

## Klasa TestedClass

1. Napisać test jednostkowy sprawdzający poprawność działania dla funkcji Addition
  - a. Funkcjonalność: sumowanie
  - b. Zakres elementów wejściowych -100 do 100
  - c. Typ elementów wejściowych i wyjściowych: int
2. Napisać test jednostkowy sprawdzający poprawność działania dla funkcji AdditionWitNoNegativeResult
  - a. Funkcjonalność: sumowanie
  - b. Jeżeli obliczona suma jest mniejsza od 100 funkcja zwraca 0
  - c. Zakres elementów wejściowych -100 do 100
  - d. Typ elementów wejściowych i wyjściowych: int
3. Dla istniejącego testu funkcji Division dopisać ciało funkcji
  - a. Dzielenie
  - b. Zakres elementów wejściowych -100 do 100
  - c. Na podstawie testu określić i zaimplementować ewentualne dodatkowe właściwości funkcji
4. Napisać wrappera klasy TestedClass i test jednostkowy dla metody ProtectedFunction\_1
  - a. Funkcjonalność: sumowanie
  - b. Zakres elementów wejściowych -100 do 100
5. Napisać: wrappera klasy TestedClass, mocka klasy Parameters(wartości które może przyjmować zmienna ExponentionBase to: 2,3, 5) oraz test jednostkowy dla metody ProtectedFunction\_2
  - a. Funkcjonalność: potęgowanie
  - b. Zakres elementów wejściowych -100 do 100
6. Napisać: wrappera klasy TestedClass, mocka klasy Parameters(wartości które może przyjmować zmienna ExponentionBase to: 2,3, 5) oraz test jednostkowy dla metody ProtectedFunction\_3
  - a. Funkcjonalność: potęgowanie
  - b. Zakres elementów wejściowych -100 do 100