



Experiment 3

Student Name: Ankur Kumar Rai

UID: 22BCS10411

Branch: CSE

Section/Group: 632/B

Semester: 6th

DOP: 02/02/2025

Subject: PBLJ Lab

Subject Code: 22CSH-359

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

Objective: To develop a Java application that calculates interest for Fixed Deposits (FDs) and Recurring Deposits (RDs) using object-oriented programming principles. The application will use inheritance to define common properties and methods for accounts while providing specific implementations for FDs and RDs based on their respective conditions.

Algorithm:

- **Create Account class** with attributes: `accountHolderName`, `principal`, `rateOfInterest`. Include methods for calculating interest (to be overridden) and displaying details.
- **Create FixedDeposit subclass** that calculates FD interest using: $\text{principal} * \text{rateOfInterest} * \text{tenureInYears} / 100$. Display FD details.
- **Create RecurringDeposit subclass** that calculates RD interest using: $(\text{monthlyDeposit} * \text{months} * (\text{months} + 1) / 2) * (\text{rateOfInterest} / (12 * 100))$. Display RD details.
- **In main method**, create instances of `FixedDeposit` and `RecurringDeposit` and display their details.

Code:

```
class Account {  
  
    String accountHolderName;  
  
    double principal;  
  
    double rateOfInterest;  
  
    public Account(String accountHolderName, double principal, double rateOfInterest) {  
  
        this.accountHolderName = accountHolderName;  
  
        this.principal = principal;  
  
        this.rateOfInterest = rateOfInterest;  
    }  
  
    public double calculateInterest() {  
  
        return 0;  
  
    }  
}
```

```
        public void displayDetails() {

            System.out.println("Account Holder: " + accountHolderName);

            System.out.println("Principal Amount: " + principal);

            System.out.println("Rate of Interest: " + rateOfInterest + "%");

        }

    }

    class FixedDeposit extends Account {

        int tenureInYears;

        public FixedDeposit(String accountHolderName, double principal, double rateOfInterest, int
            tenureInYears) {

            super(accountHolderName, principal, rateOfInterest);

            this.tenureInYears = tenureInYears;

        }

        public double calculateInterest() {

            return principal * rateOfInterest * tenureInYears / 100;

        }

        public void displayDetails() {

            super.displayDetails();

            System.out.println("Tenure (Years): " + tenureInYears);

            System.out.println("Interest Amount: " + calculateInterest());}

    class RecurringDeposit extends Account {

        int months;

        double monthlyDeposit;

        public RecurringDeposit(String accountHolderName, double monthlyDeposit, double rateOfInterest,
            int months) {
```

```
super(accountHolderName, 0, rateOfInterest);

this.monthlyDeposit = monthlyDeposit;

this.months = months;
}

public double calculateInterest() {

    // RD interest formula:  $P(n(n+1)/2) * (r / 12 * 100)$ 

    double n = months;

    return (monthlyDeposit * n * (n + 1) / 2) * (rateOfInterest / (12 * 100));
}

public void displayDetails() {

    System.out.println("Account Holder: " + accountHolderName);

    System.out.println("Monthly Deposit: " + monthlyDeposit);

    System.out.println("Number of Months: " + months);

    System.out.println("Rate of Interest: " + rateOfInterest + "%");

    System.out.println("Interest Amount: " + calculateInterest());
}
}

public class InterestCalculator {

    public static void main(String[] args) {

        // Example FD account

        FixedDeposit fd = new FixedDeposit("Sakshi", 100000, 5.5, 3);

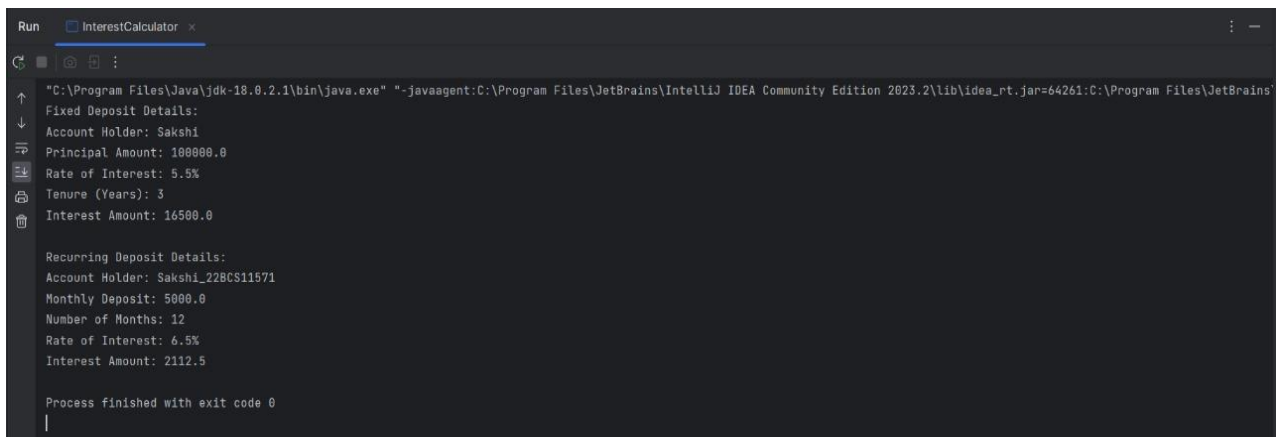
        System.out.println("Fixed Deposit Details:");

        fd.displayDetails();

        System.out.println();
    }
}
```

```
RecurringDeposit rd = new RecurringDeposit("Sakshi_22BCS11571", 5000, 6.5, 12);  
  
System.out.println("Recurring Deposit Details:");  
  
rd.displayDetails();  
  
}  
  
}
```

Output:



```
Run InterestCalculator x  
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2\lib\idea_rt.jar=64261:C:\Program Files\JetBrains"  
Fixed Deposit Details:  
Account Holder: Sakshi  
Principal Amount: 100000.0  
Rate of Interest: 5.5%  
Tenure (Years): 3  
Interest Amount: 16500.0  
  
Recurring Deposit Details:  
Account Holder: Sakshi_22BCS11571  
Monthly Deposit: 5000.0  
Number of Months: 12  
Rate of Interest: 6.5%  
Interest Amount: 2112.5  
  
Process finished with exit code 0  
|
```

Learning Outcomes:

- **Inheritance:** Use of base and derived classes for shared attributes and methods.
- **Method Overriding:** Custom implementation of methods in subclasses.
- **Constructor:** Initializing object attributes using constructors.
- **Encapsulation:** Storing and manipulating data within objects.
- **Polymorphism:** Different behavior of `calculateInterest()` based on object type.
- **Interest Calculation:** Implementing FD and RD interest formulas.
- **Class Interaction:** Creating objects and calling methods to display details.