

## Uppgift 5 – Implementera testsviten och kör

### Enhetstest 1

Testklass som kördes:

```
class TestLogin
{
    // ENHETSTEST 1 - Testar metoder i klassen Login.

    // Testfall 1 - Testar metoden SuccessfulLogin.
    public static void TestSuccessfulLogin()
    {
        //Testfixtur är klassen Login
        Boolean login = Login.SuccessfulLogin();

        if (login == true)
        {
            Console.WriteLine("Testet lyckades");
        }
        else
        {
            Console.WriteLine("Testet misslyckades");
        }
    }

    // Testfall 2 - Testar metoden UnsuccessfulLogin.
    public static void TestUnsuccessfulLogin()
    {
        //Testfixtur är klassen Login
        Boolean login = Login.UnsuccessfulLogin();

        if (login == false)
        {
            Console.WriteLine("Testet lyckades");
        }
        else
        {
            Console.WriteLine("Testet misslyckades");
        }
    }
}
```

### Testresultat

**Testfall 1:** "Testet lyckades" (true).

**Testfall 2:** "Testet lyckades" (false).

## Enhetstest 2

### Testklass som kördes:

```
class TestUserDetails
{
    // ENHETSTEST 2 - Testar metod i klassen UserDetails.

    // Testfall 1 - Testar validering av korrekt användarnamn och lösenord.
    public static void TestCorrectUserDetails()
    {
        //Testfixtur är klassen UserDetails
        string username = "username1";
        string password = "password1";
        UserDetails userDetails = new UserDetails(username, password);
    }

    // Testfall 2 - Testar validering av inkorrekt användarnamn och lösenord.
    public static void TestIncorrectUserDetails()
    {
        //Testfixtur är klassen UserDetails
        string username = "username2";
        string password = "password2";
        UserDetails userDetails = new UserDetails(username, password);
    }

    // Testfall 3 - Testar validering av korrekt användarnamn och inkorrekt lösenord.
    public static void TestCorrectUserIncorrectPass()
    {
        //Testfixtur är klassen UserDetails
        string username = "username1";
        string password = "password2";
        UserDetails userDetails = new UserDetails(username, password);
    }

    // Testfall 4 - Testar validering av inkorrekt användarnamn och korrekt lösenord.
    public static void TestIncorrectUserCorrectPass()
    {
        //Testfixtur är klassen UserDetails
        string username = "username2";
        string password = "password1";
        UserDetails userDetails = new UserDetails(username, password);
    }
}
```

### Testresultat

**Testfall 1:** "Rätt användarnamn", "Rätt lösenord".

**Testfall 2:** "Fel användarnamn", "Fel lösenord".

**Testfall 3:** "Rätt användarnamn", "Fel lösenord".

**Testfall 4:** "Fel användarnamn", "Rätt lösenord".

## **Analys**

Samtliga tester gick som förväntat, vilket innebär att både klasser och dess metoder fungerar som de är tänkta att göra. Det finns självklart enorma förbättringsmöjligheter både för klasserna och metoderna, och förmodligen för hela det här testupplägget, men det ligger långt utanför mina nuvarande kunskaper i ämnet. Jag försökte hålla det på en nivå som jag behärskar.