

Uppgift 3

3a.

Jag kommer att använda mig av Bottom-up strategin, då jag kommer att lägga mer extra tid för att implementera uppgiften. Anledningen till att jag behöver mer tid än planerat är för att jag inte har arbetat så mycket med programmering och kommer därför göra extra kontroller under implementeringen.

Jag tänker mig också en kod som fungerar och är körbar, men som inte uppfyller alla kraven, och att därefter bygga till det som saknas.

Jag börjar med att läsa igenom uppgiften och plocka ut de bitar som behövs för att få en körbar kod. Oftast brukar jag vänta med att läsa in koden från tangentbordet och bara använda mig av en

Sträng men i detta fall skriver jag den koden från början. Då kan jag kontrollera koden snabbt med att mata in "lol" för true eller två valfria tangenter för false.

Genom att snabbt kunna se om det är några kommer jag skriva ut de variabler som koden innefatta. Det gör det lättare att se var felet ligger.

Nästa steg är att göra en omvänd loop.

Värdena från loopen kommer att ges till en ny variabel.

Sedan skapa en if sats om dessa två värden är lika, och då returneras en text true eller false. Om koden körs som den ska kan jag lägga till metoden ToLower() för att vara säker på att de är gemener. Jag kan sedan snygga till koden vid behov.

Jag har ett färdigt dokument med färdig skrivna ord och meningar som kan testköras för att slippa att grubbla på detta under tiden jag kodar.

Ord som returnera true: lol, HeH, 222, nn nn

Ord som returnera false: jag är roy, leka, pl, PaA

Jag har planerat hela koden ganska precis som jag tänker implementera den. Planeringen tog 30 min och koden kommer att ta mindre tid då jag inte behöver fundera, vilket är väldigt tidskrävande.

Planerad tid 15

min testning 5

min sökning 0

verklig tid 20 min

Orsak till avvikelse

– Jag gav variabeln equal inget värde vilket tog någon minut att fatta

– Jag satte > på fel håll i loopen

3b.

Här kan jag inte ens använda Top-down strategin då jag har svårt att genomföra uppgiften på grund av brist på kunskap. Jag kommer använda mig utav Bottom-up strategin då jag kan ändra under planeringen och ge mig extra tid för att prova mig fram en fungerande kod som jag kommer få göra då jag har brist på kunskap

Jag börjar med att läsa igenom uppgiften och plocka ut de metoder och dylikt som ska implementeras i koden. Jag använder mig av den befintliga stegen som angavs i uppgiften och arbeta genom den.

Jag implementera de metoder, egenskaper och konstruktörer jag har kunskaper utav på ett simpelt vis och de metoder som jag inte kan får jag hitta en lösning på vägen.

Här tar min planering i stort sett slut då jag har ingen aning hur jag ska gå vidare. Jag har ingen aning om hur man implementera bråktal så jag får helt enkelt prova mig fram och söka mig fram till lösningar.

Testimplementeringen får jag också prova mig fram med men det är inte många rader kod för att hämta en class och köra den.

Tidsplaneringen kommer i stort sett vara mer av en gissning än en planerad tid då jag har brist på kunskap.

Planerad tid 6 tim

Verklig tid 20 tim

Orsak till avvikelse

– Jag hade ingen aning att uppgiften skulle vara så komplicerad.

Reflektera kring planering och genomförande

Här skiljer de båda planeringarna rejält både tidsmässigt och planeringsmässigt.

Första uppgiften var mycket simplare då jag har jobbat med dessa typer av koder förut och kunde där med planera in hur uppgiften skulle byggas upp.

Jag är nöjd med mitt tidsresultat fast det gick lite över tidsplaneringen men det är minimal tid som man får räkna med, då det är svårt att planera in exakt tid samt man kan inte räkna med de små stör moment runtomkring sig.

Det blev lite mindre fel i koden som gjorde att programmet inte kördes, dessa mindre fel känner jag att man får räkna med och behöver inte planeras in, planeringen hade iså fall tagit väldigt lång tid för väldigt små detaljer.

Den andre uppgiften skiljer sig från den andre uppgiften då det är en mycket större uppgift samt mina kunskaper fallera här.

Uppgiften i sig är väldigt svår och den har fler sidor moment som ska implementeras i koden som försvårar planeringen ytterligare.

Jag planerade in de metoder, egenskaper och konstruktörer som uppgiften hade med i kravspecifikationen och de metoder jag inte visste hur jag skulle implementera p.g.a. brist på kunskap lämnade jag kvar tomma.

Detta gjorde att jag inte kunde genomföra en fullständig planering utan fick i efterhand fylla i och ändra i de metoder som lämnades kvar tomma.

Jag är inte nöjd med min tidsplanering men den var mer av en gissning än en planerad tid.