

Uppgift 4 – Enhetstestning

Testsvit

Syftet:

Att säkerställa att valda klasser fungerar enligt funktionella och kvalitativa krav som är beskrivna i dokumentationen.

Enhetstest på klasser *Arbiter* och *Season*

Testfixtures:

- Instans av klassen *Arbiter* med namnet John Locke
- Instans av klassen *Season* med namnet Championship

Arbiter()

Testfallet initialiserar en instans av klassen *Arbiter* med namnet *newArbiter*.

Scenario 1:

1. Skapas en instans av klassen *Arbiter*
2. Tilldelar ett värde till *FirstName*
3. Returnerar ett sträng

Scenario 2:

4. Skapas en instans av klassen *Arbiter*
5. Tilldelar ett värde till *LastName*
6. Returnerar ett sträng

Scenario 3:

7. Skapas en instans av klassen *Arbiter*
8. Tilldelar ett värde till *Season*
9. Returnerar ett sträng

Scenario 4:

1. Tilldelar tom sträng till FirstName
2. Ett lämpligt felmeddelande returneras

Scenario 5:

1. Tilldelar tom sträng till LastName
2. Ett lämpligt felmeddelande returneras

Scenario 6:

1. Tilldelar tom sträng till Season
2. Ett lämpligt felmeddelande returneras

Primärt flöde:

FirstName, LastName och Season returneras som en sträng

Alternativt flöde:

FirstName, LastName eller Season har ogiltiga värden och lämpligt meddelande returneras.

Season()

Testfallet initialiserar en instans av klassen *Season* med namnet *Championship*.

Scenario 1:

1. Skapas en instans av klassen Season
2. Tilldelar ett värde till Name och Year egenskaperna.
3. Returneras ett värde för Name som sträng, och ett värde för Year som sträng

Scenario 2:

1. Tilldelar ett tomt sträng till Name
2. Ett fel meddelande returneras

Scenarion 3:

1. Tilldelar ett tomt sträng till Year
2. Ett fel meddelande returneras

Primärt flöde:

Name och Year returneras som en sträng

Alternativ flöde:

Name eller Year har ogiltiga värden och lämpligt meddelande returneras.

Integrationtest Användningsfall - Registrera ny domare

Testfixtures:

- Ett objekt av klassen Arbiter skapas
- Ett objekt av klassen Season skapas

Syftet:

Testar funktionalitet beskriven i Användningsfall 4.

Primärt flöde:

1. Administratör lägger till ny domare med namnet John Locke
2. Administratör lägger ny säsong med namnet Championship till ett objekt av klassen Arbiter

Alternativt flöde:

1. Objekt av klassen Season med namnet Championship inte finns
2. Ett undantag kastas

Scenario 1:

1. Ny domare med namnet John Locke skapas

2. Tilldelas en säsong för domare
3. Säsongen finns redan
4. Presenterar data med en domare, och gällande säsong

Scenario 2:

1. Ny domare med namnet John Locke skapas
2. Tilldelas en säsong för domare
3. Säsongen finns inte
4. Ett fel meddelande visas

Motivation

Tester genomförs för att säkerställa att det går att initialisera instanser av klasser Arbiter och Season och att klasserna setter och getter metoder returnerar korrekta värden. Alternativt, säkerställs det att ett lämpligt meddelande returneras i fall att korrekta värden inte returneras. För testens skull är alla ogiltiga testvärden är en tom sträng. Jag har valt att testa på detta sätt för att förenkla uppgiften, eftersom min erfarenhet i tester är otillräckligt. Jag har förklarat i laboration 2 att pre villkor för de flesta användningsfall är att inmatning av grunddata är korrekt, och därför väljer jag att göra tester just på klasserna som används för inmatning av data (inte beräkningar eller presentation av data).