Mini-project Report

On

Simple Calculator of the Mini Project

submitted in partial fulfillment of the requirements of

Degree of Second Year Artificial Intelligence & Data Science

by

Parth Puri 24

Divyah Mandavia. 11

Devashree Pawar 22



**University of Mumbai**

**Vidyavardhini's College of Engineering & Technology**

**Department of Artificial Intelligence & Data Science**



**(A.Y. 2021-22)**

# CERTIFICATE

This is to certify that the Mini Project entitled **“Calculator”** is submitted by **Parth Puri (Roll No. 24), Divyah Mandavia (Roll No. 11), Devashree Pawar (Roll No. 22)** for the subject of Object-Oriented Programming of Java Lab in the Department of Artificial Intelligence & Data Science as a record of work done by him/her under our supervision and guidance.

**Internal Examiner**

**Guide**

**Asst. Prof. Tina D’abreo**

# Contents

1. **Introduction Page No**
   1. Introduction to the project
2. **OOPJ Concepts Page No**
   1. Features of Java
      * Object Oriented
      * Compiled and Interpreted
      * Platform-Independent and Portable
   2. Concepts of OOPJ
      * Classes, object, methods
      * Method/ Constructor overloading
      * String
      * Inheritance
      * AWT
3. **Code Page No.**
   1. Source code
   2. Snapshots of Result

1. **Conclusion Page No**
2. **Introduction**

* 1. Introduction to the project

We have to write a program in Java such that it creates a calculator which allows basic operations of addition, subtraction, multiplication, division and modulus.

**Java AWT** (Abstract Window Toolkit) is an API to develop Graphical User Interface (GUI) or windows-based applicationsin Java.

Java AWT components are platform-dependent i.e., components are displayed according to the view of operating system. AWT is heavy weight i.e., its components are using the resources of underlying operating system (OS).

Simple calculator to perform Addition, Subtraction, Multiplication, division and modulus. Added Action Listener to each button. Additional Feature of this calculator is that it can accept multiple input with different operators. The Calculator has various functions in all.

1. **OOPJ Concepts**

* 1. Features of Java
* Object Oriented –

Object Oriented programming (OOP) is a programming paradigm that relies on the concept of classes and objects. It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects.

* Compiled and Interpreted

The Java source code first compiled into a binary byte code using Java compiler, then this byte code runs on the JVM (Java Virtual Machine), which is a software-based interpreter. So, Java is considered as both interpreted and compiled.

We can execute a Java application by following two steps.

1) Compile the Java program

2) Execute the application

* Platform-Independent and Portable

The java compiled code (byte code) can run on all operating systems. A program is written in a language that is a human-readable language. It may contain words, phrases, etc which the machine does not understand. For the source code to be understood by the machine, it needs to be in a language understood by machines, typically a machine-level language. So, here comes the role of a compiler. The compiler converts the high-level language (human language) into a format understood by the machines. Therefore, a compiler is a program that translates the source code for another program from a programming language into executable code.

* 1. Concepts of OOPJ
     1. Classes, object, methods

Classes-: The Object class is the parent class of all the classes in java by default. In other words, it is the topmost class of java. The Object class is beneficial if you want to refer any object whose type you don't know. Notice that parent class reference variable can refer the child class object, known as upcasting.

The Object class provides some common behaviours to all the objects such as object can be compared, object can be cloned, object can be notified etc.

Object-: An object in Java is the physical as well as a logical entity, whereas, a class in Java is a logical entity only.

Method-: A method is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation. It is used to achieve the reusability of code. We write a method once and use it many times. We do not require to write code again and again. It also provides the easy modification and readability of code, just by adding or removing a chunk of code. The method is executed only when we call or invoke it. The most important method in Java is the main() method.

* + 1. String

A Java string is a sequence of characters that exist as an object of the class java. ... Once created, a string is immutable -- its value cannot be changed. methods of class String enable: Examining individual characters in the string.

methods of class String enable:

Examining individual characters in the string.

Comparing strings.

Searching strings.

Copying strings with characters converted from upper to lower case or vice versa.

* + 1. Inheritance

Inheritance in Java is a concept that acquires the properties from one class to other classes; for example, the relationship between father and son. In Java, a class can inherit attributes and methods from another class. The class that inherits the properties is known as the sub-class or the child class.

* + 1. Method/ Constructor overloading

A constructor in Java is a block of code similar to a method that's called when an instance of an object is created. ... Unlike methods, constructors are not considered members of a class. A constructor is called automatically when a new instance of an object is created.

* + 1. AWT

The Abstract Window Toolkit (AWT) is Java's original platform-dependent windowing, graphics, and user-interface widget toolkit, preceding Swing. The AWT is part of the Java Foundation Classes (JFC) — the standard API for providing a graphical user interface (GUI) for a Java program. AWT is also the GUI toolkit for a number of Java ME profiles. For example, Connected Device Configuration profiles require Java runtimes on mobile telephones to support the Abstract Window Toolkit.

1. **OOPJ Concepts**

* 1. Source code

import java.awt.\*;

import java.awt.event.\*;

class Calculator1 implements ActionListener

{

//Declaring Objects

Frame f=new Frame();

Label l0=new Label("Calculator");

Label l1=new Label("First Number");

Label l2=new Label("Second Number");

Label l3=new Label("Result");

Label l4=new Label("Simple");

TextField t1=new TextField();

TextField t2=new TextField();

TextField t3=new TextField();

Button b1=new Button("Add");

Button b2=new Button("Sub");

Button b3=new Button("Mul");

Button b4=new Button("Div");

Button b5=new Button("Cancel");

Button b6=new Button("Mod");

Calculator1()

{

//Giving Coordinates

l0.setBounds(150,75,175,20);

l1.setBounds(50,100,100,20);

l2.setBounds(50,140,100,20);

l3.setBounds(50,180,100,20);

l4.setBounds(160,55,175,30);

t1.setBounds(200,100,100,20);

t2.setBounds(200,140,100,20);

t3.setBounds(200,180,100,20);

b1.setBounds(50,250,50,20);

b2.setBounds(110,250,50,20);

b3.setBounds(170,250,50,20);

b4.setBounds(230,250,50,20);

b5.setBounds(290,250,50,20);

b6.setBounds(125,290,50,20);

//Adding components to the frame

f.add(l0);

f.add(l1);

f.add(l2);

f.add(l3);

f.add(l4);

f.add(t1);

f.add(t2);

f.add(t3);

f.add(b1);

f.add(b2);

f.add(b3);

f.add(b4);

f.add(b5);

f.add(b6);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

b5.addActionListener(this);

b6.addActionListener(this);

f.setLayout(null);

f.setVisible(true);

f.setSize(400,350);

}

public void actionPerformed(ActionEvent e)

{

float n1=Integer.parseInt(t1.getText());

float n2=Integer.parseInt(t2.getText());

if(e.getSource()==b1)

{

t3.setText(String.valueOf(n1+n2));

}

if(e.getSource()==b2)

{

t3.setText(String.valueOf(n1-n2));

}

if(e.getSource()==b3)

{

t3.setText(String.valueOf(n1\*n2));

}

if(e.getSource()==b4)

{

t3.setText(String.valueOf(n1/n2));

}

if(e.getSource()==b6)

{

t3.setText(String.valueOf(n1%n2));

}

if(e.getSource()==b5)

{

System.exit(0);

}

}

public static void main(String args[])

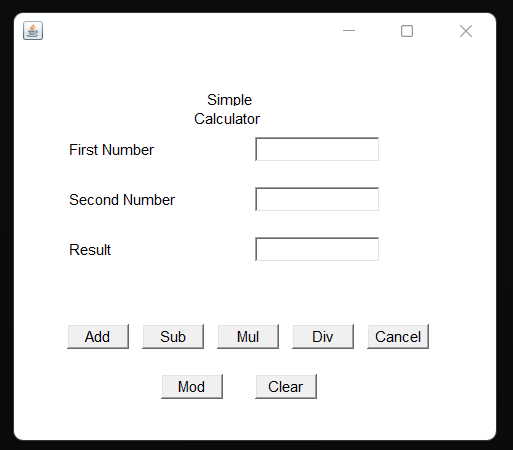
{

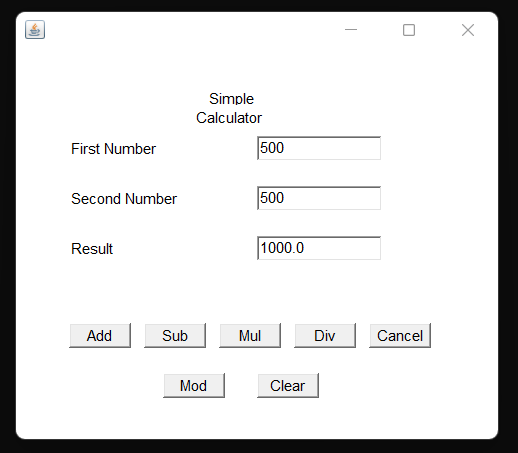
new Calculator1();

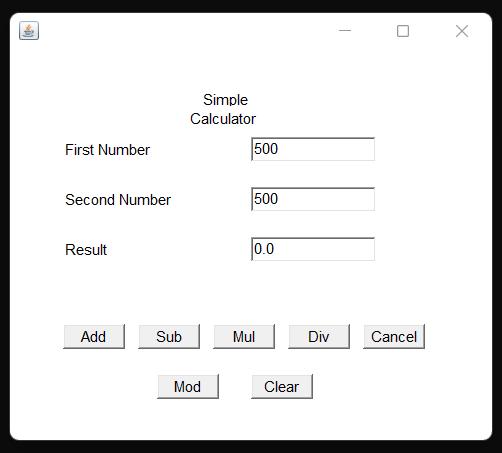
}

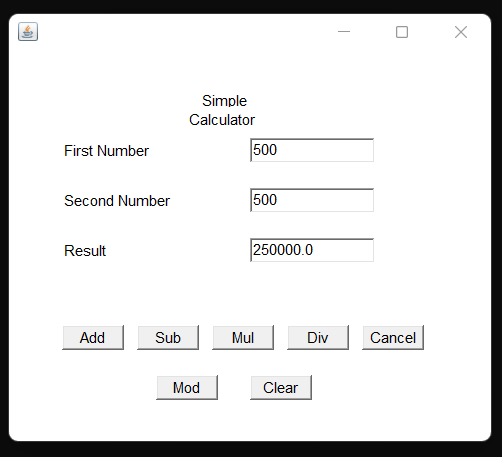
}

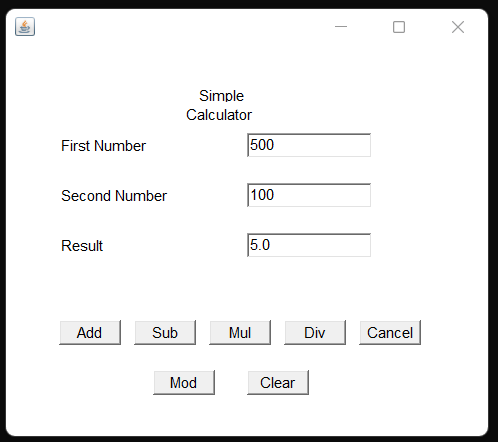
* 1. Snapshots of Result

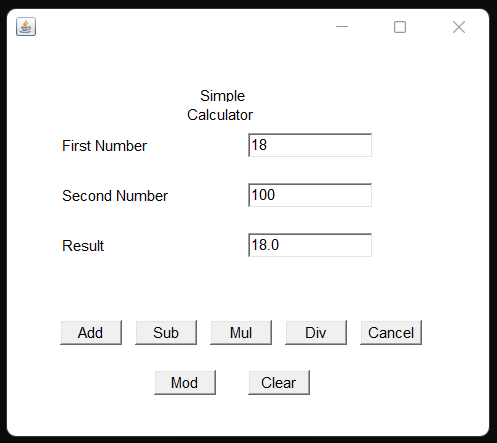












1. **Conclusion**

In this project we have implemented the program to display a simple calculator by using various components java awt. The awt components helps us in providing action listeners, different buttons and etc. In this Calculator the user can perform 7 different functions. The calculator gives us perfectly accurate values in the result text field.