Laboratorium 8

Otwarto: środa, 24 listopada 2021, 15:30
Wymagane do: środa, 24 listopada 2021, 17:05

Proszę napisać klasę `Complex`, reprezentującą liczbę zespoloną, która będzie posiadała następujące konstruktory:

- `Complex()` - tworzący liczbę 0+0i,
- `Complex(double Re)` - tworzący liczbę Re+0i,
- `Complex(double Re, double Im)`- tworzący liczbę Re+Im * i,

oraz metodę `toString()`. Następnie w klasie `ComplexTest` proszę napisać testy jednostkowe do każdego z konstruktorów, które będą sprawdzały poprawność utworzonych obiektów na podstawie łańcucha znaków zwracanego przez `toString()`.

Następnie proszę dodać "atrapy" metod (zwracające wartości 0+0i/nie robiące nic):
- `set(double Re, double Im)`,
- `setRe(double Re)`,
- `getRe()`,
- `setIm(double Im)`,
- `getIm()`,
- `mod()` - oblicza moduł liczby zespolonej,
- `conjugate()` - zamienia liczbę na jej sprzężenie,
- `opposite()` - zamienia liczbę na przeciwną,

a kolejno przygotować testy do każdej z metod, które zakończa się niepowodzeniem. Proszę zaimplementować ciała podanych metod i sprawdzić, czy testy zakończa się pomyślnie (jest to tzw. technika **test-driven development**).

Następnie proszę dodać analogicznie metody statyczne umożliwiające **dodawanie**, **odejmowanie**, **mnożenie** i **dzielenie** dwóch liczb zespolonych ze sobą, liczby zespolonej z liczbą typu `double` oraz liczby typu `double` z liczbą zespoloną (trzy metody dla każdej z operacji) i utworzyć dla nich testy jednostkowe sprawdzające ich działanie.

Proszę napisać przeciążenie metody `equals()` i napisać testy sprawdzające poprawność jej działania.

W pliku Main.java proszę umieścić przykładowy fragment kodu pokazujący działanie podstawowych funkcji - jego zawartość dzisiaj jest mało istotna, najważniejsze jest działanie przedstawione w testach jednostkowych.

Do generacji przykładów testowych można skorzystać ze strony WolframAlpha.

Uruchomienie testów jednostkowych z lini poleceń:

- należy pobrać JUnit Platform Console i umieścić pobrany plik .jar w tym samym katalogu co pliki źródłowe,
- kompilacja i uruchomienie testów odbywa się z wykorzystaniem poniższych komend:

```
mkdir out
javac -d out Main.java
javac -d out -cp out:junit-platform-console-standalone-1.8.1.jar ComplexTest.java
java -jar junit-platform-console-standalone-1.8.1.jar --class-path out --scan-class-path
```