



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-3

Student Name: Deepak Yadav

Branch: BE-CSE

Semester: 6th

Subject Name: Project Based Learning in Java

UID:23ICS80029

Section/Group: 902/B

Date of Performance: 25.1.25

Subject Code: 22CSH-359

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

2.Objective: To design and implement a Java program that calculates interest for various account types (FD, RD, SB) using object-oriented principles, focusing on abstraction, method overriding, and dynamic input validation.

3.Implementation/Code:

```
abstract class Account {
    double interestRate;
    double amount;
    abstract double calculateInterest();
}
class FDAccount extends Account {
    int noOfDays;
    int ageOfACHolder;

    FDAccount(double amount, int noOfDays, int ageOfACHolder) {
        this.amount = amount;
        this.noOfDays = noOfDays;
        this.ageOfACHolder = ageOfACHolder;
    }
    @Override
    double calculateInterest() {
        if (amount < 10000000) { // Less than 1 crore
            if (noOfDays >= 7 && noOfDays <= 14) interestRate = ageOfACHolder >= 60 ? 5.0 : 4.5;
            else if (noOfDays >= 15 && noOfDays <= 29) interestRate = ageOfACHolder >= 60 ? 5.25 : 4.75;
            else if (noOfDays >= 30 && noOfDays <= 45) interestRate = ageOfACHolder >= 60 ? 6.0 : 5.5;
            else if (noOfDays >= 45 && noOfDays <= 60) interestRate = ageOfACHolder >= 60 ? 7.5 : 7.0;
            else if (noOfDays >= 61 && noOfDays <= 184) interestRate = ageOfACHolder >= 60 ? 8.0 : 7.5;
            else if (noOfDays >= 185 && noOfDays <= 365) interestRate = ageOfACHolder >= 60 ? 8.5 : 8.0;
        } else { // Greater than or equal to 1 crore
            if (noOfDays >= 7 && noOfDays <= 14) interestRate = 6.5;
            else if (noOfDays >= 15 && noOfDays <= 29) interestRate = 6.75;
            else if (noOfDays >= 30 && noOfDays <= 45) interestRate = 6.75;
        }
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
else if (noOfDays >= 45 && noOfDays <= 60) interestRate = 8.0;
else if (noOfDays >= 61 && noOfDays <= 184) interestRate = 8.5;
else if (noOfDays >= 185 && noOfDays <= 365) interestRate = 10.0;
}
return amount * interestRate / 100;
}
}

class RDAccount extends Account {
    int noOfMonths;
    double monthlyAmount;
    int ageOfACHolder;

    RDAccount(double monthlyAmount, int noOfMonths, int ageOfACHolder) {
        this.monthlyAmount = monthlyAmount;
        this.noOfMonths = noOfMonths;
        this.ageOfACHolder = ageOfACHolder;
    }

    @Override
    double calculateInterest() {
        if (noOfMonths == 6) interestRate = ageOfACHolder >= 60 ? 8.0 : 7.5;
        else if (noOfMonths == 9) interestRate = ageOfACHolder >= 60 ? 8.25 : 7.75;
        else if (noOfMonths == 12) interestRate = ageOfACHolder >= 60 ? 8.5 : 8.0;
        else if (noOfMonths == 15) interestRate = ageOfACHolder >= 60 ? 8.75 : 8.25;
        else if (noOfMonths == 18) interestRate = ageOfACHolder >= 60 ? 9.0 : 8.5;
        else if (noOfMonths == 21) interestRate = ageOfACHolder >= 60 ? 9.25 : 8.75;
        return monthlyAmount * noOfMonths * interestRate / 100;
    }
}

class SBAccount extends Account {
    String accountType;
    SBAccount(double amount, String accountType) {
        this.amount = amount;
        this.accountType = accountType;
    }

    @Override
    double calculateInterest() {
        interestRate = accountType.equalsIgnoreCase("NRI") ? 6.0 : 4.0;
        return amount * interestRate / 100;
    }
}
```

4. 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:
50000
Enter account type (Normal/NRI):
normal
Interest gained: Rs. 2000.0
Select the option:
1. Interest Calculator  SB
2. Interest Calculator  FD
3. Interest Calculator  RD
4. Exit
```

5. Learning outcomes:

1. Understand the concept of abstract classes and method overriding in Java.
2. Learn to implement real-world scenarios using object-oriented principles.
3. Develop skills to validate user input for different account types.
4. Gain knowledge of calculating interest dynamically based on conditions.
5. Enhance problem-solving abilities by applying conditional logic effectively.