

Objekt med rätt beteende



2.1

Huvudbegrepp att diskuteras

- Testning av program
- Felsökning
- Automatisering av testning
- Hur man underlättar underhåll av program.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jakob Mälic

2

Fel gör man ...

- Enkla fel är ofta syntaxfel.
 - Kompilatorn hittar dem.
 - Rättas snabbt.
- Knepigare fel är de logiska (semantiska) felen (logic errors i kursboken).
 - Kompilatorn kan inte hjälpa till.
 - Kräver ofta att man måste analysera vad programmet gör under exekveringen..
- En del logiska fel visar sig inte medetsamma ...
 - Det finns få komplexa program som är helt felfria.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jakob Mälic

3

Förhindra vs. Upptäcka (Utvecklare vs. Underhållare)

- Vi kan minska sannolikheten för fel.
 - Använd bra programmeringsmetodik, t.ex. inkapsling.
- Vi kan öka sannolikheten för att upptäcka fel.
 - Använd bra programmeringsmetodik, t.ex. modularisering and dokumentering.
- Man kan träna sin förmåga att hitta fel.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jakob Mälic

4

Test- och felsökningsmetodik

- Helt nödvändig för större projekt .
- Systematiska tester upptäcker att något är fel.
- Felsökning syftar till att upptäcka vad som är orsaken till felet.
 - Ett fel behöver inte uppträda i anslutning till den felaktiga programkoden.



Test- och felsökningsmetoder

- Varje enhet testas för sig (finns stöd i BlueJ).
- Automatisering av tester.
- Manuell genomgång av programkoden.
- Testutskrift.
- Felsökningsverktyg (debugger).



Test av enheter

- Varje kodenhet i en tillämpning kan testas separat.
 - Metod, klass, modul (paket i Java).
- Under programutvecklingen:
 - Testsekvenser bör konstrueras fortlöpande. (Om man hittar och korrigerar fel tidigt under ett projekt sparar man kostnader.)
 - Används också senare.



Viktiga testprinciper

- Förstå vad en enhet skall göra – dess kontrakt.
 - Leta efter brott mot kontraktet.
 - Använd positiva och negativa tester.
- Testa gränser.
 - Noll, En, Full.
 - Sök efter element i tomma kollektioner.
 - Lägg till ett element i fulla kollektioner.



Enhetstester med BlueJ

- Objekt av klasser i projektet kan skapas och användas för tester.
- Metoder kan anropas med testparametrar.
- Inspektorn ger möjlighet till att kontrollera ett objekts aktuella tillstånd.
- Projektet diary-prototype är tänkt att användas för att öva sig i testmetodik och felsökning.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

9

Testautomatisering

- Bra uttestning är en skapande process, men ...
- ... är tidsödande och ofta måste man upprepa samma sak.
- Regressionstest innebär att man sparar en testsekvens för att köra den vid ett annat tillfälle (när koden har ändrats).
- En testtrigg kan spara mycket tid.
 - Man skriver speciella klasser som används för att testa programmet.
 - En kreativ process!



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

10

diary-prototype projektet

- Har fältet appointments tillräckligt många platser för de bokningar som kan behöva göras?
- Skriver metoden showAppointments ut en korrekt lista över de bokningar som gjorts?
- Uppdaterar metoden makeAppointment fältet appointments på ett korrekt sätt när en bokning görs?
- Returnerar metoden findSpace ett korrekt resultat när den anropas för att hitta ett rum för en ny bokning?



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

11

Test av findSpace

- Kan vi hitta en ledig plats på en obokad dag? (positiv test)
- Kan vi hitta en ledig plats på en delvis bokad dag? (positiv test)
- Kan vi hitta en ledig plats på en fullbokad dag? (negativ test)
- Kan vi hitta plats för två timmar när det inte finns två lediga konsekutiva timmar? (negativ test)



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

12

Testautomatisering

- Undersök diary-testing projektet.
 - Mänsklig tolkning av resultat fortfarande behövs.
- Undersök mer komplet automatisering i diary-test-junit projekten.
 - Man måste ingripa endast om fel rapporteras.

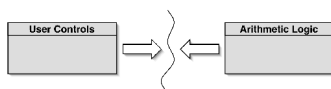


Modularisering och gränssnitt

- En stor tillämpning måste ofta delas upp i delar (moduler).
 - Varje modul implementeras av olika (lag av) programmerare.
- Det blir då väldigt viktigt att gränssnitten mellan de olika modulerna är väl specificerade.
 - Innebär att implementeringen av delarna kan ske oberoende av varandra.
 - Ökar sannolikheten för att problemen vid sammanslagningen blir hanterbara.



Modularisering i en kalkylator



- Den ena modulen behöver inte känna till implementeringen av den andra.
 - 'User controls' skulle kunna använda ett grafiskt gränssnitt eller hårdvara.
 - 'Arithmetic Logic' kan implementeras i mjukvara eller kan använda hårdvara.



Metodsignaturer specificerar gränssnittet (interface)

```
// Return the value to be displayed.
public int getDisplayValue();

// Call when a digit button is pressed.
public void numberPressed(int number);

// Call when a plus operator is pressed.
public void plus();

// Call when a minus operator is pressed.
public void minus();

// Call to complete a calculation.
public void equals();

// Call to reset the calculator.
public void clear();
```



Felsökning

- Om man skall bli en snabb och säker programmerare gäller det att vara tränad i felsökning.
 - Ofta får man söka fel i andras kod.
- Använd bra felsökningsverktyg!
- Prova själv BlueJ:s felsökare på 'calculator'-projektet.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

17

Manuell genomgång

- Bra komplettering till användning av en felsökare.
 - Bör speciellt göras innan man provkör första gången.
- Regel: Sitt inte vid datorn utan skriv ut programlistor!
- Exekvera programmet i huvudet med hjälp av anteckningar för att hålla reda på "var man är".
- Genomgång på hög nivå (Step) eller låg nivå (Step into).



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

18

Bokför vad som händer med objekten (tabulation)

- Ett objekts beteende beror på dess tillstånd.
- Felaktigt beteende beror ofta på att tillståndet är felaktigt.
- Bokför vilka värden de olika fälten har.
- Skriv upp ev. tillståndsförändringar efter varje metदानrop.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

19

Verbal genomgång

- Förklara för någon vad programmet gör.
 - Om man har tur: Denne upptäcker felet.
 - Men - bara att man förklarar gör att man konkretiserar för sig själv.
- Det finns formaliserade processer att utföra genomgångar i grupp (eng. inspections).



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

20

Utskrifter

- Den mest populära tekniken.
- Inga speciella verktyg behövs.
- Stöds av alla programmeringsspråk.
- Kräver att man placerar utskrifterna på bra ställen.
- Utskrifterna kan bli omfattande!
- Man bör fundera på hur man på ett enkelt sätt kan slå på och slå av utskrifterna.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

21

Att välja en testmetodik

- Man måste vara medveten om existerande testningsmetoder.
- Välj en metodik lämpad till utvecklingsläget.
- Automatisera om möjligt.
 - Reducerar den jobbiga delen.
 - Reducerar mänskliga fel.
 - Underlättar (om)testning.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

22

Avlusare (Debuggers)

- Avlusare brukar vara språk- och omgivningsberoende.
 - BlueJ har en integrerad avlusare.
- Stödjer brytpunkter (breakpoints).
- Step och Step-into exekvering.
- Anropsekvens (stacken).
- Objektens tillstånd.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

23

Sammanfattning

- Fel finns i program.
- Bra mjukvaruutvecklingsmetodik kan reducera dess förekomst.
- Testning- och avlusningfärdigheter är grundläggande.
- Testa som vana.
- Automatisera testning om möjligt.
- Öva ett antal avlusningstekniker.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacek Malec

24