Dagens program

- Studieintroduktion modul 3
- Opfølgning på sidste opgave
- Unit testing
- Test cases
- Opgave: test billetautomat med JUnit

Sidste opgave: test billetautomat

Hvilke fejl har I fundet?

(liste på Campusnet)

Unit testing

Et teknisk produkt, f.eks. et stykke software, består typisk af flere dele. Når man udvikler produktet er det fornuftigt at teste hver del for sig. Dette kaldes *unit test*.

Til sidst sætter man alle delene sammen og tester det samlede produkt. Dette kaldes *integration test*.

Hvis man laver integration test uden forudgående unit test kaldes det *big bang test*. Dette går næsten aldrig godt!

Automatiseret test

Hvis et software produkt ofte bliver ændret eller videreudviklet er det en fordel at automatisere test-processen.

I Java kan vi bruge et framework der hedder JUnit til automatisk test. Dette er et stykke kode der tester et andet stykke kode.

Det er vanskeligt at teste et brugerinterface med JUnit, især hvis det er et grafisk interface der bruger mus.

Lav brugerinterfacet i en klasse for sig, adskilt fra den del af programmet der laver beregninger.

Denne video viser hvordan man laver JUnit test i Eclipse:

https://www.youtube.com/watch?v=v2F49zLLj-8

NB. Videoen kalder den klasse de tester for Junit. Dette er et forvirrende navn. Giv klassen et navn der afspejler dens primære funktion, ikke det faktum at vi vil teste den.

Her er en længere tutorial om JUnit testing for de særligt interesserede:

http://www.vogella.com/tutorials/JUnit/article.html

Eksempel: beregning af cylinder rumfang med tilhørende JUnit test: cylindertest.zip.

Formelle test cases

Test case ID	TC01
Summary	Test that calculated volume is correct
Requirements	Requirement specification RS01 and RS03
Preconditions	A user has started the program
Test procedure	1. Choose menu option "calculate volume". 2. Enter a value for radius. 3. Enter a value for height. 4. Read out the result.
Test data	radius = 0.1 cm, 5.0 cm, 30.0 cm. height = 0.1 cm, 8.0 cm, 100 cm.
Expected result	The correct value is shown with the correct measuring unit and the correct number of decimals.
Actual result	0.0031, 628.3185, 282743.3388 cubic centimeters.
Status	Passed.
Tested by	Agner Fog.
Date	2016-09-14.
Test environment	Eclipse 4.6.0 on Windows

Test case ID	TC18
Summary	Test handling of overflow
Requirements	Not specificed
Preconditions	A user has started the program
Test procedure	1. Choose menu option "calculate volume". 2. Enter a value for radius. 3. Enter a value of height. 4. Read out the result.
Test data	radius = 1E150 cm. height = 1E150 cm.
Expected result	Error message indicating overflow.
Actual result	The volume is Infinity cubic centimeters.
Status	Failed.
Tested by	Agner Fog.
Date	2016-09-14.
Test environment	Eclipse 4.6.0 on Windows.

Traceability matrix

Test case	Requirement number			
	RS01	RS02	RS03	RS04
TC01	X		X	
TC02				
TC03			X	
TC04	X		X	X
TC05			X	
TC06				X
TC07	X			X

Hvad kan vi se ud af denne matrix?

Opgave:

Lav en JUnit test suite til billetautomaten der tester alle metoderne i klassen Billetautomat, undtagen metoden udskrivBillet().