



UNIwersytet Gdański

Wprowadzenie do narzędzia Hamcrest

Mateusz Miotk

12 marzec 2018

Instytut Informatyki UG

Hamcrest - co to?

Narzędzie **Hamcrest** udostępnia wiele metod nazywanych matcherami, w których każdy z nich ma wykonywać określoną operację porównania.

Asercje oparte na narzędziu Hamcrest są bardziej zwarte. Ponadto Hamcrest udostępnia znacznie większy zestaw asercji, co pozwala uniknąć szablonowego kodu, a jednocześnie sprawia, że kod staje się bardziej czytelny i zwarty.

Hamcrest vs JUnit

Asercja z platformy **JUnit**:

```
Assert.assertEquals(5, friendships.getFriendsList("Joe").size())
```

Asercja z narzędzia **Hamcrest**:

```
assertThat(friendships.getFriendsList("Joe"), hasSize(5));
```

Różnice między Hamcrest a JUnit

Hamcrest w odróżnieniu od JUnit, prawie zawsze bezpośrednio korzysta z obiektów. W platformie JUnit trzeba uzyskać wielkość w postaci liczby całkowitej, a następnie porównać ją z oczekiwaną liczbą.

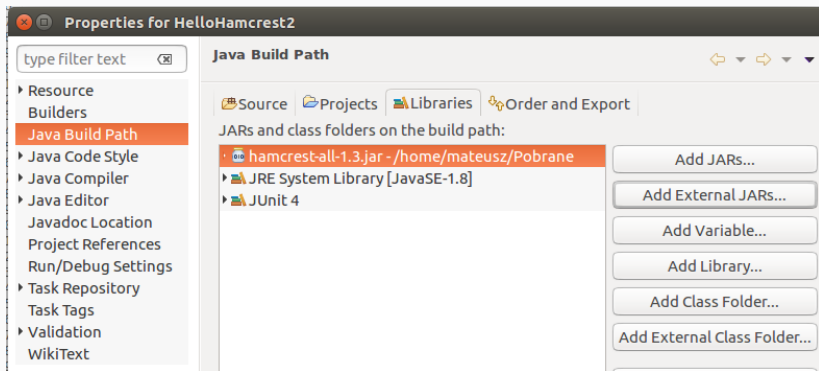
Argumenty podawane są w innej kolejności: pierwszym z nich jest rzeczywista wartość.

Hamcrest - instalacja

Domyślnie w JUnit4 jest okrojona wersja biblioteki Hamcrest.

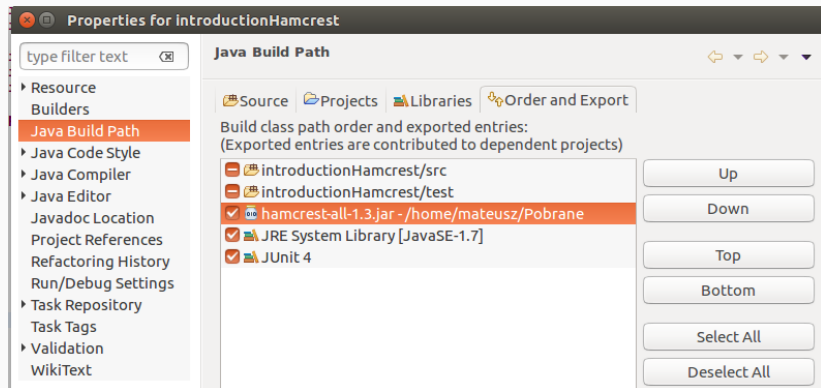
Jednakże, aby móc korzystać z pełnej funkcjonalności narzędzia Hamcrest należy ściągnąć odpowiednią paczkę (plik w formacie .jar), a następnie dodać ją do projektu lub w przypadku narzędzia Maven czy Gradle odpowiednio skonfigurować plik konfiguracyjny.

Przykład dodania pliku w środowisku Eclipse:



Dodawanie Hamcrest do projektu

Aby narzędzie Hamcrest działało poprawnie należy jego użycie wywołać przed JUnit (patrz rysunek poniżej):



Rysunek 2:

Przykładowy kod z użyciem narzędzia Hamcrest

Stworzymy program oraz testy do niego dla listy liczb całkowitych, w których będzie metoda wczytania i pobrania tej listy. Tworzymy projekt i dodajemy narzędzie Hamcrest (jak na slajdzie poprzednim). Następnie zgodnie z TDD piszemy pierwszy test:

```
// Te linie musimy dodać sami
import static org.hamcrest.MatcherAssert.assertThat;
import static org.hamcrest.Matchers.*;

public class SquareArrayTest {

    private SquareArray array;

    @Test
    public void getArray(){
        array.setList(Arrays.asList(1,2,3));
        List <Integer> tempList = array.getList();
        assertThat(tempList,contains(1,2,3));
    }
}
```

Klasa SquareArray

```
public class SquareArray {  
  
    private List<Integer> list;  
  
    public List<Integer> getList() {  
        return list;  
    }  
  
    public void setList(List<Integer> list) {  
        this.list = list;  
    }  
  
}
```


Inne narzędzia - AssertJ

Oprócz narzędzia **Hamcrest** istnieje również narzędzie **AssertJ**, które ma podobne zastosowania ale ponadto można w tym narzędziu łączyć asercję.

Asercje przy użyciu platformy JUnit:

```
Assert.assertEquals(5, friendships.getFriendsList("Joe").size());  
List<String> friendsOfJoe = Arrays.asList("Audrey", "Peter");  
Assert.assertTrue(friendships.getFriendsList("Joe")  
    .containsAll(friendsOfJoe));
```

Można połączyć przy pomocy AssertJ następująco:

```
assertThat(friendships.getFriendsList("Joe").hasSize(5)  
    .containsOnly("Audrey", "Peter"));
```



V. Farcic, A. Garcia, *TDD. Programowanie w Javie. Sterowanie testami*, Wydawnictwo Helion, 2016.



<http://hamcrest.org/>



<http://joel-costigliola.github.io/assertj/>