Klasa Calculations

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