Java Programming

Assignment1: Simple Command Line Calculator

Due Date: Sunday, Jun 2, 2024 5:59 PM

Submission: A zipped file containing the project source folder and screenshots.

Coverage

- **Types**
- **Variables**
- **Control flow**
- **Operators**
- **Functions**

Description

The assignment aims at building a Command Line Interface program whose utility is to do basic calculations.
The program should be able to perform the following operations:

- Addition:
    - Do an addition between two numerical variables of the same sign.
        - 5 + 2 = 7
    - Do an addition between more than two numerical variables of the same sign.
        - 5 + 2 + 6 = 13
    - Do an addition between two numerical variables, respectively integer and float.
        - 5 + 3.2 = 8.2

- Subtraction:
    - Do a subtraction between two numerical variables of the same sign.
        - 5 - 2 = 3
    - Do a subtraction between more than two numerical variables of the same sign.
        - 5 - 2 - 4 = -1
    - Do a subtraction between two numerical variables, respectively integer and float.
        - 5 - 3.2 = 1.8
- Multiplication:
    - Do a multiplication between two numerical variables of the same sign.
        - 5 * 2 = 10
    - Do a multiplication between more than two numerical variables of the same sign.
        - 5 * 2 * 5 = 50

    - Do a multiplication between two numerical variables, respectively integer and float.
        - 5 * 2.3 = 11.5

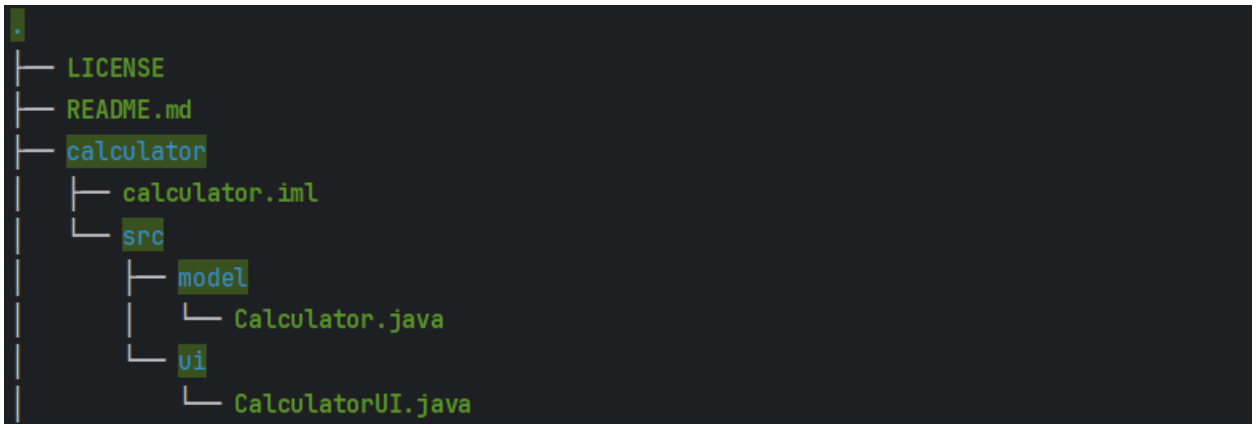
- Divide:

    - Divide two numerical variables of the same sign.
        - 10 / 2 = 5
    - Divide two numerical variables, respectively an integer and a float.
        - 24 / 2.5 = 9.6
    - Check for division errors.
        - 34 / 0 = division per 0 error.

Task

Download the starter code from the provided URL (starter code) and unzip it into a folder.

Your project should have the following structure:



Notes:

- *You must have Java installed on your machine to be able to proceed with the assignment. In case you don't have it installed, follow the instructions here: java installation guide.*
- *Make sure that you have a version of JDK from 17.0 onward. Versions anterior to 17 do not support some of the syntax and expressions in this assignment.*
- *You can use any editor of your choice for this assignment but also choose one of the following: IntelliJ IDEA, Eclipse IDE, or blueJ in case you are new to Java…*
- *Use the **javac** command to compile Java class files into binary and use **java** command to run as an executable.*

- *You can only modify the files asked to be modified by the instructions. You should not change any code or create additional files unless specified in the instructions.*
-

1- In the file called Calculator.java (in the model package), complete the following functions as specified:

- add:

```
public double add(double... numbers) {  Pericles001
    /**
     * Method to add numbers.
     * @param numbers The numbers to add.
     */
    // TODO: Use a loop to iterate through 'numbers' and add them together. Return the sum.
    return 0;
}
```

- Create a variable to store the results of calculations (Decide the appropriate type to use).
- Use a looping statement to iterate through the numbers.
- Use the '+' operator to add variables.
- Return the result.

- subtract:

```
public double subtract(double... numbers) {  3 usages  Pericles001
    /**
     * Method to subtract numbers.
     * @param numbers The numbers to subtract.
     */
    // TODO: Use a loop to iterate through 'numbers' and subtract them from the first number. Return the result.
    return 0;
}
```

- Create a variable to store the results of calculations (Decide the appropriate type to use).
- Use a looping statement to iterate through the numbers.
- Use the '-' operator to subtract between variables.
- Round the result to two decimal places if there is a need.
- Return the result.

- multiply:

```
public double multiply(double... numbers) {  3 usages  Pericles001
    /**
     * Method to multiply numbers.
     * @param numbers The numbers to multiply.
     */
    // TODO: Use a loop to iterate through 'numbers' and multiply them together. Return the product.
    return 0;
}
```

- Create a variable to store the results of calculations (Decide the appropriate type to use).
- Use a looping statement to iterate through the numbers.
- Use the '*' operator to subtract between variables.
- Return the result.

- divide:

```
public double divide(double... numbers) throws IllegalArgumentException {  3 usages  Pericles001
    /**
     * Method to divide numbers.
     * @param numbers The numbers to divide.
     */
    // TODO: Divide the first number by the second number. Make sure to handle the case where the second number is zero to avoid division by zero.
    return 0;
}
```

- Create a variable to store the results of calculations (Decide the appropriate type to use).
- Use a looping statement to iterate through the numbers.
- Check for basic division conditions.
- Use the '/' operator to subtract between variables.
- Return the result.

- advanced

```java
public double advanced(String expression) {  2 usages  Pericles001
  /**
   * Method to evaluate an expression.
   * @param expression The expression to evaluate.
   */
  // TODO: Implement a way to parse the expression into tokens (numbers and operators), call the variable tokens.
    // HINT: You can use String.split() method, but consider how to handle spaces
  String[] tokens = expression.split( regex: "Change the delimiter to split the expression into tokens.");

  double result = Double.parseDouble(tokens[0]);
  for (int i = 1; i < tokens.length; i += 2) {
    String operator = tokens[i];
    double number = Double.parseDouble(tokens[i + 1]);
    switch (operator) {
      // TODO: Implement the correct logic for each operator
      case "+" -> {
        // TODO: Add 'number' to 'result'
      }
      case "-" -> {
        // TODO: Subtract 'number' from 'result'
      }
      case "*" -> {
        // TODO: Multiply 'result' by 'number'
      }
      case "/" -> {
        if (number == 0) {
          // TODO: Throw an IllegalArgumentException if 'number' is zero
        }
        // TODO: Divide 'result' by 'number'
          // HINT: Use the yield statement to return the result of the division
      }
      // TODO: Throw an IllegalArgumentException for unknown operators.
        // HINT: use default -> throw ... expression to throw an exception for unknown operators
    };
  }
  // TODO: Return the final result
  return 0;
}
```

- Create a variable to store the results of calculations (Decide the appropriate type to use).
- Use a looping statement to iterate through the numbers.
- Inside the loop, use the switch-case statement to identify the operator.
- Use the appropriate operation based on the identified operator (+, -, *, /) to calculate and store the result in the result variable.
- Check for basic division conditions (Divide by zero…).
- Return the result.

2- Run the program from the following files:

- **CalculatorUI.java**, in the package **ui**:

- Do the following operations:
    - Operation1: addition(5, 20)
    - Operation2: subtract(10, 12)
    - Operation3: multiply(20, 10)
    - Operation4: divide(10, 2)
    - Operation5: advanced (5 + 2 * 10)

- Save the output in the screenshot (named firstname_screenshot1) as follows:

```
Choose an operation: add, subtract, multiply, divide, advanced, or type 'help' for help, 'exit' to quit
add
Enter numbers separated by spaces:
5 20
Result: 25.0
Choose an operation: add, subtract, multiply, divide, advanced, or type 'help' for help, 'exit' to quit
subtract
Enter numbers separated by spaces:
10 12
Result: -2.0
Choose an operation: add, subtract, multiply, divide, advanced, or type 'help' for help, 'exit' to quit
multiply
Enter numbers separated by spaces:
20 10
Result: 200.0
Choose an operation: add, subtract, multiply, divide, advanced, or type 'help' for help, 'exit' to quit
divide
Enter numbers separated by spaces:
10 2
Result: 5.0
Choose an operation: add, subtract, multiply, divide, advanced, or type 'help' for help, 'exit' to quit
advanced
Enter an expression:
5 + 2 * 10
Result: 70.0
Choose an operation: add, subtract, multiply, divide, advanced, or type 'help' for help, 'exit' to quit
```

Submission Guidelines

- Create a folder named “firstname_assignment1_java” and put the source folder into it.
- Create a subfolder called screenshots where to add the screenshots from the instructions.
- Zip the folder into a zip file and upload that file on the canvas submission page.
- An example of submission might look like this :
    - Assuming a student named Toby, **toby_assignment1_java** is the zipped file
    - The screenshot folder contains the **toby_screenshot1** file

Grading policies

Your submission will be evaluated on the following components :

- Functionality:
    - Does the submission contain the functions for addition, subtraction, multiplication, and division? (Your functions will be evaluated)
- Code quality:
    - Does the submission show a good usage of variable declarations, control flow, operator usage, error handling and code organization? (Your variable names, control flow, operator usage, error handling and code organization will be evaluated)
- Correctness of your solution:
    - Does the solution pass all the provided test cases? (Your solution will be tested on the provided test cases in the description section, but also with our custom test cases)