

Ćwiczenia nr 2 – Testy parametryzowane

Wstęp

Celem ćwiczeń jest zapoznanie się z pojęciem testów parametryzowanych. Aby zacząć wykonaj następujące kroki:

1. Ściągnij projekt z poprzednich zajęć za pomocą instrukcji:
'git clone <https://github.com/anicos/quickstart.git>'
2. Przełącz się na branch 'parameterized_tests'
3. Ściągnij zależności używając komendy 'npm install'
4. Uruchom test używając komendy 'npm test'
5. W plik 'temperature-converter.service.spec.ts' znajdziesz przykład testu z parametrami
6. Rozwiąż poniższe zadania bazując na powyższym przykładzie

Zadanie 1. Pole trapezu.

Należy napisać klasę o nazwie 'TrapezoidFieldCalculatorService' obliczającą pole trapezu. Klasa powinna mieć metodę 'calculate', która jako argumenty przyjmuje dane niezbędne do policzenia pola, takie jak wysokość i długość boków. Metoda calculate powinna zwracać liczbę zmiennoprzecinkową. Uwaga! Jeśli jeden z boków lub wysokość jest mniejsze lub równe zero metoda powinna rzuć error.

Zgodnie z założeniami TDD pracę powinniśmy zacząć od napisania testów. Powinny być stworzone dwa zestawy testów z parametrami testujące następujące przypadki:

- a) wszystkie argumenty metody są poprawne – metoda zwraca poprawną wartość pola trapezu
- b) jeden lub więcej argumentów są nie poprawne – metoda rzuca error

Zadanie 2. Równanie kwadratowe

Analogicznie do zadania pierwszego należy stworzyć nową klasę o nazwie „QuadraticEquationCalculatorService”. Klasa powinna mieć metodę 'calculate', która jako argumenty przyjmuje współczynniki 'a', 'b' i 'c'. Metoda calculate powinna zwracać wyniki równia w postaci tablicy liczb zmiennoprzecinkowych. Jeśli delta jest mniejsza od 0 zwracana tablica jest pusta. Wszystkie wzory można znaleźć pod adresem [https://pl.wikipedia.org/wiki/R%C3%B3wnanie kwadratowe](https://pl.wikipedia.org/wiki/R%C3%B3wnanie_kwadratowe).

Zgodnie z założeniami TDD pracę powinniśmy zacząć od napisania testów.

Zadanie 3. Liczby rzymskie.

Wykorzystując parametryzowane testy napisać konwerter liczb rzymskich na arabskie. Dokładny opis zadania można znaleźć w dokumencie cw0 z poprzednich ćwiczeń. Rozwiązanie w javascript można znaleźć pod adresem <https://www.selftaughtjs.com/algorithm-sundays-converting-roman-numerals/>