Project-INTE

Introduktion till testing av it-system 2015-10-30

Niklas Anner	adal1234
Torbjörn Håkansson	bebe5678
Daniel Karlsson	caga9012
Pontus Valldor	dide3456
Sebastian Wärnström	

Link till projektet på GitHub: https://github.com/Sebwar/Project-INTE

Introduktion

En kort introduction till projektet där ni också listar de verktyg ni använt. Om ert versionshanteringssystem går att komma åt ska adressen dit finnas med, annars ska det finnas en länk från vilken man kan ladda hem källkoden.

Vi har valt att implementera och testa ett kassasystem. Systemet är anpassat och avgränsat till att användas för en liten butik som säljer dagligvaror, typ en kiosk. Vid köp av en eller flera varor ska ett kvitto skrivas ut där varans namn, pris och eventuella rabatter anges. Om kunden vill ha sitt namn, adress och organisationsnummer angivet (för tex bokföringsändamål) ska även detta finnas med på kvittot.

De verktyg vi använt är:

Språk/IDE: **Java/Eclipse**

Ramverk för enhetstestning: **JUnit**

Versionshantering: **GIT**

Byggsystem: **Gradle**

Kodkritiksystem: **FindBugs**

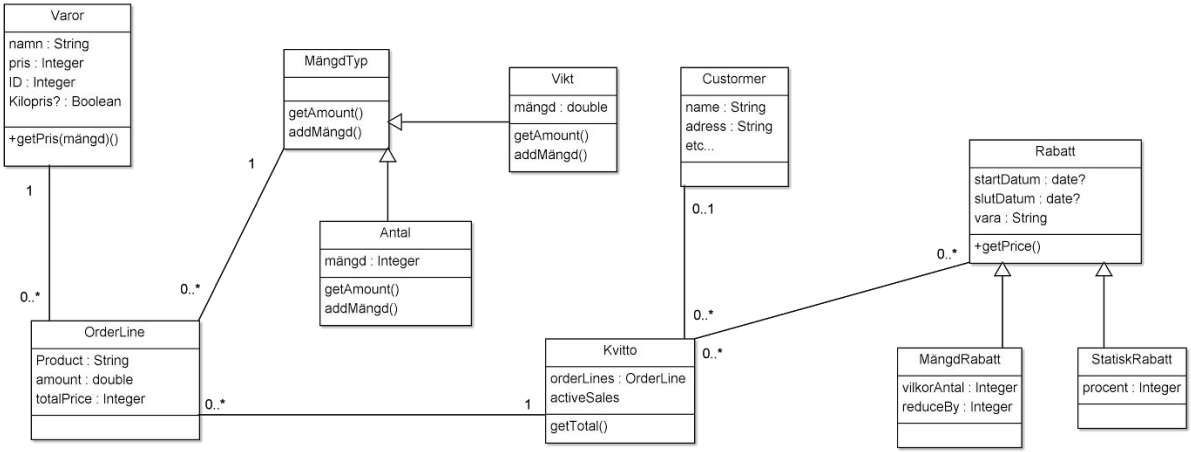
Statiska mått: **Metrics**

Täckning/code coverage: **EclEmma**

Profiler: **Eclipse+jvmmonitor**

Kommunikation: **Facebook, Google Docs, email, telefon, Skype och fysiska möten.**

Slutlig design



En övergripande modell över systemet. Lämpligt format är ett eller flera klassdiagram, plus eventuella andra modeller som behövs för att förstå hur systemet är uppbyggt. Diagrammen ska vara läsbara. Det är dock fullständigt okej att de är

detaljerade, bara det går att zooma in ordentligt på dem. Ett tips är att börja med ett översiktligt diagram som inte innehåller mer än paket och klassnamn, och att sedan lägga till mer detaljerade diagram efter det.

Coupon: Klassen används vid inlösning av en kupong. Om totalsumman är över ett visst värde dras kupongvärdet av.

Customer: Klassen används om en kund vill ha ett kvitto med kunduppgifter för tex bokföring.

Testdriven utveckling – process

En översikt över hur ni tillämpat TDD med exempel från olika personer och olika faser i projektet. Om ni har använt versionshanteringssystemet ordentligt bör all information som efterfrågas här finnas i det. Tänk på att kodexemplen ska vara läsbara.

Vi tillämpar oss utav TDD när vi skapar klasser som inte endast ska hålla data, som ska kunna hanteras med olika publika "get" och "set" metoder. När vi har använt oss av TDD har vi börjat med att skriva tester innan vi skrivit någon funktionell kod. På det sättet får vi automatiskt full statement coverage, samt blir det lättare att skriva bättre kod. I början när vi skriver den funktionella koden, skriver vi endast tillräckligt för att testen ska kompilera. Därefter bygger vi ut koden så att den integreras med hela klassen.

Testerna skapade vi med hjälp av specifikationerna som vi skapade vid våra möten, där vi gemensamt jobbade på systemets design.

Testdriven utveckling – erfarenheter

En diskussion om vilka era erfarenheter ni dragit av att tillämpa TDD. Det finns inget rätt eller fel här. Enda sättet att bli underkända är att bara fuska över punkten och säga något pliktskyldigt.

Ekvivalensklassuppdelning – namn på del

En kort presentation av vad ni valt ut för att tillämpa ekvivalensklassuppdelning på. Ni ska kort motivera valet, och ge tillräckligt med information för att det ska gå att bedöma er. Detta avsnitt och de tre följande (till och med testmatrisen) ska finnas för samtliga delar ni tillämpat ekvivalensklassuppdelning på.

Ekvivalensklasser – namn på del

Samtliga ekvivalensklasser för denna del presenterade på ett tydligt sätt.

Testfall – namn på del

Testfallen som ni fått fram från ekvivalensklasserna. Observera att vi inte vill ha någon kod här, utan bara en tydlig presentation av testfallen i någon lämplig tabellform.

Testmatris – namn på del

En testmatris som visar sambandet mellan ekvivalensklasserna och testfallen för denna del.

Tillståndsbaserad testning

En kort presentation av vad ni valt ut för att tillämpa tillståndsbaserad testning på och vilket täckningskriterium ni valt att använda er av. Ni ska kort motivera valen, och ge tillräckligt med information för att det ska gå att bedöma er. Glöm inte att ta med själva modellen.

Testfall för tillståndsbaserad testning

Testfallen som ni fått fram från tillståndsmaskinen. Observera att vi inte vill ha någon kod här, utan bara en tydlig presentation av testfallen i någon lämplig tabellform. Det ska enkelt gå att mappa testfallen till tillståndsmaskinen.

Granskning

En kort presentation av den del av koden ni valt ut för att göra en formell granskning av och processen ni använt er av inklusive eventuella checklistor, scenarier, edyl. Ni ska kort motivera valen, och ge tillräckligt med information för att det ska gå att bedöma er.

Granskningsrapport

En lista över de påträffade felen och hur pass allvarliga ni bedömer dem.

Granskning – erfarenheter

En diskussion om vilka era erfarenheter ni dragit av att tillämpa granskning. Det finns inget rätt eller fel här. Enda sättet att bli underkända är att bara fuska över punkten och bara säga något pliktskyldigt. Ni förväntas förhålla er till såväl kursboken som utdelat material och IEEE Std 1028.

Kodkritiksystem

En presentation av de problem som hittats med hjälp av verktyg för statisk analys och en diskussion av dem enligt anvisningarna. Det räcker alltså inte med att bara lista problemen, ni måste förhålla er till dem också. Tänk också på att ni ska göra detta både på koden som den såg ut före granskningen och på koden efter att ni rättat det som kommit fram under granskningen.

Statiska mått

En presentation och diskussion kring ett antal lämpliga statiska mått på koden. Att vi inte specificerar exakt vilka mått som ska tas upp beror på att olika verktyg har olika uppsättningar, men vi förväntar oss fler och mer intressanta mått än bara rena storleksmått som LOC, #klasser, #metoder, etc. Även här är det viktigt att förhållas sig till måtten, inte bara lista dem.

Täckningsgrad

En översikt över vilken täckningsgrad era testfall uppnått. Denna kan antagligen tas rakt av från verktyget ni använt för att mäta den. Om ni inte uppnått fullständig täckning så ska detta förklaras och motiveras.

Profiler

En kort presentation av hur ni gått tillväga för att testa koden med en profiler och vilka resultat ni fick fram.

Byggscript

Byggscriptets första (seriösa) version, och den slutliga.

Övrigt

Här kan ni ta upp övrigt av relevans för bedömningen av ert arbete. Om avsnittet inte behövs kan det plockas bort.