**PI DOCUMENTATION**

**CONTENTS**

1. Problem description
2. Explanation for solution

Description my solution to calculate the constant Pi

1. Design and Diagrams

* Class diagram details
* Classes description

1. How to build and run it
2. Pros and Cons the program
3. Orientation for the program
4. **Problem description**

My task is developing a program to calculate an approximation to the constant Pi formula. The solution to calculate Pi must use Java language to develop and judge on a number of criteria, including:

* It build and run actually produce an approximation to Pi.
* We can easily determine that the calculations are being done correctly.
* It’s using effectively the available resources as CPU, memory, etc.
* Another developer is able to pick up the solution and easily understand it.
* It easy to make changes to the code in the future.

1. **Explanation for solution**

Using the following formula to pi calculation:



First I will design formula to calculate pi like this:

**double**piNo = 0.0;

**for** (**long**x = startNo; x<= endNo; x++) {

**if** (x % 2 == 0) {

piNo += (1.0 / (2.0 \* x + 1));

} **else** {

piNo -= (1.0 / (2.0 \* x + 1));

}

}

I have a variable name is piNo it will storage sum of pi value return after formula completed pi is calculated.

Second I create a number of threads and each of them has information to calculate Pi as below:

startNo, EndNo  
PiNo=0.0  
x = 0

X <= endNo

PiNo-= (1.0 / (2.0 \* x + 1))

X % 2 == 0

PiNo += (1.0 / (2.0 \* x +1))

X++

Return PiNo = PiNo \* 4.0

When I have a formula below. I get available processor of the computer and depends on it, I will divide list of number as a number block which the block will put to callable to calculate pi, one number block will create one a callable. When the calculate process is finish callable return a result. I get return results and show on console for everyone can see what result are.

On the running time we can stop running immediately, program will stop callable and get return result printout console.

In this program I used strategy pattern for formula design and some singleton patterns to manager created object.

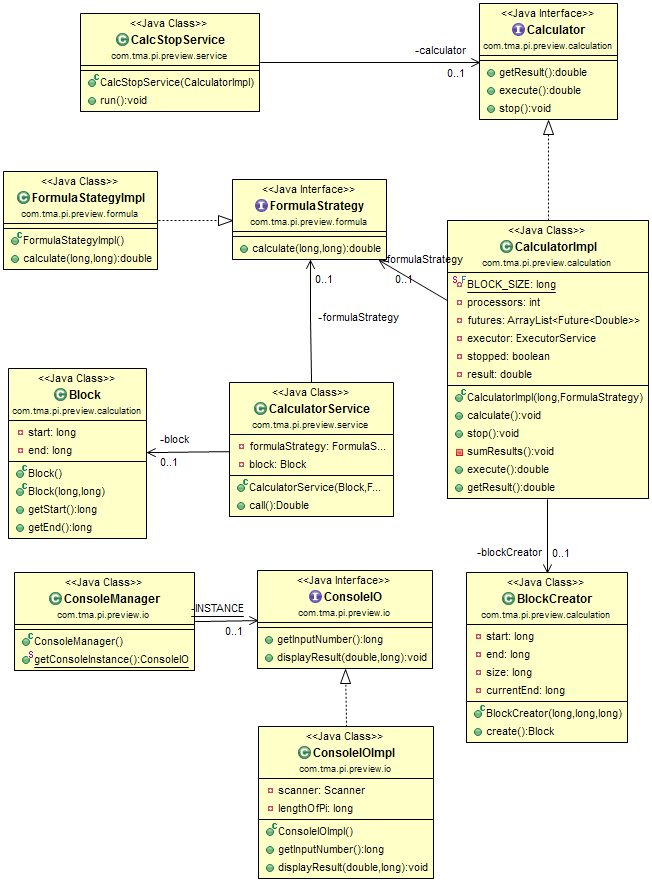
I also used maven to build my program and junit easily determine that calculations are done correctly.

Efficiency is very good because I separated a big list of number to be a small number block and used callable that is a type of thread to increase calculate speed.

Maintainability, it is very easy to add new formula to calculate pi, or change to another view type.. Because I used some of design pattern in the program make it easy.

1. **Design and Diagrams**

* Class Diagram



* List classes on the app

|  |  |
| --- | --- |
| **Class name** | **Description** |
| Block | Define bean number to pi calculate data |
| Calculator | The interface define methods to client access |
| CalculatorImpl | This class implement FactoryManager Interface |
| CalcStopService | Create a thread to stop calculate processing |
| BlockCreator | Define a service to call number bean |
| CalculatorService | Provide a callable service to calculate pi |
| FormulaStrategy | FormulaStrategy interface define |
| FormulaStrategyImpl | In this class implement business process to calculate pi |
| ConsoleIO | This interface define |
| ConsoleImpl | Implement input and output the data from console |
| ConsoleManager | Manager all of access from outside to consoleImpl class |
| StartUp | Define main class to start you application |

1. **How to build and Run**

To build the program you need to sure that JDK and Maven installed on your computer if it’s already. Please do some steps below to build jar file.

You can use commands below to check your JDK and maven on your machine.

From terminal on linux or unix and command line on Windows you type:

**Java –version** for maven check: **mvn –v**

You create a now folder where you have permission to access and copy folder src and pom.xml file into that folder and run command below to create jar file of the program.

mvn clean package [enter]

When the command finished with the messages is build successful have a target folder created in parent folder you cd into target folder you will see a jar file with the name is pi-exercise.jar

To run the program in the target folder you run the command java –jar pi-exercise.jar [enter] if there is no error appear it’s mean your program start already, you can enter length of the pi you want to calculates and press enter waiting some second you will have pi result as you want. You can press enter to stop calculate processing the program will return the result on the command line for you.

1. Pros and Cons the program

* Pros:

The program has calculate speed good, easy to test and maintenance

Another developer be able to pick up solution and easily understand

It can use of the available resources (Cpu, memory etc)

* Cons:

Because this is small application to demonstration so that I have no more time to implement with other web framework like spring, struts… Just have implemented with version console. (it doesn’t friendly for user)

Algorithm to calculate pi it’s slow need to find out other algorithm to increase calculate speed.

1. Orientation for program

* Implementation for spring web version
* Integrate with mockito test to sure the program run correctly
* Implement some formula to check how pattern do..