Experiment No: 1.3

Student Name: Kamal Ale Magar **UID: 21BCS10155**
Branch: CSE **Section/Group: 616/B**
Semester: 6th **Date of Performance: 01/22/24**
Subject: Project Based Learning in Java with Lab **Subject Code: 21CSH-319**

1. Aim:

Create an application to calculate interest for FDs, RDs based on certain conditions usinginheritance.

Task:

Calculate interest based on the type of the account and the status of the account holder. The rates of interest changes according to the amount (greater than or less than 1 crore), age of account holder (General or Senior citizen) and number of days if the type of account is FD or RD.
Some sample rates are given in the below tables:

	Current Rates of interest	
Maturity Period	General	Senior Citizen
7 days to 14 days	4.50	5.00
15 days to 29 days	4.75	5.25
30 days to 45 days	5.50	6.00
	Current Rates of interest	
Maturity Period	General	Senior Citizen
6 months	7.50	8.00
9 months	7.75	8.25
12 months	8.00	8.50

Requirements:

1. Separate classes should be created for the different types of accounts.
2. All classes should be derives from an abstract class named ‘Account’ which contains a methodcalled ‘calculateInterest’.
3. Implement the calculateInterest method according to the type of the account, interest rates,amount and age of the account holder.

4. If the user is entering any invalid value (For eg. Negative value) in any fields, raise a user definedexception.

Sample class structure is given below:

Account(Abstract)
`double interestRate`
`double amount`

Sample Output:

Select the option:

1. Interest Calculator –SB
2. Interest Calculator –FD
3. Interest Calculator –RD
4. Exit

1
Enter the Average amount in your account:
10000
Interest gained: Rs. 400

2. Objectives:

- To learn about concept of Inheritance.
- To learn about Abstract classes, Exception Handling.

3. Input/Apparatus Used:

Hardware Requirements: - Minimum 384MB RAM, 100 GB hard Disk, processor with 2.1 MHzSoftware Requirements: - VS Code Eclipse, NetBeans, IntelliJ etc

4. Procedure/Algorithm/Pseudocode:

Step 1: Import necessary packages:
Step 2: Create an abstract class Account:
Step 3: Create a class FDAccount that extends Account:
Step 4: Create a class SBAccount that extends Account:
Step 5: Create a class RDAccount that extends Account:
Step 6: Create the main class InterestCalculator:
Step 7: Inside main method, take user input and perform calculations based

on the selectedaccount type:

Step 8: Implement a menu-driven system to allow the user to choose the type of interestcalculation (SB, FD, RD) and exit the program.

5. Code:

```
import java.util.Scanner;

abstract class Account {
    double interestRate;
    double amount;

    abstract double calculateInterest() throws InvalidValueException;

    static class InvalidValueException extends Exception {
        public InvalidValueException(String message) {
            super(message);
        }
    }
}

class FDAccount extends Account {
    int noOfDays;
    int ageOfACHolder;

    @Override
    double calculateInterest() throws InvalidValueException {
        if (amount < 0 || noOfDays <= 0 || ageOfACHolder < 0) {
            throw new InvalidValueException("Invalid input values");
        }

        double interest = 0;

        if (amount <= 10000000) {
            if (noOfDays >= 7 && noOfDays <= 14) {
                interestRate = ageOfACHolder >= 60 ? 5.00 : 4.50;
            } else if (noOfDays >= 15 && noOfDays <= 29) {
                interestRate = ageOfACHolder >= 60 ? 5.25 : 4.75;
            } else if (noOfDays >= 30 && noOfDays <= 45) {
```

```
            interestRate = ageOfACHolder >= 60 ? 6.00 : 5.50;
        } else if (noOfDays >= 45 && noOfDays <= 60) {
            interestRate = ageOfACHolder >= 60 ? 7.50 : 7.00;
        } else if (noOfDays >= 61 && noOfDays <= 184) {
            interestRate = ageOfACHolder >= 60 ? 8.00 : 7.50;
        } else if (noOfDays >= 185 && noOfDays <= 365) {
            interestRate = ageOfACHolder >= 60 ? 8.50 : 8.00;
        } else {
            throw new InvalidValueException("Invalid number of days");
        }
    } else {
        if (noOfDays >= 7 && noOfDays <= 14) {
            interestRate = 6.50;
        } else if (noOfDays >= 15 && noOfDays <= 29) {
            interestRate = 6.75;
        } else if (noOfDays >= 30 && noOfDays <= 45) {
            interestRate = 6.75;
        } else if (noOfDays >= 45 && noOfDays <= 60) {
            interestRate = 8.00;
        } else if (noOfDays >= 61 && noOfDays <= 184) {
            interestRate = 8.50;
        } else if (noOfDays >= 185 && noOfDays <= 365) {
            interestRate = 10.00;
        } else {
            throw new InvalidValueException("Invalid number of days");
        }
    }

    interest = (amount * interestRate * noOfDays) / (36500);

    return interest;
  }
}

class SBAccount extends Account {
  @Override
  double calculateInterest() throws InvalidValueException {
    if (amount < 0) {
        throw new InvalidValueException("Invalid input values");
    }
```

```
        interestRate = 4.0;
        return (amount * interestRate) / 100;
    }
}

class RDAccount extends Account {
    int noOfMonths;
    double monthlyAmount;

    @Override
    double calculateInterest() throws InvalidValueException {
        if (amount < 0 || noOfMonths <= 0 || monthlyAmount < 0) {
            throw new InvalidValueException("Invalid input values");
        }

        double maturityAmount = amount;

        for (int i = 0; i < noOfMonths; i++) {
            maturityAmount += monthlyAmount;
            maturityAmount += (maturityAmount * interestRate) / 1200;
        }

        return maturityAmount - amount;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("Select the option:");
            System.out.println("1. Interest Calculator –SB");
            System.out.println("2. Interest Calculator –FD");
            System.out.println("3. Interest Calculator –RD");
            System.out.println("4. Exit");

            int choice = scanner.nextInt();

            if (choice == 4) {
                break;
```

```
        }

        Account account = null;

        switch (choice) {
            case 1:
                account = new SBAccount();
                break;
            case 2:
                account = new FDAccount();
                break;
            case 3:
                account = new RDAccount();
                break;
            default:
                System.out.println("Invalid choice. Please enter a valid option.");
                continue;
        }

        try {
            getInputValues(account, scanner);
            double interest = account.calculateInterest();
            System.out.println("Interest gained: Rs. " + interest);
        } catch (Account.InvalidValueException e) {
            System.out.println(e.getMessage());
        }
    }
}

private static void getInputValues(Account account, Scanner scanner) {
    System.out.println("Enter the amount:");
    account.amount = scanner.nextDouble();

    if (account instanceof FDAccount) {
        System.out.println("Enter the number of days:");
        ((FDAccount) account).noOfDays = scanner.nextInt();

        System.out.println("Enter your age:");
        ((FDAccount) account).ageOfACHolder = scanner.nextInt();
    } else if (account instanceof RDAccount) {
        System.out.println("Enter the number of months:");
```

```
            ((RDAccount) account).noOfMonths = scanner.nextInt();

            System.out.println("Enter the monthly amount:");
            ((RDAccount) account).monthlyAmount = scanner.nextDouble();
        }
    }
}
```

6. Result/Output:

```
Select the option:
1. Interest Calculator ?SB
2. Interest Calculator ?FD
3. Interest Calculator ?RD
4. Exit
1
Enter the amount:
100
Interest gained: Rs. 4.0
Select the option:
1. Interest Calculator ?SB
2. Interest Calculator ?FD
3. Interest Calculator ?RD
4. Exit
2
Enter the amount:
2000
Enter the number of days:
1
Enter your age:
14
Invalid number of days
Select the option:
1. Interest Calculator ?SB
2. Interest Calculator ?FD
3. Interest Calculator ?RD
4. Exit
```

```
3
Enter the amount:
3000
Enter the number of months:
12
Enter the monthly amount:
100
Interest gained: Rs. 1200.0
Select the option:
1. Interest Calculator ?SB
2. Interest Calculator ?FD
3. Interest Calculator ?RD
4. Exit
```

7. Learning Outcomes:

- Understanding of OOP concepts, including classes, objects, inheritance, and encapsulation.
- Application of abstract classes and extending classes to create specific account types.
- Experience in using the Scanner class to take user input from the console.
- Use of conditional statements (if-else) to determine interest rates based on various conditions such as thetype of account, the number of days, and the age of the account holder.
- Implementation of basic error handling using conditional statements to check for invalid input.

- Development of a menu-driven program, allowing users to choose between different options (SB, FD,RD) and exit the program.