Testplan opdracht 1

Integration & communication

Edwin de Roos 1672073

2017

# Versie

|  |  |
| --- | --- |
| **Versie** | **Datum** |
| 1.0 | 22-06-2017 |
|  |  |

# Introductie

Dit is het testplan geschreven voor opdracht 1 van de herkansing van I&C. Er is een webservice gerealiseerd dat BMI kan berekenen.

# Testomgeving

De service die getest wordt is genaamd BMISoapService. Deze service berekent het BMI op basis van een gegeven lengte en gewicht. De service draait op een tomcat server en kan worden getest in SoapUI.

# Smoketest

|  |  |
| --- | --- |
| **Inputwaarden** | **Verwachte uitkomst** |
| Length = k , weight = sd | Error:Numberformatexception |
|  |  |

# Blackbox tests

|  |  |
| --- | --- |
| **Inputwaarden** | **Verwachte uitkomst** |
| Length = 1.80 , weight = 80 | Een berekende BMI |
| Length = -1.80, weight = 90 | Error: inputwaarden moeten positief zijn |
| Length = 1.80, weight = -90 | Error: inputwaarden moeten positief zijn |
| Length = 1.80, weight = 0 | Error: gewicht moet groter dan nul zijn |

# Unit tests

Hieronder de source code van de unit tests gemaakt in java.

package test.java;

import static org.junit.Assert.\*;

import org.junit.Test;

import com.w3schools.www.EdwindeRoos.BMIvalues;

public class TestBMIvalues {

private BMIvalues bmi;

private float length;

private float weight;

public TestBMIvalues(){

length = 1.80f;

weight = 80f;

bmi = new BMIvalues(length,weight);

}

@Test

public void testGetLength() {

assertEquals(length,bmi.getLength(),0.01);

}

@Test

public void testSetLength() {

float newLength = 1.80f;

bmi.setLength(newLength);

assertEquals(newLength,bmi.getLength(),0.01);

}

@Test

public void testGetWeight() {

assertEquals(weight,bmi.getWeight(),0.01);

}

@Test

public void testSetWeight() {

float newWeight = 80f;

bmi.setWeight(newWeight);

assertEquals(newWeight,bmi.getWeight(),0.01);

}

}

package test.java;

import static org.junit.Assert.\*;

import org.junit.Test;

public class TestError {

public com.w3schools.www.EdwindeRoos.Error error;

public String input;

public String ErrorMessage;

public TestError(){

input = "Weight = -80, length = 1.80 ";

ErrorMessage = "Inputvalues should be positive";

error = new com.w3schools.www.EdwindeRoos.Error(input,ErrorMessage);

}

@Test

public void testGetInput() {

assertEquals(input,error.getInput());

}

@Test

public void testSetInput() {

String newInput = "test";

error.setInput(newInput);

assertEquals(newInput,error.getInput());

}

@Test

public void testGetErrorMessage() {

assertEquals(ErrorMessage,error.getErrorMessage());

}

@Test

public void testSetErrorMessage() {

String newErrorMessage = "dit is fout";

error.setErrorMessage(newErrorMessage);

assertEquals(newErrorMessage,error.getErrorMessage());

}

}