# Integrationstestning

**Stegvis beskrivning för hur du arbetade fram testfallen och testdatan.**

För att registrera en användare krävs det att valid data av namn-, lösenord-, och email matas in,

eller för poängregistrering krävs valid data för namn och poäng.

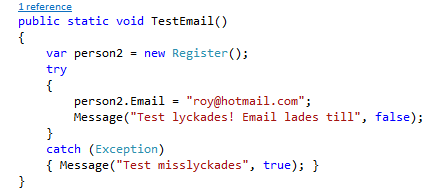
Jag valde att kontrolera hur regestrering och poängregistrering betede sig när de fick data inmatade som systemet skulle godkänna och inte godkänna.

Jag delade upp de olika testfallen i klasser (Registrering, poängregistrering) och kappslade in varje testfall i en metod och skapade en ny instans objekt av varje testfall.

För att kontrolera om testfallet är godkänt eller inte godkänt använder jag mig av try-cath satser för att få ut ett resultat med hjälp av en annan metod som returnera boolesk värde sant eller falskt.

Alla reultat renderas ut i konsolfönstret.

En visuel bild av hur en testmetod ser ut.

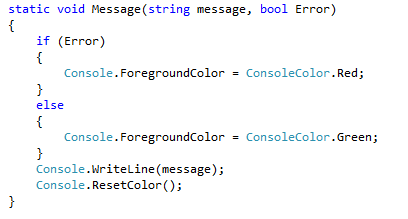


Bilden testar om en ny användare kan mata in email adressen [roy@hotmail.se](mailto:roy@hotmail.se)

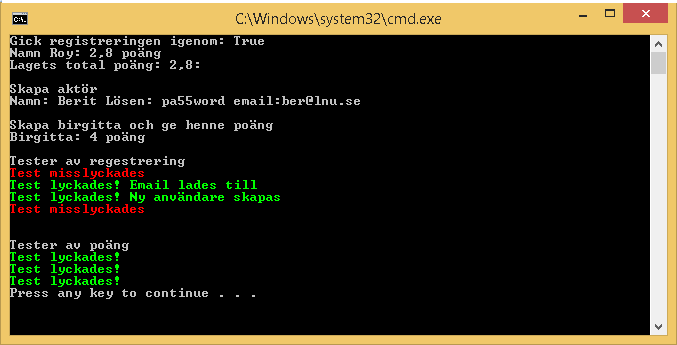
Testet blev godkänt.

Resterande tester är lika med olika typer av data.

Samtliga tester hämtar metoden Message för att få rätt färgbeteckning.



En visuel bild av alla tester som ligger i samma ordning som testsviterna.



**Test Data**

### Test av skapa nya instanser från båda användningsfallen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Typ av test** | **Test data** | **Metod** | **kommentar** | **Resultat** |
| Skapa gymnast och ge henne namn och poäng | Namn = ”Birgitta”  Poäng = 4.0 | Skapa en instans av klassen Secretary och skapa en ny användare och tilldela användaren poäng | Förväntat resultat. | Godkännt |
| Skapa en ny aktör och tilldela samtliga fält | Namn =”Berit”  Lösenord=”pa55sword”  Email=”ber@lnu.se” | Skapa en instans av klassen Register och skapa en ny aktör och tilldela aktören värden | Förväntat resultat. | Godkännt |

### Testsvit 1- Register

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Typ av test** | **Test data** | **Metod** | **kommentar** | **Resultat** |
| Kontroll av inloggning | Namn = ”Åke”  Lösenord = ”pass12”  Email =”roy.lnu.se” | Skapa en instans av klassen Register och skapa en ny konstruktor som innehåller tre parametrar. | Förväntat resultat.  Kräver valid email | Misslyckat |
| Kontroll av email | Email =”roy@lnu.se” | Skapa en instans av klassen Register och skapa en ny konstruktor som innehåller parametern email. | Förväntat resultat  Kollar om email är valid | Godkänt |
| Kontroll av inloggning | Namn = ”Roy”  Lösenord = ”Brun1”  Email =”roy@lnu.se” | Skapa en instans av klassen Register och skapa en ny konstruktor som innehåller tre parametrar. | Förväntat resultat | Godkänt |
| Kontroll av inloggning | Namn = ””  Lösenord = ””  Email =”” | Skapa en instans av klassen Register och skapa en ny konstruktor som innehåller tomma parametrar. | Förväntat resultat  Kräver ifylda parametrar | Misslyckat |

### Testsvit 2- Secretary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Typ av test** | **Test data** | **Metod** | **kommentar** | **Resultat** |
| Lägga till poäng | Namn = ”Roy”  Poäng = 5 | Skapa en instans av klassen Secretary och skapa en ny konstruktor som innehåller två parametrar. | Förväntat reultat | Godkänt |
| Lägga till poäng | Namn = ”Roy Lnu”  Poäng = 10.4 | Skapa en instans av klassen Secretary och skapa en ny konstruktor som innehåller två parametrar innehållande decimal tal och efternamn | Förväntat reultat | Godkänt |
| Testar metoderna | Namn = ””  Poäng = 0 | Skapa en instans av klassen Secretary och skapa en ny konstruktor som innehåller två parameter var den ena har tomt namn | Åtgärd av tomma namn fält | Godkänt |
| Lägga till poäng | Namn = ”3”  Poäng = ”lnu” | Skapa en instans av klassen Secretary och skapa en ny konstruktor som innehåller två parametrar i fel ordning. | Förväntat reultat  Kan inte använda string i Poäng eller siffror i Namn | Misslyckad |