Task 1:
Identify Code smells

1.Long Method:

Where: This code smells occur in Calculator class..And the method name is calculateBiImpl().
Description: It is code smell that is Bloaters.
Why : In this class this method violate Single class responsibility..here it doing add , minus, multiply, divide etc operation..

2.Large Class:

Where: This code smells occur in calculator test.The class name is CalculatorTest.
Description: It is code smells of Bloaters.
Why: This class contains too many fields/methos like calculate add test, BinormalTest etc.. they get bloated as the program grows..

3.Switch Statements:

Where: The code smells in UI.java files and it occurs in actionPerformed methods..

Description: It is a code smells of Object-Orientation Abusers.

Why: long if else block used here instead of polymorphism..AS a rule of thumb you should think polymorphisms when you see soo many conditionals.

Task 2:

Identify Design Patterns

The problem is occur in the file of calculator.java..The calculator class has two strategy binary and unary operation but in future if we add ternaryoperation strategy we must change the calculator class..Instead of having one class with many operations variations of the same behavior you separate each beahavior its own strategy..

The refactored file name is Strategy folder which in the simplejavacalculator folder.

Task 3: i will implement a pattern that is proxy..this pattern helps me to get the cache image without disturbing the real resources..

My location of this feature is proxy folder which is in the simple javaCalculator..