The Calculator Application

Learning Objectives

The development process of the Calculator application will aid the students to:

- Create a simple Java console application
- Understand the object-oriented concepts of inheritance, polymorphism and data hiding
- Create application which request input from users, validate, process the input received and provide desired output.
- Use features of java like type conversion, interfaces, inheriting interfaces, looping and branching, packages and I/O classes.

Understanding the Calculator Application

The Calculator application performs both basic and scientific operations. The application provides user an option to choose between the basic mode and scientific mode. Based on the option selected by the user, the application calls the corresponding class and the user can perform various mathematical operations provided in the class. There is a base class in the application which contains all the methods for calculation, basic as well as scientific. The application validates the user input also and provides appropriate messages when wrong input is given by the user.

Creating the Calculator Application

To create the Calculator application, 5 java files were created. First, an interface iCalc, with the file name “iCalc.java” is created. Then, we create the base class Calculate, with the file name “Calculate.java” which contains all the methods for calculation. After the base class, two classes, Calculator and ScientificCalculator, with the file names as “Calculator.java” and “ScientificCalculator.java” are created. These classes call the methods defined in the base class Calculate. Class Calculator contains an instance of Class Calculate, whereas Class ScientificCalculator inherits Class Calculate and then uses its methods. After creation of all the above classes, a main class UseCalculate is created, with the file name “UseCalculate.java” which provides creates instances of Class Calculator or Class ScientificCalculator, based on the option selected by user.

Creating the Java Files

The iCalc Interface (iCalc.java)

Interface iCalc provides the structure of methods which can be used in any calculator application. It contains the following two methods:

- doCalculation(): Declares a method for providing methods for calculation.
- getResult(): Declares a method for extracting the result of the calculation.

The Calculate Class (Calculate.java)

Class Calculate contains the business logic of the Calculator application. It contains the methods for calculation of various mathematical operations like addition, divide and tangent. Class Calculate uses interfaces by implementing Interface iCalc. The class contains following methods:

Method	Description
Calculate()	Default constructor for the class without any arguments.
Calculate(Double dblNum, char cOperator)	Constructor containing two arguments. This constructor is used for scientific calculations.
Calculate(int iFirstNum, char cOperator, int iSecondNum)	Constructor containing three arguments. This constructor is used for basic calculations.
doCalculation()	Calculates the result based on the numbers and operator inputted by the user. Overriding the doCalculation function.of iCalc interface.
getResult()	Prints the result of calculation. Overriding the getResult function.of iCalc interface.
checkSecondNum()	In case of division of two numbers, it checks for value 0 in the second number entered.
checkInt()	Checks if basic calculation is performed.
checkDouble()	Checks if scientific calculation is performed.

The Calculator Class (Calculator.java)

Class Calculator calculates basic operations, namely, addition, subtraction, multiplication and division of two numbers. The class provides option to user to enter first number to be calculated, then the operation to be performed and then, the second number to be used for calculation. The input is received using java class BufferedReader. Class calculator creates an object of Class Calculate, by calling its constructor by passing three arguments, First Number, Operator and Second Number. After the creation of object of Class Calculate, doCalculation() method is called followed by getResult() method, which presents the result of calculation to the user.

Class Calculator also uses a do-while loop to provide an option to the user perform multiple calculations, till the user does not indicate the end of processing by typing ‘n’.

The ScientificCalculator Class (ScientificCalculator.java)

Class Calculator performs calculation of scientific operations, namely, sine, cosine, tangent and log of a number. The class provides option to user to enter the operation to be performed and the number to be calculated, The input is received using java class BufferedReader. Class ScientificCalculator inherits Class Calculate to use its methods. The class passes the user entered values to Class Calculate by calling its constructor having two arguments, Operator and the Number. The class calls doCalculation() method which is followed by getResult() method, which shows the result of calculation to the user.

Class ScientificCalculator also uses a do-while loop to provide an option to the user perform multiple calculations, till the user does not indicate the end of processing by typing ‘n’.

The UseCalculator Class (UseCalculator.java)

Class Calculator provides two options to the user, Basic or Scientific. Based on the option entered by the user, the class creates an instance of Class Calculator for Basic operations or an instance of Class ScientificCalculator for scientific operations.

Class UseCalculator also uses a do-while loop to provide an option to the user perform multiple calculations, till the user does not indicate the end of processing by typing ‘n’.

Generating the Class Files

Command for generating the class files:

javac <classname.java>

The steps for generating the class files for Calculator application are:

- Place all java files in a directory named as “Calculator”.
- In the command prompt, go to the directory where the java files are stored for the Calculator application.
- Compile the following java files in the sequence given below using the command for compiling the java files:
    - iCalc.java
    - Calculate.java
    - Calculator.java
    - ScientificCalculator.java
    - UseCalculator.java

Working with the Calculator application

The steps for working with the Calculator application are:

- In the command prompt, go to the parent directory of “Calculator” directory which contains the class files for Calculator application.
- Enter the following command to run the Calculator application:

    java Calculator.UseCalculator

- Enter ‘b’ (for Basic operations) or ‘s’ (for scientific operations) depending on the operations to be performed.
- If ‘b’ is entered, the following input is to be entered by the user:
    - First Number
    - Operator
    - Second Number
- The result is shown on the command prompt based on the above values.
- Enter ‘y’ to continue or ‘n’ to discontinue using the application.
- If ‘s’ is entered, the following input is to be entered by the user:
    - Operator
    - Number

- The result is shown on the command prompt based on the above values.
- Enter ‘y’ to continue or ‘n’ to discontinue using the application.

Code for the Calculator Application

iCalc.java

```
/*************

-- Interface iCalc represents the basic methods for the Calculate class
-- Creates Interface Structure
-- Can be used for creating any class which would do any sort of calculations

*************/

// Adds the Interface to the Package
package Calculator;

//Interface Definition
interface iCalc
{
        public void doCalculation();
        public void getResult();
}
```

Calculate.java

```
/*************

-- Class Calculate has all methods for calculation as required by any Calculator classes
-- Implements an interface iCalc

*************/

// Adds the Class to the Package
package Calculator;

// Class Definition
class Calculate implements iCalc
{
        private char Operator;
        private int  iFNum, iSNum;
        private Double dblNumber=new Double(0);
        private Double dblResult=new Double(0);
        private int iResult=0;
        private boolean typeDouble=false;
        private boolean typeInt=false;

        // Defines a constructor for scientific calculations
        public Calculate()
        {}

        public Calculate(Double dblNum, char cOperator)
        {
```

```
    dblNumber=dblNum;
    Operator=cOperator;
    typeDouble=true;
}

// Defines a constructor for basic calculations
public Calculate(int iFirstNum, char cOperator, int iSecondNum)
{
    iFNum=iFirstNum;
    iSNum=iSecondNum;
    Operator=cOperator;
    typeInt=true;
}

// Calculates the Result based on the operator selected by the user
public void doCalculation()
{
    iResult=0;
    dblResult=0.0;

    switch (Operator)
    {
        case '+':
            checkInt();
            iResult = iFNum + iSNum;
            break;

        case '-':
            checkInt();
            iResult = iFNum - iSNum;
            break;

        case '*':
            checkInt();
            iResult = iFNum * iSNum;
            break;

        case '/':
            checkInt();
            if(!checkSecondNum())
            {
                iResult = iFNum / iSNum;
                break;
            }

        case 'S':
        case 's':
            checkDouble();
            dblResult = Math.sin(dblNumber);
            break;

        case 'C':
        case 'c':
            checkDouble();
            dblResult = Math.cos(dblNumber);
            break;
```

```
                        case 'T':
                        case 't':
                                checkDouble();
                                dblResult = Math.tan(dblNumber);
                                break;

                        case 'L':
                        case 'l':
                                checkDouble();
                                dblResult = Math.log(dblNumber);
                                break;

                        default :
                                iResult=0;
                                dblResult=0.0;
                                System.out.println("***Operation Not Available. Please select
any of the available options.***");
                                break;
                }
        }

        // Displays the result of calculation to the user
        public void getResult()
        {
                if(typeInt)
                {
                        System.out.println("The result is: " + iResult);
                }
                else if(typeDouble)
                {
                        System.out.println("The result is: " + dblResult);
                }
        }

        // Checks for zero
        public boolean checkSecondNum()
        {
                if(iSNum==0)
                {
                        System.out.println("Zero Not allowed");
                        System.exit(0);
                        return true;
                }
                else
                {
                        return false;
                }
        }

        public void checkInt()
        {
                if(!typeInt)
                {
                        iResult=0;
```

```
            System.out.println("***Operation Not Available. Please select any of the 
available options.***");
            System.exit(0);
        }
    }

    public void checkDouble()
    {
        if(!typeDouble)
        {
            dblResult=0.0;
            System.out.println("***Operation Not Available. Please select any of the 
available options.***");
            System.exit(0);
        }
    }
}
```

Calculator.java

```
/*************

-- Class Calculator Performs basic operations like Add, Substract, Multiply, Division for two 
numbers
-- Uses class Calculate by creating its objects and then calling its methods

*************/

// Adds the Class to the Package
package Calculator;

// Imports Required Packages
import java.io.*;

// Class Definition
class Calculator
{
    public void Calc() throws java.io.IOException
    {
        boolean next;

        do
        {
            Integer iFirstNumber=new Integer(0);
            Integer iSecondNumber=new Integer(0);

            BufferedReader buffer
              = new BufferedReader(new InputStreamReader(System.in));

            // Gets User Input
            System.out.println("Please enter First Number: ");
            System.out.flush();

            try
            {
                iFirstNumber=Integer.parseInt(buffer.readLine());

            }
            catch(NumberFormatException e)
            {
                System.out.println("***Please provide numeric values.***");
                System.exit(0);
            }

            System.out.println("Please enter the Operation (Add : +, Minus : -,
Product : *, Divide : /):");
            System.out.flush();
            String option=buffer.readLine();

            System.out.println("Please enter Second Number: ");
            System.out.flush();

            try
            {
                iSecondNumber=Integer.parseInt(buffer.readLine(),10);
            }
            catch(NumberFormatException e)
            {
                System.out.println("***Please provide numeric values.***");
                System.exit(0);
            }

            if(option.length()==1)
            {
                // Creates Calculate Class Instance
                Calculate c= new
Calculate(iFirstNumber,option.charAt(0),iSecondNumber);

                // Calls the class methods
                c.doCalculation();
                c.getResult();
            }
            else
            {
                System.out.println("***Operation Not Available. Please select
any of the available options.***");
            }

            // Checks if the user would like to compute again
            System.out.println("Would you like to calculate again (y/n)?");
            System.out.flush();
            char response=(char)buffer.read();
            if ((response=='y') || (response=='Y'))
            {
                next=false;
            }
            else
            {
                next=true;
            }
```

```
        }while (!next);

    }

}
```

ScientificCalculator.java

```
/*************

-- Class ScientificCalculator Performs scientific calculations like Sine, Cosine, Tangent and Log of
a number
-- Inherits class Calculate
-- Methods of Super class Calculate can be directly called by using the object of this Sub class
ScientificCalculator

*************/

// Adds the Class to the Package
package Calculator;

// Imports Required Packages
import java.io.*;

// Class Definition
class ScientificCalculator extends Calculate
{
    char Operator;
    Double dblNumber=new Double(0);

    ScientificCalculator(){}

    ScientificCalculator(double dblNumber,char Operator)
    {
        // Calls Super Class Constructor
        super(dblNumber,Operator);

        this.Operator=Operator;
        this.dblNumber=dblNumber;
    }

    public void Calc() throws java.io.IOException
    {
        boolean next;

        do
        {
            Double d=new Double(0);

            BufferedReader buffer
              = new BufferedReader(new InputStreamReader(System.in));

            // Gets User Input
            System.out.println("Please enter the Operation (Sine-s, Cosine-c,
Tangent-t, Log-l):");
            System.out.flush();
            String option=buffer.readLine();

            System.out.println("Please enter a Value: ");
            System.out.flush();
            try
            {
                d=Double.valueOf(buffer.readLine());
            }
            catch(NumberFormatException e)
            {
                System.out.println("***Please provide numeric values.***");
                System.exit(0);
            }

            if(option.length()==1)
            {
                // Creates Class Instance
                ScientificCalculator sc=new
ScientificCalculator(d,option.charAt(0));

                // Calls Super Class Methods
                sc.doCalculation();
                sc.getResult();
            }
            else
            {
                System.out.println("***Operation Not Available. Please select
any of the available options.***");
            }

            // Checks if the user would like to compute again
            System.out.println("Would you like to calculate again (y/n)?");
            System.out.flush();
            char aa=(char)buffer.read();
            if ((aa=='y') || (aa=='Y'))
            {
                next=false;
            }
            else
            {
                next=true;
            }

        }while (!next);
    }
}
```

UseCalculator.java

```
/*************

-- Class UseCalculator is the Main Class for the Calculator Application
-- Provides options to the user: Basic or Scientific Calculator

*************/
```

```
// Adds the Class to the Package
package Calculator;

// Imports Required Packages
import java.io.*;

// Class Definition
class UseCalculator
{
    public static void main(String[] args) throws java.io.IOException
    {
        BufferedReader buffer
            = new BufferedReader(new InputStreamReader(System.in));

        // Gets User Input
        System.out.println("Select the Calculator: Basic - B or Scientific - S.");
        System.out.flush();
        String option=buffer.readLine();

        if(option.length()==1)
        {
            if (option.equals("B") || option.equals("b"))
            {
                // Calls the Basic Calculator Application
                Calculator c=new Calculator();
                c.Calc();
            }
            else if(option.equals("S") || option.equals("s"))
            {
                // Calls the Scientific Calculator Application
                ScientificCalculator sc=new ScientificCalculator();
                sc.Calc();
            }
            else
            {
                System.out.println("***Please enter option 'B' or 'S'.***");
            }
        }
        else
        {
            System.out.println("***Please enter option 'B' or 'S'.***");
        }
    }
}
```

Exercises:

1. Write the java code to calculate scientific operations like asin, atan and acos in the Calculator application.

2. Write the java code to use both, basic and scientific operations available in Class Calculate using multiple inheritance.