

# Sprawozdanie

Temat:  
Serwer komunikatora

Termin:  
15-11-2015

Data oddania:  
22-112015

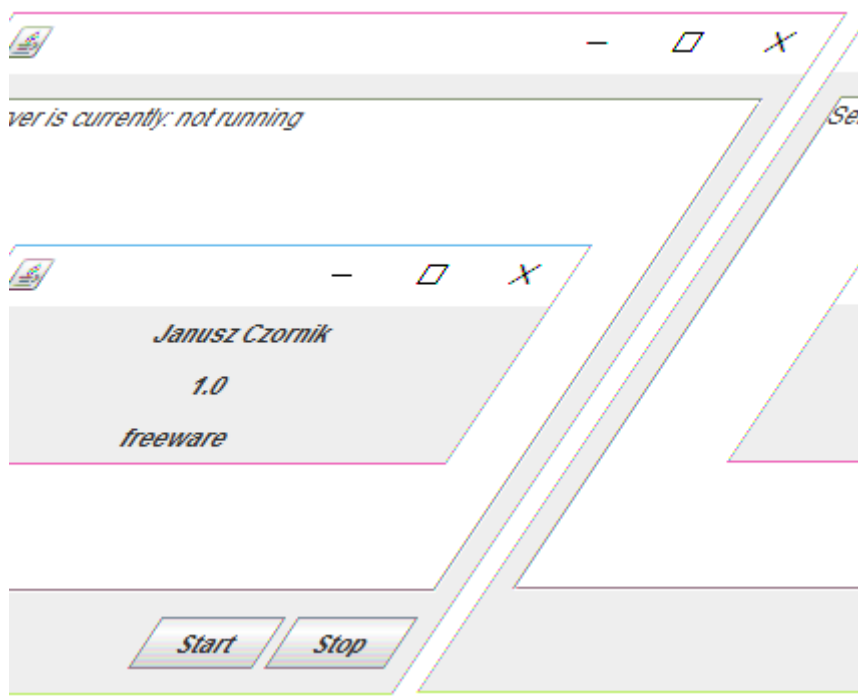
Punkty karne:  
1

Autor:  
Janusz Czornik

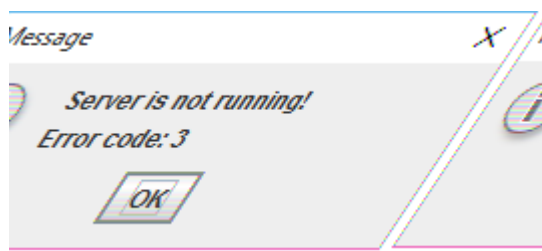
## Wymagania:

- pełny zakres raportu nr 1,
- dodanie typowobezpiecznej kolekcji obiektów do klasy modelu, (w ListeningForConnection pole newClientList)
- wykorzystanie standardowych adnotacji, (override, deprecated, SuppressWarnings("unused"))
- wykorzystanie pętli *foreach* (dwa razy raz w *ListeningForConnections* w metodzie *terminate* i raz w *InfoView* w metodzie *loadCpomponents()*)
- wykorzystanie jednego z następujących elementów:
  - typ wyliczeniowy, (w *StandardServerException* *ErrorCode*)
  - własny typ generyczny, (nie mam)
  - zmienna liczba parametrów metody, (w konstruktorze *StandardServerException*)
  - własna adnotacja, (w pakiecie *annotations*)
  - wyrażenie lambda. (w *terminateWithLambda*)
- zdefiniowanie testów jednostkowych dla wszystkich publicznych metod znajdujących się w modelu (za wyjątkiem konstruktorów, setterów i getterów)
- testy powinny obejmować sytuacje poprawne, niepoprawne i graniczne dla każdej testowanej jednostki,
- nie testujemy wartościami granicznymi z zakresu wartości typów prostych.

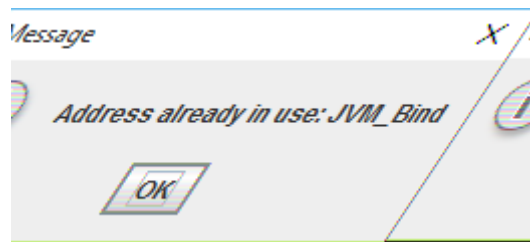
## PrintScreeny



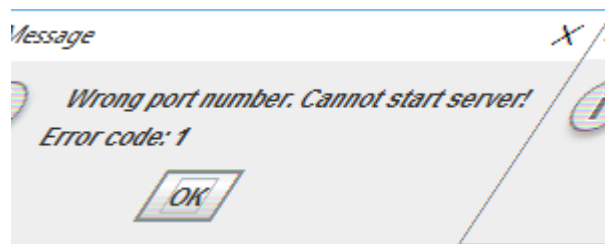
jeżeli się zatrzyma serwer, który nie działa wyskoczy błąd:



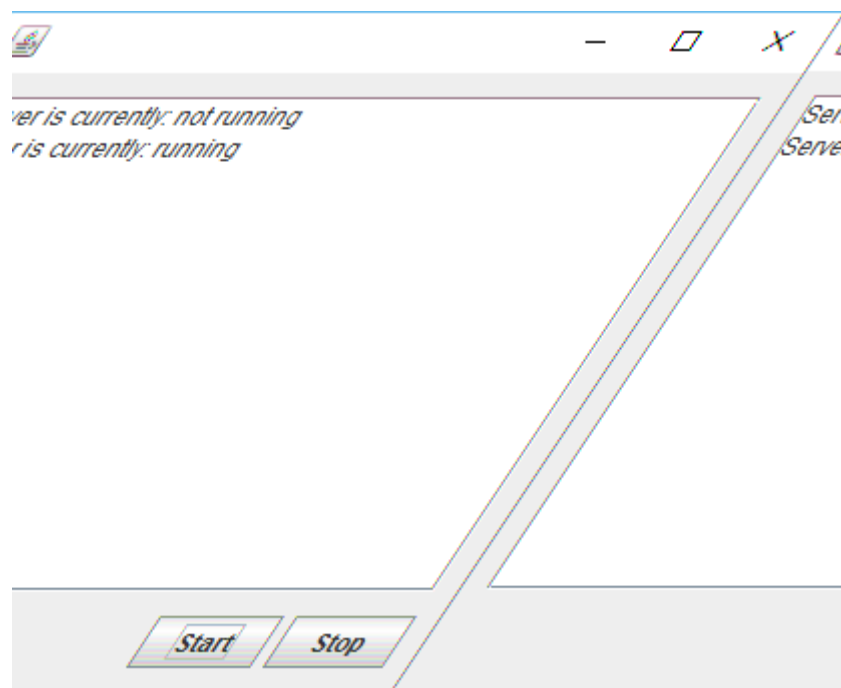
Analogicznie uruchomienie serwera, który już działa wywoła okno:

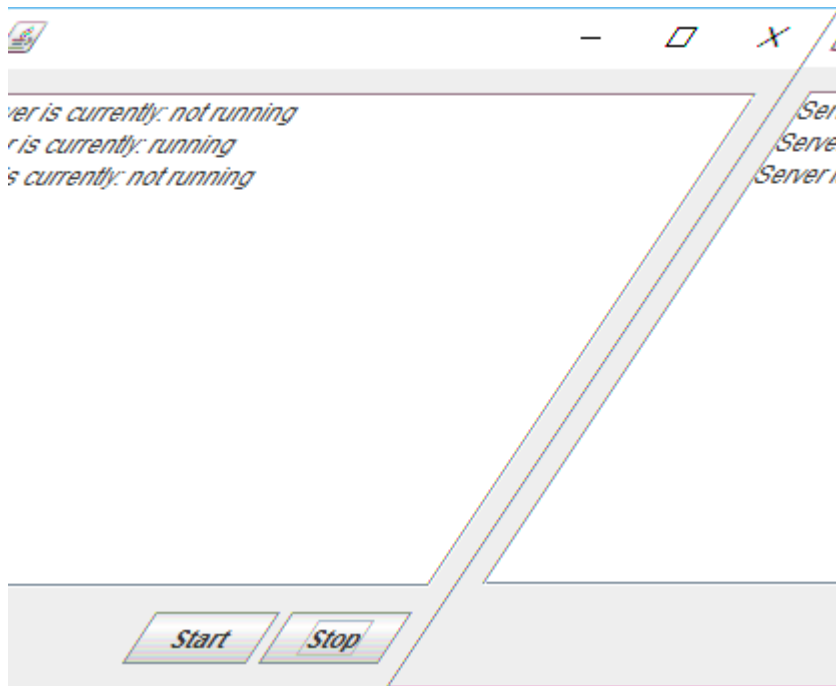


Uruchomienie serwera ze złym portem np. -100 skutkuje:



Następne screeny pokazują poprawne załączenie i wyłączenie serwera:





Wnioski:

Java nie jest przenośna jeżeli chodzi o testy. Na komputerze z Windows 10 `assertEquals` dla dwóch stringów działa poprawnie jednak na Mac OS wyskakuje błąd.