|  |  |  |  |
| --- | --- | --- | --- |
|  | Acronym | Concept | My application of the concept |
| S | SRP | Single Responsibility Principle | Double num1  Double num2  These values only have one reason, to store the numbers to perform the calculations to be done. |
| O | OCP | Open/closed Principle | Private double num1  Private double num2  We use private sometimes for this as we want to display the value, they first input so we display it somewhere else, and within the scope of this statement, we do the calculations. |
| L | LSP | Liskov substitution principle | Public Calculator() {  logger = new Logger();  logger.setLogLevel(LogLevel.Trace);  }  We in this example are using a constructor to make display the logger values for our calculations, if we were to use the regular class if should work aswell. |
| I | ISP | Interface segregation principle | Public double add  Public double substract  Public double multiply  Public double divide  We separate these into different function so that calculation don’t need to be done all in one calculation |
| D | DIP | Dependency inversion principle | @Test  Void add() {  }  We use the add class, create a abstract version of it and use it as a way to test our the processes. |