Loan Management System

**Grade settings**: Maximum grade: 100  
**Run**: Yes **Evaluate**: Yes  
**Automatic grade**: Yes

***Loan Management System***is an automated loan Interest Rate Calculation system. Bank Authorities are in the mode of wooing customers, in order to increase the customer base. To achieve this , they have come up with the initial plan to provide discounts on loan Interest Rate, for the loans taken by majorly 2 categories namely “Student” & Economically Weaker” sections, therefore they are depending upon this automated system to come out with faster Interest Calculation charts.

They have developed an application for the above purpose. The details of the various functions supported by the system are provided in this case study.

You are required to write Junit test case and check the correctness of the application developed.

Discounted Interested is calculated based on the Table given below:

|  |  |  |
| --- | --- | --- |
| **Loan Type** | **Loan Amt** | **DiscountedInterest(Actual)** |
| Educational Loan | 50000-100000 | 5% (6%) |
| Educational Loan | 100001 - 200000 | 4.5%(5.5%) |
| Educational Loan | >200000 | 4.0%(5%) |
| Personal Loan | 50000 - 100000 | 5%(7%) |
| Personal Loan | 100001 - 200000 | 4.5%(6.5%) |
| Personal Loan | 200001 - 1000000 | 4%(6%) |
| Housing Loan | 1000000 - 2000000 | 6.35%(8%) |
| Housing Loan | 2000001 - 5000000 | 6.25%(7.50%) |
| Housing Loan | > 5000000 | 6.0%(7%) |

**Functional Requirements :**

The application has the below classes and methods implemented.

You are provided with a model class **Loan**

**Component Specification : Loan (Model Class)**

|  |  |  |
| --- | --- | --- |
| **Type(Class)** | **Attributes** | **Methods** |
| Loan | String loanApplicationId  String customerId  String customerName  Date dateOfApplication  String[] documentSubmitted  String loanType;  String category  double loanAmtInlacRs  double interestInRs  String month  String year  int tenureInYears | Necessary getters and setters are provided.  An overloaded constructor in the is provided.  Also the toString methods are overridden. |

Here the **loanType** can take a value either “EducationalLoan” or “PersonalLoan” or “HousingLoan”.  If a Loan object that is being evaluated has any other value, it should throw the below mentioned exception.

**Component Specification :  InvalidLoanTypeException  (This class inherits the Exception Class)**

|  |  |
| --- | --- |
| **Type(Class)** | **Methods** |
| InvalidLoanTypeException | Provided with a single argument constructor – InvalidLoanTypeException (String message) |

And the **category** can take a value either “Student” or “EconomicallyWeaker” If a Loan object that is being evaluated has any other value, it should throw the below mentioned exception.

**Component Specification :  InvalidCustCategoryException  (This class inherits the Exception Class)**

|  |  |
| --- | --- |
| **Type(Class)** | **Methods** |
| InvalidCustCategoryException | Provided with a single argument constructor – InvalidCustCategoryException  (String message) |

**Component Specification :  LoanMgmtException  (This class inherits the Exception Class)**

Following Exception is thrown, when an Invalid Object of Loan is added , during the addition of the Loan objects to the List or during the calculation of the discounted Interest amount.

|  |  |
| --- | --- |
| **Type(Class)** | **Methods** |
| LoanMgmtException | Provided with a single argument constructor – LoanMgmtException(String message) |

You are also provided with an utility class LoanMgmtService with business methods.

**Component Specification :**LoanMgmtService **(Utility Class)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | **Type(Class)** | **Attributes** | **Methods** |
| LoanMgmtService | LoanMgmtService | List <Loan>  loanRequestsList | Getter and setter for the loanRequestsList are provided. |

The below are the requirements  implemented in the Utility class for which JUnit test cases are to be written and tested.

**Component Specification:  LoanMgmtService (Utility Class)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Type(Class)** | **Methods** | **Responsibilities** | **Exception** |
| Validating the **loanType** | LoanMgmtService | public boolean validateLoanType(String loanType) | Validate the loanType.. If valid return true , else this method should throw an userdefined exception | Throw an user defined exception “InvalidLoanTypeException”  if the loanType Is not  “EducationalLoan” or  “PersonalLoan”  or “HousingLoan” |
| Validating the **customerCategory** | LoanMgmtService | public boolean validateCustCategory(String category) | Validate the customer category.. If valid return true , else this method should throw an userdefined exception | Throw an user defined exception “InvalidCustCategoryException”  if the customerType Is not  “Student” or  “EconomicallyWeaker” . |
| Add a Loan  Record to List | LoanMgmtService | public boolean addLoanRecord(  Loan loan) | This method should validate the loanType. If type is valid add Loan object to the loanRequestsList. Else throw an user defined exception. | If loadType  is invalid, it will throw “Invalid LoanTypeException”, if customercategory is invalid it will throw an “InvalidCustCategoryException” |
| Calculate the DiscountedInterest for EducationalLoan(EL) | LoanMgmtService | public double getDiscountedInterestForEL(Loan loan)    EL- Education Loan | This method should return the discounted Interest Rate for EL of the Loan object passed as argument else throw an user defined exception. | Throw an user defined exception “LoanMgmtException”  if the object passed is invalid. |
| Calculate the DiscountedInterest for PersonalLoan(PL) | LoanMgmtService | public double getDiscountedInterestForPL(Loan loan)    PL – Personal Loan | This method should return the discounted Interest Rate for  PL of the Loan object passed as argument else throw an user defined exception. | Throw an user defined exception “LoanMgmtException”  if the object passed is  invalid. |
| Calculate the DiscountedInterest for HousingLoan(HL) | LoanMgmtService | public double getDiscountedInterestForHL(Loan loan)    HL –Housing Loan | This method should return the discounted Interest Rate for  HL of the Loan object passed as argument else throw an user defined exception. | Throw an user defined exception “LoanMgmtException”  if the  object  passed is invalid. |
| Convert the given String into Date | LoanMgmtService | public java.util.Date  stringToDateConverter(String stringDate) | This method converts the given string (in the  format “yyyy-MM-dd”) into date |  |

 You need to write Junit test for the LoanMgmtService class.

**Testing Scenarios:**

You are provided with a class  “LoanMgmtServiceTest”  to do this testing.

**Note :**

              To perform testing the loanRequestsList should contain objects of Loan.

              To do this, in LoanMgmtServiceTest class you are provided with a setup method.  Use this method or any ways , initialize the  loanRequestsList attribute in LoanMgmtService class.  Create few objects for  Loan,  populate a list with these objects and set the loanRequestsList to this list using the setLoanRequestsList method in LoanMgmtService class or otherwise also generate a List of Loan Objects for Testing.

The below are the test methods  to be implemented in LoanMgmtServiceTest class.

|  |  |
| --- | --- |
| **Test Method** | **Scenarios / Responsibilities** |
| test11ValidateLoanTypeWhenEducationalLoan | This method should test the validateLoanType method when a valid loan type  “EducationalLoan” is passed as parameter to this method. |
| test12ValidateLoanTypeWhenPersonalLoan | This method should test the validateLoanType method when a valid loan type  “PersonalLoan” is passed as parameter to this method. |
| test13ValidateLoanTypeWhenHousingLoan | This method should test the validateLoanType method when a valid loan type  “HousingLoan” is passed as parameter to this method. |
| test14ValidateLoanTypeWhenInvalidType | This method should test the validateLoanType method when an invalid loan Type is passed to this method.  validateLoanType method is expected  to throw InvalidLoanTypeException when loanType is invalid.  Write JUnit to test for the exception thrown  either by using appropriate annotation or by using try catch block. |
| test15ValidateCustCategoryWhenStudent | This method should test the validateCustCategory method when a valid customer Category  “Student” is passed as parameter to this method. |
| test16ValidateCustCategoryWhenEconomicallyweaker | This method should test the validateCustCategory method when a valid customer Category  “EconomiacallyWeaker” is passed as parameter to this method. |
| test17ValidateCustCategoryWhenInvalidCategory | This method should test the validateCustCategory method when an invalid customer Category  is passed to this method.  validateCustCategory method is expected  to throw InvalidCustCategoryException  when customer Category is invalid.  Write JUnit to test for the exception thrown  either by using appropriate annotation or by using try catch block. |
| test18AddLoanRecordForValidLoanType | This method should test the addLoanRecord method when  valid loanType is provided for the Loan object.  Test for the success scenario of addition of Loan object into the list. |
| test19AddLoanRecordForInvalidLoanType | This method should test the addLoanRecord method when  invalid loanType is provided for the Loan object. In this case, addLoanRecord method is expected to throw InvalidLoanTypeException  Write JUnit to test for the exception thrown  either by using appropriate annotation or by using try catch block. |
| test20AddLoanRecordForValidCustomerType | This method should test the addLoanRecord method when  valid customer Category is provided for the Loan object.  Test for the success scenario of addition of Loan object into the list. |
| test21AddLoanRecordForInvalidCustomerType | This method should test the addLoanRecord method when  invalid customer Category is provided for the Loan object. In this case, addLoanRecord method is expected to throw InvalidCustCategoryException  Write JUnit to test for the exception thrown  either by using appropriate annotation or by using try catch block. |
| test22GetDiscountedInterestForEL | This method should test the correctness of  getDiscountedInterestForEL method.  Perform testing for the correctness of the value returned. |
| test23GetDiscountedInterestForPL | This method should test the correctness of  getDiscountedInterestForPL method.  Perform testing for the correctness of the value returned. |
| test24GetDiscountedInterestForHL | This method should test the correctness of  getDiscountedInterestForHL method.  Perform testing for the correctness of the value returned. |

Implement the test methods and provide the needed annotation to all the methods in LoanMgmtServiceTest class.

Also provide the required annotation, so that the test methods are executed in ascending order of the test method names.

You are provided with a Main class with the main method to check the correctness of the test methods written in LoanMgmtServiceTest class.

Having completed writing the test methods, uncomment the code in Main class and execute the main method.