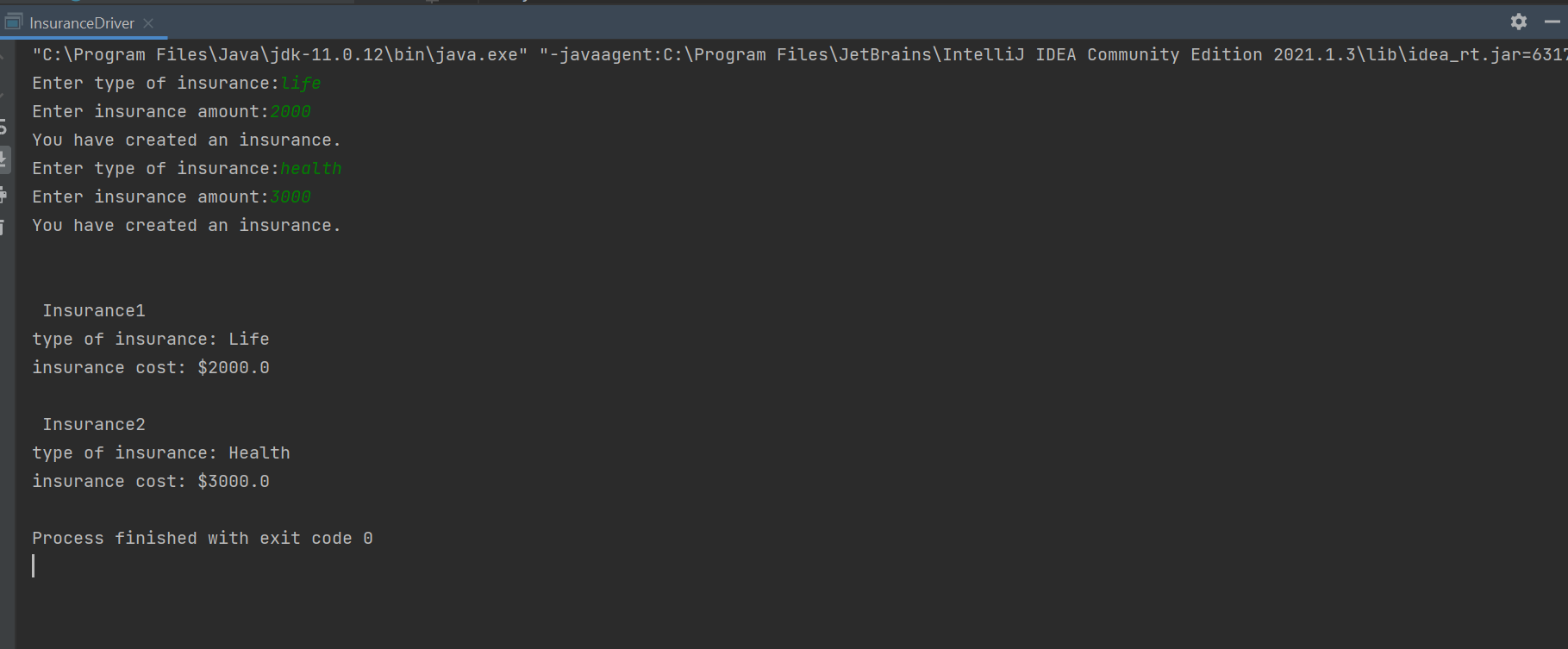
(4 marks)

**Evaluation:**

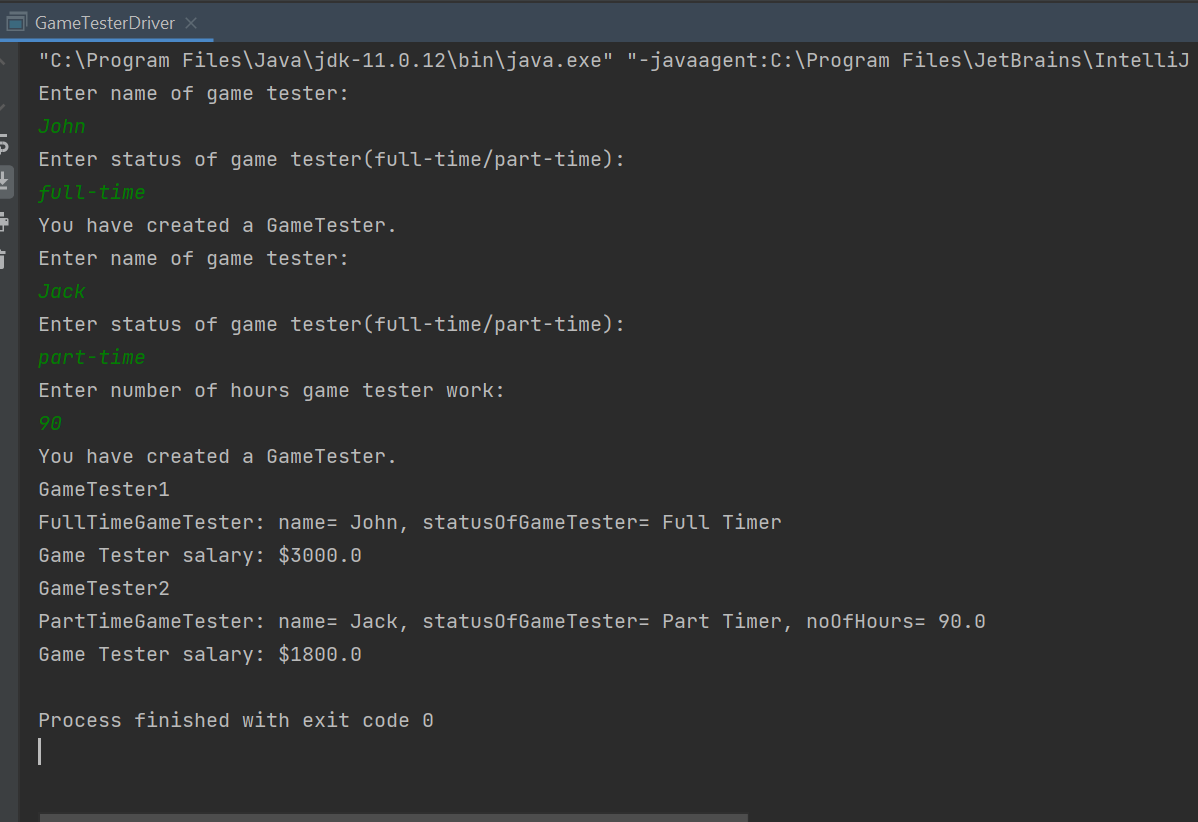
|  |  |
| --- | --- |
| **Functionality** |  |
| Correct implementation of classes (instance variable declarations, constructors, getter and setter methods, etc.) | 40% |
| Correct implementation of driver classes (declaring and creating objects, calling their methods, interacting with user, displaying results) | 40% |
| Comments, correct naming of variables, methods, classes, etc. | 5% |
| **Friendly input/output** | 15% |
| **Total** | 100% |

**Outputs:**

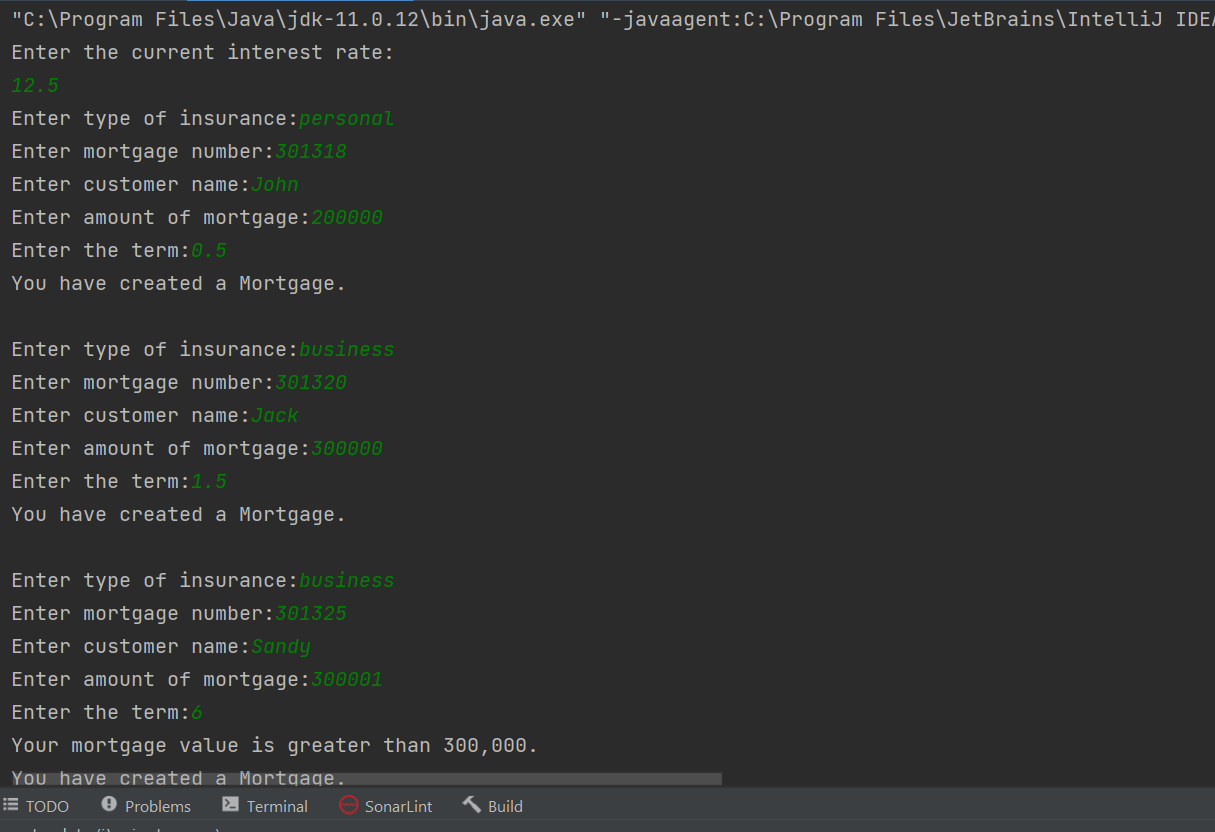
**Exercise1:**

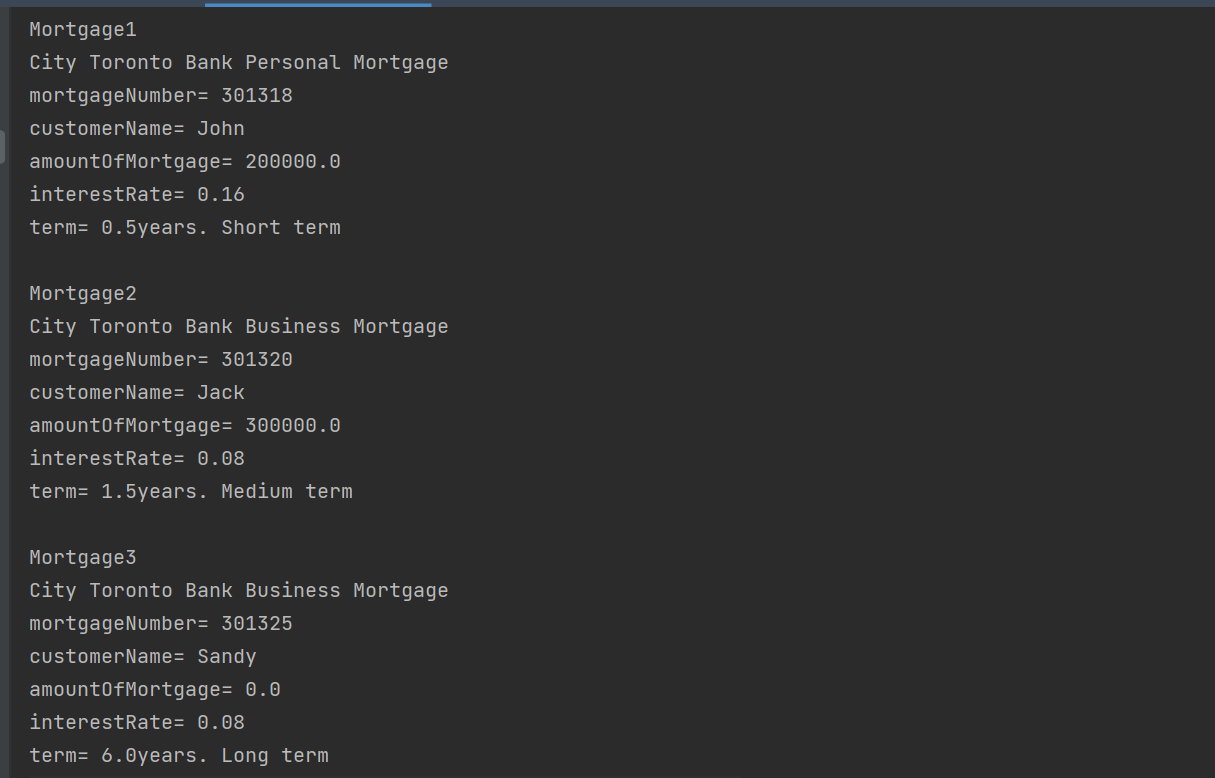
****

**Exercise2:**

****

**Exercise3:**

****

****

**Code:**

**Exercise1:**

**Insurance.java**

package exercise1;  
  
public abstract class Insurance {  
 protected String typeOfInsurance;  
 protected double insuranceCost;  
  
 public String getTypeOfInsurance() {  
 return typeOfInsurance;  
 }  
  
 public double getMonthlyCost() {  
 return insuranceCost;  
 }  
  
 public abstract void displayInfo();  
 public abstract void setInsuranceCost(double insuranceCost);  
}

**Life.java**

package exercise1;  
  
public class Life extends Insurance{  
 Life(){  
 this.typeOfInsurance = "Life";  
 this.insuranceCost = insuranceCost;  
 }  
 @Override  
 public void displayInfo() {  
 System.*out*.println("type of insurance: "+typeOfInsurance+"\n"+"insurance cost: $"+insuranceCost);  
 }  
  
 @Override  
 public void setInsuranceCost(double insuranceCost) {  
 this.insuranceCost = insuranceCost;  
 }  
}

**Health.java**

package exercise1;  
  
public class Health extends Insurance{  
 Health(){  
 this.typeOfInsurance = "Health";  
 }  
 @Override  
 public void displayInfo() {  
 System.*out*.println("type of insurance: "+typeOfInsurance+"\n"+"insurance cost: $"+insuranceCost);  
 }  
  
 @Override  
 public void setInsuranceCost(double insuranceCost) {  
 this.insuranceCost = insuranceCost;  
 }  
}

**InsuranceDriver.java**

package exercise1;  
  
import java.util.Scanner;  
  
public class InsuranceDriver {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 Insurance[] insurances = new Insurance[3];  
  
 for(int i = 0; i < 2; i++){  
 System.*out*.print("Enter type of insurance:");  
 String typeOfInsurance = scanner.next();  
 System.*out*.print("Enter insurance amount:");  
 double insuranceCost = scanner.nextDouble();  
 if(typeOfInsurance.equals("life")){  
 insurances[i] = new Life();  
 insurances[i].setInsuranceCost(insuranceCost);  
 System.*out*.println("You have created an insurance.");  
 }  
 else if(typeOfInsurance.equals("health")){  
 insurances[i] = new Health();  
 insurances[i].setInsuranceCost(insuranceCost);  
 System.*out*.println("You have created an insurance.\n");  
 }  
 else{  
 i--;  
 System.*out*.println("You entered incorrect insurance type. Try Again.");  
 }  
 }  
 for(int i = 0; i < 2; i++){  
 System.*out*.println("\n Insurance"+(i+1));  
 insurances[i].displayInfo();  
 }  
 }  
}

**Exercise2:**

**GameTester.java**

package exercise2;  
  
public abstract class GameTester {  
 String name;  
 boolean statusOfGameTester;  
 // true = fulltime and false = partime  
 public abstract double determineSalary();  
}

**FullTimeGameTester.java**

package exercise2;  
  
public class FullTimeGameTester extends GameTester{  
 FullTimeGameTester(String name, boolean statusOfGameTester){  
 this.name = name;  
 this.statusOfGameTester = statusOfGameTester;  
 }  
  
 @Override  
 public double determineSalary() {  
 return 3000;  
 }  
  
 @Override  
 public String toString() {  
 return "FullTimeGameTester: " +  
 "name= " + name +  
 ", statusOfGameTester= " + (statusOfGameTester? "Full Timer": "Part Timer");  
 }  
}

**PartTimeGameTester.java**

package exercise2;  
  
public class PartTimeGameTester extends GameTester{  
 double noOfHours;  
 PartTimeGameTester(String name, boolean statusOfGameTester, double noOfHours){  
 this.name = name;  
 this.statusOfGameTester = statusOfGameTester;  
 this.noOfHours = noOfHours;  
 }  
  
 @Override  
 public double determineSalary() {  
 return noOfHours\*20;  
 }  
  
 @Override  
 public String toString() {  
 return "PartTimeGameTester: " +  
 "name= " + name +  
 ", statusOfGameTester= " + (statusOfGameTester? "Full Timer": "Part Timer") +  
 ", noOfHours= " + noOfHours;  
 }  
}

**GameTesterDriver.java**

package exercise2;  
  
import java.util.Scanner;  
  
public class GameTesterDriver {  
 public static void main(String[] args){  
 GameTester[] gameTesters = new GameTester[3];  
 Scanner scanner = new Scanner(System.*in*);  
 for(int i = 0 ; i < 2; i++){  
 System.*out*.println("Enter name of game tester:");  
 String name = scanner.next();  
 System.*out*.println("Enter status of game tester(full-time/part-time):");  
 String status = scanner.next();  
 if(status.equals("full-time")){  
 gameTesters[i] = new FullTimeGameTester(name, true);  
 System.*out*.println("You have created a GameTester.");  
 }  
 else if(status.equals("part-time")){  
 System.*out*.println("Enter number of hours game tester work:");  
 double hrs = scanner.nextDouble();  
 gameTesters[i] = new PartTimeGameTester(name, false, hrs);  
 System.*out*.println("You have created a GameTester.");  
 }  
 else{  
 System.*out*.println("wrong status of gamer. Try again.");  
 i--;  
 }  
 }  
 for(int i = 0 ; i < 2; i++){  
 System.*out*.println("GameTester"+(i+1));  
 System.*out*.println(gameTesters[i].toString());  
 System.*out*.println("Game Tester salary: $"+ gameTesters[i].determineSalary());  
 }  
 }  
}

**Exercise3:**

**MortgageConstants.java**

package exercise3;  
  
public interface MortgageConstants {  
 final int *shortTerm* = 1;  
 final int *mediumTerm* = 3;  
 final int *longTerm* = 5;  
 final String *bankName* = "City Toronto Bank";  
 final double *maximumMortgageAmount* = 300000;  
}

**Mortgage.java**

package exercise3;  
  
public abstract class Mortgage implements MortgageConstants{  
 int mortgageNumber;  
 String customerName;  
 double amountOfMortgage;  
 double interestRate;  
 double term;  
 Mortgage(int mortgageNumber, String customerName, double amountOfMortgage, double term){  
 this.customerName = customerName;  
 this.mortgageNumber = mortgageNumber;  
 this.amountOfMortgage = amountOfMortgage;  
 this.term = term;  
 }  
 public abstract void displayMortgageInfo();  
  
 public abstract void checkMortgageAmount();  
}

**PersonalMortgage.java**

package exercise3;  
  
public class PersonalMortgage extends Mortgage{  
 PersonalMortgage(int mortgageNumber, String customerName, double amountOfMortgage,double interestRate, double term){  
 super(mortgageNumber, customerName, amountOfMortgage, term);  
 this.interestRate = 2/interestRate;  
 }  
 public void checkMortgageAmount(){  
 if(amountOfMortgage > *maximumMortgageAmount*){  
 System.*out*.println("Your mortgage value is greater than 300,000.");  
 amountOfMortgage = 0;  
 }  
 }  
 public String checkTerm(){  
 if(term <= *shortTerm*)  
 return "Short term";  
 else if(term <= *mediumTerm*)  
 return "Medium term";  
 else  
 return "Long term";  
 }  
 @Override  
 public void displayMortgageInfo() {  
 System.*out*.println(  
 *bankName*+  
 " Personal Mortgage" +  
 "\nmortgageNumber= " + mortgageNumber +  
 "\ncustomerName= " + customerName +  
 "\namountOfMortgage= " + amountOfMortgage +  
 "\ninterestRate= " + interestRate +  
 "\nterm= " + term +"years. "+ checkTerm());  
 }  
}

**BusinessMortgage.java**

package exercise3;  
  
public class BusinessMortgage extends Mortgage{  
 BusinessMortgage(int mortgageNumber, String customerName, double amountOfMortgage, double interestRate,double term){  
 super(mortgageNumber, customerName, amountOfMortgage, term);  
 this.interestRate = 1/interestRate;  
 }  
 public void checkMortgageAmount(){  
 if(amountOfMortgage > *maximumMortgageAmount*){  
 System.*out*.println("Your mortgage value is greater than 300,000.");  
 amountOfMortgage = 0;  
 }  
 }  
 public String checkTerm(){  
 if(term <= *shortTerm*)  
 return "Short term";  
 else if(term <= *mediumTerm*)  
 return "Medium term";  
 else  
 return "Long term";  
 }  
 @Override  
 public void displayMortgageInfo() {  
 System.*out*.println(  
 *bankName*+  
 " Business Mortgage"+  
 "\nmortgageNumber= " + mortgageNumber +  
 "\ncustomerName= " + customerName +  
 "\namountOfMortgage= " + amountOfMortgage +  
 "\ninterestRate= " + interestRate +  
 "\nterm= " + term +"years. "+checkTerm());  
 }  
}

**MortgageDriver.java**

package exercise3;  
  
import exercise1.Health;  
import exercise1.Life;  
  
import java.util.Scanner;  
  
public class ProcessMortgage {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 Mortgage[] mortgages = new Mortgage[3];  
 System.*out*.println("Enter the current interest rate: ");  
 double currentInterestRate = scanner.nextDouble();  
 for(int i = 0; i < 3;i++){  
 System.*out*.print("Enter type of insurance:");  
 String typeOfMortgage = scanner.next();  
 if(typeOfMortgage.equals("personal")){  
 System.*out*.print("Enter mortgage number:");  
 int mortgageNumber = scanner.nextInt();  
 System.*out*.print("Enter customer name:");  
 String name = scanner.next();  
 System.*out*.print("Enter amount of mortgage:");  
 double amountOfMortgage = scanner.nextDouble();  
 System.*out*.print("Enter the term:");  
 double term = scanner.nextDouble();  
 mortgages[i] = new PersonalMortgage(mortgageNumber,name,amountOfMortgage,currentInterestRate,term);  
 mortgages[i].checkMortgageAmount();  
 System.*out*.println("You have created a Mortgage.\n");  
 }  
 else if(typeOfMortgage.equals("business")){  
 System.*out*.print("Enter mortgage number:");  
 int mortgageNumber = scanner.nextInt();  
 System.*out*.print("Enter customer name:");  
 String name = scanner.next();  
 System.*out*.print("Enter amount of mortgage:");  
 double amountOfMortgage = scanner.nextDouble();  
 System.*out*.print("Enter the term:");  
 double term = scanner.nextDouble();  
 mortgages[i] = new BusinessMortgage(mortgageNumber, name, amountOfMortgage,currentInterestRate,term);  
 mortgages[i].checkMortgageAmount();  
 System.*out*.println("You have created a Mortgage.\n");  
 }  
 else{  
 i--;  
 System.*out*.println("You entered incorrect mortgage type. Try Again.");  
 }  
 }  
 for(int i = 0; i < 3;i++){  
 System.*out*.println("\nMortgage"+(i+1));  
 mortgages[i].displayMortgageInfo();  
 }  
 }  
}