Programmering 2: om rättning och testning av inlämningsuppgiften, del 2

Version: 1.0
Uppdaterad: 2021-05-13, se dokumenthistoriken sist i dokumentet för info.
Författare: Patrick Wentzel

Det här dokumentet beskriver rättningen och hur den automatiserade testningen av inlämningsuppgiftens andra del går till.

Dokumentet och den refererade filmen visar hur det fungerar när dokumentet författades och kan bli inaktuell i takt med att testerna förändras.

Innehållsförteckning

Inledning

Eftersom kursen har ett stort antal deltagare och varje uppgift skulle ta mellan 5-15 minuter att testa funktionellt (dvs att alla olika interaktioner med gränssnittet får förväntat resultat), så tar redan den funktionella testningen (som normalt har gjorts en enda gång efter deadline) av inlämningarna potentiellt flera veckor, plus tid för att granska kod etc.

Precis som tidigare uppgifter genomförs en del av testningen automatiskt. Motiven till att använda automatiserad testning är flera:

- att frigöra tid som kan användas till t.ex. undervisning och handledning.
- att möjliggöra snabbare återkoppling och stödja ett iterativt arbetssätt och inkrementell utveckling.
- att få en ökad rättssäkerhet i bedömningar - testprogrammet bedömer allt lika.
- att ta bort ett oerhört tidskrävande mekaniskt moment

Rättningen

Till skillnad från övningsuppgifterna och del 1 av inlämningsuppgiften godkänns inte del 2 automatiskt i VPL utan uppgiften rättas även manuellt av någon av kursens medarbetare. I samband med rättning görs även plagiatkontroller.

Testerna i VPL är tänkta att kontrollera att stil- och kodkonventioner följs (precis som med de tidigare uppgifterna), samt försöka testa att programmet fungerar som det ska.

Om testerna godkänns kommer inlämningen få betyget "Klar för manuell rättning" och då kommer den att rättas manuellt, senast tre veckor efter deadline.

Testningen i VPL

Filmen som hör till dokumentet visar hur en körning av testerna ser ut, med 2 sekunder långa pauser inlagda mellan varje steg i testerna. I verkligheten går testningen lite snabbare och tar ungefär 31 sekunder.

Testerna är en form av black-box testning, dvs. tester som utförs utan tillgång till källkoden, utan bara med vetskap om vilka element som borde finnas i gränssnittet (namngivna knappar och menyer) och vad som borde hända i olika situationer.

Eftersom uppgiftsbeskrivningen inte ställer (så många) specifika krav på inlämningen är testprogrammet tvunget att gissa hur t.ex. dialogrutor är uppbyggda, och det händer att det blir fel. Vi försöker löpande följa upp hur testerna fungerar och förbättra funktion och felmeddelanden.

För den som är intresserad så är testerna skrivna som enhetstester för JUnit 5 och använder ramverket TestFX för att testa det grafiska gränssnittet.

I skrivande stund körs 10 funktionstester med följande steg:

1. test01_check_title

Kontroll av att titel är satt till PathFinder.

2. test02_new_map

En ny karta laddas in. Verifieras genom att undersöka att bilden har rätt URL och bredd och höjd, samt att grafen är tom.

3. test03_new_place

Flera nya platser skapas. Verifieras genom att se att de går att hitta på id, samt genom att titta i grafens noder och antalet barn till komponenten med id outputArea. Det kontrolleras också att platserna går att markera och avmarkera.

4. test04_new_connection

En ny förbindelse etableras. Verifieras genom att kolla i grafen att bågen finns. Testar också att minst två noder måste vara markerade och om bågen redan finns.

5. test05_open

Den sparade filen europa.graph laddas in. Verifieras genom att kolla i grafen, på outputArea och att noder är klickbara.

6. test06_save

En ny karta laddas och ett par noder och en båge läggs till. Filen sparas sedan och innehållet jämförs med facit. En ny karta laddas in och filen laddas in. Verifieras att grafen och outputArea innehåller rätt platser.

7. test07_save_image

En bild sparas. Kontrolleras genom att se att bilden finns.

8. test08_find_path

Två noder markeras och sen kontrolleras att Find Path innehåller rätt information.

9. test09_show_connection

Två noder markeras och sedan kontrolleras att Show Connection visar rätt information för bågen mellan de.

10.test10_change_connection

Två noder markeras och sedan kontrolleras att Change Connection leder till ändringar av vikt på båge. rätt information för bågen mellan de.

Dokumenthistorik

Version	Datum	Ändringar
1.0	2021-05-13	Första version.