1DV404 Laboration 1

# Planering och förbättring

av Oskar Klintrot, oklib08

## Uppgift 1

### Planering av uppgift 1a

|  |  |  |
| --- | --- | --- |
|  | Steg | Tidsåtgång: 1 h 30 min |
|  | Inmatning till sträng | 15 min |
|  | Skapa en metod för att jämföra hur många ”a” och ”A” det finns i strängen. | 45 min |
|  | Skriv ut resultatet till användaren. | 15 min |
|  | Reflektion | 15 min |

### Utförande av uppgift 1a

|  |  |  |  |
| --- | --- | --- | --- |
| Steg | Fel vid programmering | Brister i planeringen | Tidsåtgång: 45 min |
|  | Inga | Nej | 5 min |
|  | Skapade en klass för metoderna, vilket var mer overkill än det lilla dry som blev i denna lilla applikation. | Behövde skapa två olika metoderna. | 20 min |
|  | Inga | Nej | 10 min |
|  | N/A | N/A | 10 min |

### Reflektion kring uppgift 1a

Planeringen gick någorlunda bra, enda tabben var att jag behövde göra två metoder och inte bara en. I utförandet kunde jag från första början hållit mig till min ursprungliga tanke att skapa metoderna direkt under Main. Hade jag repeterat mer från C#-kursen hade jag mer troligen gjort rätt från början. Detta misstag tog dock så lite tid i anspråk att det i praktiken antagligen hade gott på ett ut, om inte att det tagit längre tid att läsa på innan jag började koda.

Att första steget skulle gå fortare än planerat var väntat men det var sagt att planeringen skulle vara i steg om 15 min. Jag gillar att ha lite slack i planeringen men i detta fall var det kanske lite onödigt att ha med det i ett eget steg på 15 min. Att skapa metoderna gick mycket snabbare än planerat, mycket tack vare att jag snabbt hittade strings replace-funktionen i kurslitteraturen till C#-kursen. Hade jag behövt leta mer, i boken eller på nätet, hade det troligtvis genast tagit mer tid i anspråk.

### Planering av uppgift 1b

|  |  |  |
| --- | --- | --- |
|  | Steg | Tidsåtgång: 2 h 15 min |
|  | Inmatning till int | 15 min |
|  | Skapa klass för nedanstående metoder. | 15 min |
|  | Skapa metod för att se hur många ”0” som finns i talet. | 30 min |
|  | Skapa metod för att se hur många udda tal som finns i talet. | 30 min |
|  | Skapa metod för att se hur många jämna tal som finns i talet. | 15 min |
|  | Skriv ut resultatet till användaren. | 15 min |
|  | Reflektion | 15 min |

### Utförande av uppgift 1b

|  |  |  |  |
| --- | --- | --- | --- |
| Steg | Fel vid programmering | Brister i planeringen | Faktisk tidsåtgång |
|  | Tankevurpa med klassens egenskaper. Anropade objektet felaktigt. Problem med att ta hand om felaktig inmatning. | Gjordes som #2. | 35 min |
|  | Inga fel men behövde läsa på en del för att få allt rätt. | Implementerades som #1. | 15 min |
|  | Problem med omvandlingen från int till string, börjar talet på 0 försvinner första nollan vid omvandligen. | Problem med felhantering. Miss att inte tänka på att inmatningen måste vara string för att få med inledande nolla. | 35 min |
|  | Problem med omvandlingarna mellan olika arrayer och problem med gammal kod som störde. |  | 40 min |
|  | Inga. Ändrade dock om i koden så att alla metoderna bakades ihop till en enda då koden för udda tal och jämna tal var mer eller mindre exakt samma, enbart ett ”!” skiljde dem åt. | Koden var i stort sett identisk med #4, vilket inte är bra med tanke på DRY. | 20 min |
|  |  |  |  |

### Reflektion kring uppgift 1b

Bla bla

### Planering av uppgift 1c

|  |  |  |
| --- | --- | --- |
|  | Steg | Tidsåtgång: 1 h 15 min |
|  | Läs in 3 heltal till var sin int. | 15 min |
|  | Skapa en metod som jämför talen med varandra och retunerar dem i ordning. | 45 min |
|  | Skriv ut det näst största talet till användaren. | 15 min |
|  |  |  |
|  |  |  |

### Utförande av uppgift 1c

|  |  |  |  |
| --- | --- | --- | --- |
| Steg | Fel vid programmering | Brister i planeringen | Faktisk tidsåtgång |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Reflektion kring uppgift 1c

Bla bla

Tidslogg

