

## Testdata: Klassen Control

### Test på logincontrol()

#### Testfall 1(16-12-2014)

Här testar jag logincontrol(), funktionen kommer i systemet att jämföra inmatade användarnamn och lösenord med registrerade användare. Om man finns med i registret så ska funktionen meddela att man är inloggade

Följande användare finns registrerade i systemet  
Användarnamn: Marco  
Lösenord: 123

#### Utförd

Test 1, funktion logincontrol(), scenario 1, resultat: du är inlogad

#### Testkod(16-12-2014)

```
public void TestMethod1()
{
    string name ="Marco";

    string password = "123";

    control b1 = new control(name, password);

    b1.logincontrol(); // jämföra inmatade användarnamn och lösenord med
registrerade användare
}
```

### Testfall 2(16-12-2014)

Här testar jag logincontrol(), funktionen kommer i systemet att jämföra inmatade användarnamn och lösenord med registrerade användare. Om man finns med i registret så ska funktionen meddela att man är inloggad

Följande användare finns inte registrerade i systemet  
Användarnamn: Zom  
Lösenord: 133

#### Utförd

Test 2, funktion logincontrol(), alternativ scenario 1, resultat: inte inloggad

### Testkod(17-16-2014)

```
public void TestMethod2()
{
    string name = "Zom";

    string password = "133";

    control b1 = new control(name, password);

    b1.logincontrol();//jämföra inmatade användarnamn och lösenord med
    registrerade användare
}
```

### Resultat

Båda testfallen funkar, funktionen logincontrol() fungerar.

## **Test på add()**

### **Testfall 1(17-16-2014)**

Här testar jag add(), funktionen kommer i systemet att lägga till nya användare i medlemsregister.

Användarnamn: Ole  
Lösenord: 6790

#### **Utförd**

Test 3, funktion add(), scenario 1, resultat: den nya användaren finns i medlemsregister

### **Testkod(17-16-2014)**

```
public void TestMethod3()
{
    string name = "Ole";

    string password = "6790";

    control b1 = new control();

    b1.add(name, password);//läger till nya användare i medlemsregister.

}
```

### **Resultat**

Funktionen add() fungerar.

### **Test på Load()**

#### **Testfall 1(16-12-2014)**

Här testar jag Load(), funktionen kommer i systemet att läsa in medlemsregister från fil. den här funktionen används i systemet när man ska loggin, lägger till ny användare.

##### **Utförd**

Test 4, funktion Load(), scenario 1, resultat: medlemsregister skrivs ut

#### **Testkod(16-12-2014)**

```
public void TestMethod4()
{
    control b1 = new control();

    b1.Load();// läser in data från text fil
}
```

##### **Resultat**

Funktionen Load() fungerar.

### **Test på Save()**

#### **Testfall 1(16-12-2014)**

Här testar jag Save(), funktionen kommer i systemet att spara nya data på medlemsregister fil. Den här funktionen används i systemet när man ska lägga till ny användare.

Åter skapar funktionen add(); i test klassen för att isolerar bara Save() funktionen

Följande användare finns inte registrerade i systemet, men kommer läggas till efter testet körts.

Användarnamn: R2D2  
Lösenord: 1905

##### **Utförd**

Test 5, funktion Save(), scenario 1, resultat: medlemsregister skrivs ut, ny data på användare finns med.

### Testkod(16-12-2014)

```
public void TestMethod5()
{
    string n = "R2D2";

    string p = "1905";

    control b1 = new control();

    b1.namen.Add(n);

    b1.password.Add(p);

    b1.Load();

    b1.Save();// sparar data till text fil
}
```

### Resultat

Funktionen Save() fungerar.

### Testdata: Klassen Regist

#### Test på Namnkontroll()

### Testfall 1(16-12-2014)

Här testar jag namnkontroll(), funktionen kommer i systemet att kontrollera om det inmatade namnet när man registrerar har mellanslag.

den här funktionen används för att kontrollera att man inte har mellanslag/blanktecken i sitt användarnamn.

Användarnamn: Pet er

#### Utförd

Test 6, funktion namnkontroll(), scenario 1, resultat: Hittar mellanslag och får tillbaks värdet: "true"

### Testkod(16-12-2014)

```
public void TestMethod6()
{
    string name = "Pe ter";
    bool loop;

    regist b1 = new regist();

    loop = b1.namnkontrol(name);// kontrollerar mellanslag/ blanktecken

    Assert.IsTrue(loop, "Tester misslyckades");

    Console.WriteLine(loop);
}
```

### Testfall 2(16-12-2014)

Här testar jag namnkontrol() i klassen, funktionen kommer i systemet att kontrollera om det inmatade namnet när man registrerar har mellanslag.

den här funktionen används för att kontrollera att man inte har mellanslag/blanktecken i sitt användarnamn.

Användarnamn: Peter

#### Utförd

Test 7, funktion namnkontrol(), alternativ scenario 1, resultat: Hittar inte mellanslag och får tillbaks värdet: "false"

### Testkod(16-12-2014)

```
public void TestMethod7()
{
    string name = "Peter";
    bool loop;

    regist b1 = new regist();

    loop = b1.namnkontrol(name); //kontrollerar mellanslag/ blanktecken

    Assert.IsTrue(loop, "Tester misslyckades");

    Console.WriteLine(loop);
}
```

## Resultat

Båda testfallen funkar, funktionen namnkontrol() fungerar.

### Test på passwordkontrol()

#### Testfall 1(16-12-2014)

Här testar jag passwordkontrol(), funktionen kommer i systemet att kontrollera om det inmatade lösenordet när man registrerar har mellanslag.

den här funktionen används för att kontrollera att man inte har mellanslag/blanktecken i sitt användarlösenord.

Lösenord: 12 67

#### Utförd

Test 8, funktion passwordkontrol(), scenario 1, resultat: Hittar mellanslag och får tillbaks värdet: "true"

#### Testkod(16-12-2014)

```
public void TestMethod8()
{
    string password = "12 67";

    bool loop;

    regist b1 = new regist();

    loop = b1.passwordkontrol(password);// kontrollerar mellanslag/ blanktecken

    Console.WriteLine(loop);
}
```

### **Testfall 2(16-12-2014)**

Här testar jag passwordkontrol(), funktionen kommer i systemet att kontrollera om det inmatade lösenordet när man registrerar har mellanslag.

den här funktionen används för att kontrollera att man inte har mellanslag/blanktecken i sitt användarlösenord.

Lösenord: 1267

#### **Utförd**

Test 9, funktion passwordkontrol(), alternativ scenario 1, resultat: Hittar inte mellanslag och får tillbaks värdet: "false"

### **Testkod(16-12-2014)**

```
public void TestMethod6()
{
    string password = "1267";

    bool loop;

    regist b1 = new regist();

    loop = b1.passwordkontrol(password);// kontrollerar mellanslag/ blanktecken

    Console.WriteLine(loop);
}
```

#### **Resultat**

Båda testfallen funkar, funktionen passwordkontrol() fungerar.



### Test på bekräfta()

#### **Testfall 1(16-12-2014)**

Här testar jag bekräfta(), funktionen kommer i systemet att ge än möjlighet att avbryta registrering.

den här funktionen används för att kontrollera att man vill registrera sig

val: 1 = JA

#### **Utförd**

Test 10, funktion bekräfta(), scenario 1, resultat: Val 1 = JA "Du är nu registrerad"

#### **Testkod(16-12-2014)**

```
public void TestMethod8()
{
    string n = "test";

    string p = "test";

    int val = 1;

    regist b1 = new regist();

    b1.bekräfta(n, p, val); // visar möjlighet att avbryta registrering.
}
```

#### **Testfall 2(16-12-2014)**

Här testar jag bekräfta(), funktionen kommer i systemet att ge än möjlighet att avbryta registrering.

den här funktionen används för att kontrollera att man vill registrera sig

val: 2 = NEJ

#### **Utförd**

Test 11, funktion bekräfta(), alternativ scenario 1, resultat: Val 2 = Nej "registrering stoppad"

### Testkod(17-12-2014)

```
public void TestMethod6()
{
    string n = "test";

    string p = "test";

    int val = 2;

    regist b1 = new regist();

    b1.bekrefta(n, p, val);// visar möjlighet att avbryta registrering.
}
```

### Resultat

Båda testfallen funkar, funktionen bekräfta() fungerar.