Experiment-1.3

Student Name: Sanchit Singal
UID: 21BCS1569
Branch: BE-CSE
Section/Group: 606_B
Semester: 6TH
Date of Performance: 6/01/2024
Subject Name: Project Based learning with Java
Subject Code: 21CSH-319

1. **Aim:** Create a application to calculate interest for FDs, RDs based on certain conditions using inheritance.

2. **Objective:**
   - To learn about concept of Inheritance.
   - To learn about Abstract classes, Exception Handling.

3. Algorithm:

1. Create main java class to take input from user by three classes:

- SavingAccount.
- FixedDepositAccount.
- RecurringDeposit.

2. Implement cases for three choices:

   1. FD 2. RD 3.SavingAccount.

case 1: FD

I. Enter the amount:

If(amount<1 crore):
Enter the maturity period in days.
Enter the age.
If(age>60):

Print->Simple Interest evaluated from the array of "General" defined in class FixedDepositAccount.

Else if(age<60):

Print->Simple Interest evaluated from the array of "SeniorCitizen" defined In class FixedDepositAccount.

Else:

Print->Incorrect age.

Else if(amount>1crore):

Enter the maturity period in days.

Print->Simple Interest evaluated from the array of "interestRate" defined in class FixedDepositAccount.

Else:

Print->incorrect Amount.

II. Exit.

Case 2: RD

I. Enter the amount.
II. Enter the maturity period in months.
III. Enter the age:

If(age>60):

Print->Simple Interest evaluated from the array of "General" defined in class RecurringDeposit.

Else if(age<60):

Print->Simple Interest evaluated from the array of "SeniorCitizen" defined in class RecurringDeposit

Else:

Print->Incorrect age.

IV. Exit.

Case 3: SavingAccount

I. Enter the amount.
II. Enter the account type:

If(accountType=="NRI"):

interestRate=6%.

Print->Simple interest gained.

Else If(accountType==”normal”):
interestRate=4%.
Print->Simple interest gained.

Else:
Print->incorrect account type.

III. Exit.

3. Raise user defined Exception if input has invalid or negative values.
4. Exit.

4. Code:

```
import java.util.Scanner;
public class InterestCalculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Select option: ");
        System.out.println("1. Interest calculator-SB");
        System.out.println("2. Interest calculator-FD");
        System.out.println("3. Interest calculator-RD");
        int choice=sc.nextInt();
        switch(choice){
            case 1:
            System.out.println("Enter the amount: ");
            int amountSD = sc.nextInt();
            SBAccount sb = new SBAccount();
            sb.amount = amountSD;
            System.out.println("Interest gained on SB: " + sb.calculateInterest());
            break;
```

```
case 2:
System.out.println("Enter the FD amount: ");
int amountFD = sc.nextInt();
System.out.println("Enter the number of days: ");
int noOfDaysFD = sc.nextInt();
System.out.println("Enter the age of the account holder: ");
int ageOfACHolderFD = sc.nextInt();
FDAccount fd = new FDAccount();
fd.amount = amountFD;
fd.noOfDays = noOfDaysFD;
fd.ageOfACHolder = ageOfACHolderFD;
System.out.println("Interest gained on FD: " + fd.calculateInterest());
break;

case 3:
System.out.println("Enter the amount: ");
int amountRD = sc.nextInt();
System.out.println("Enter the number of Months: ");
int noOfDaysRD = sc.nextInt();
System.out.println("Enter the age of the account holder: ");
int ageOfACHolderRD = sc.nextInt();
RDAccount rd = new RDAccount();
rd.amount = amountRD;
rd.noOfMonths = noOfDaysRD;
rd.ageOfACHolder = ageOfACHolderRD;
System.out.println("Interest gained on RD: " + rd.calculateInterest());
sc.close();
break;
```

```
            default:
            System.out.println("Invalid Choice:");
        }
        sc.close();
    }
}
abstract class Account {
    double interestRate;
    double amount;
    abstract double calculateInterest();
}

class FDAccount extends Account {
    int noOfDays;
    int ageOfACHolder;

    double calculateInterest() {
        if (amount < 10000000) {
            if (noOfDays >= 7 && noOfDays <= 14) {
                if (ageOfACHolder >= 60) {
                    interestRate = 5.00;
                } else {
                    interestRate = 4.50;
                }
            } else if (noOfDays >= 15 && noOfDays <= 29) {
                if (ageOfACHolder >= 60) {
                    interestRate = 5.25;
```

```
    } else {
        interestRate = 4.75;
    }
} else if (noOfDays >= 30 && noOfDays <= 45) {
    if (ageOfACHolder >= 60) {
        interestRate = 6.00;
    } else {
        interestRate = 5.50;
    }
} else if (noOfDays >= 45 && noOfDays <= 60) {
    if (ageOfACHolder >= 60) {
        interestRate = 7.50;
    } else {
        interestRate = 7.00;
    }
} else if (noOfDays >= 61 && noOfDays <= 184) {
    if (ageOfACHolder >= 60) {
        interestRate = 8.00;
    } else {
        interestRate = 7.50;
    }
} else if (noOfDays >= 185 && noOfDays <= 365) {
    if (ageOfACHolder >= 60) {
        interestRate = 8.50;
    } else {
        interestRate = 8.00;
    }
}
```

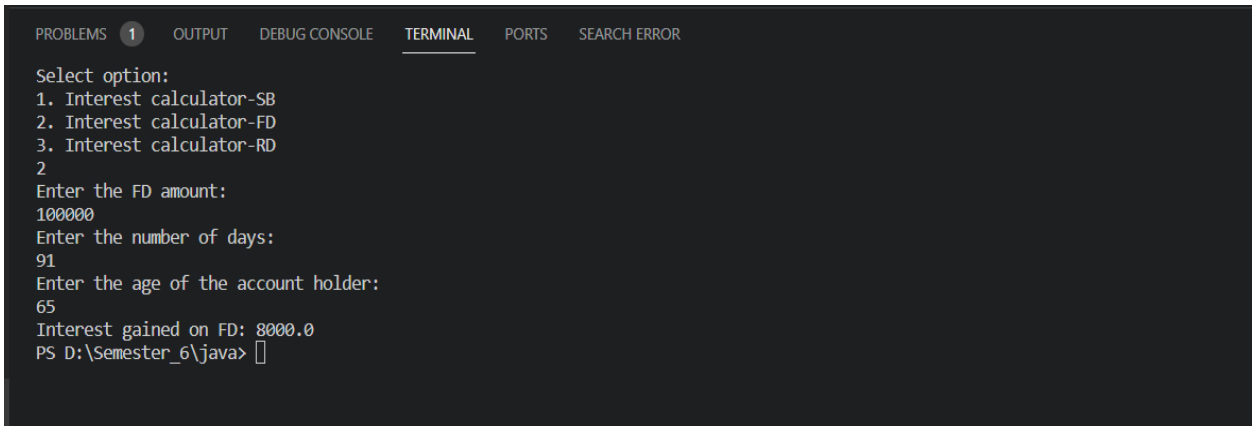
```
        } else {
            if (noOfDays >= 7 && noOfDays <= 14) {
                interestRate = 6.50;
            } else if (noOfDays >= 15 && noOfDays <= 29) {
                interestRate = 6.75;
            } else if (noOfDays >= 30 && noOfDays <= 45) {
                interestRate = 6.75;
            } 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;
            }
        }
        return (amount * interestRate) / 100;
    }
}

class SBAccount extends Account {
    double calculateInterest() {
        interestRate = 4.00;
        return (amount * interestRate) / 100;
    }
}
class RDAccount extends Account {
    int noOfMonths;
    int ageOfACHolder;
```

```
double calculateInterest() {
    if (noOfMonths == 6) {
        if (ageOfACHolder >= 60) {
            interestRate = 8.00;
        } else {
            interestRate = 7.50;
        }
    } else if (noOfMonths == 9) {
        if (ageOfACHolder >= 60) {
            interestRate = 8.50;
        } else {
            interestRate = 7.75;
        }
    } else if (noOfMonths == 12) {
        if (ageOfACHolder >= 60) {
            interestRate = 8.50;
        } else {
            interestRate = 8.00;
        }
    } else if (noOfMonths == 15) {
        if (ageOfACHolder >= 60) {
            interestRate = 8.75;
        } else {
            interestRate = 8.25;
        }
    } else if (noOfMonths == 18) {
        if (ageOfACHolder >= 60) {
            interestRate = 9.00;
```

```
            } else {
                interestRate = 8.50;
            }
        } else if (noOfMonths == 21) {
            if (ageOfACHolder >= 60) {
                interestRate = 9.25;
            } else {
                interestRate = 8.75;
            }
        }
        return (amount * interestRate) / 100;
    }
}
```

5. Output:



6. Learning Outcomes:

1) Array of Employee object to store multiple employ Information
2) Oops concept
3) Switch Statements