

## JUnit tesztek eredménye

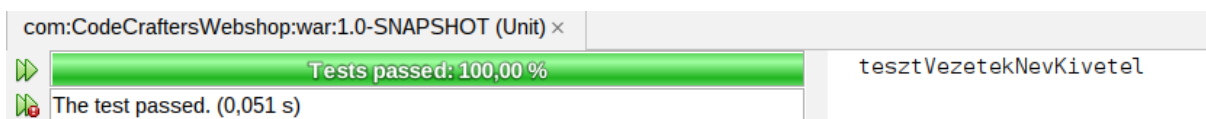
A java alapú backend a JUnit 4 egységteszt-keretrendszer használatával lett tesztelve. A Felhasználó és a Játék entitás validáló metódusainak működése, a megfelelő kivétel osztályok alkalmazása és a kivétel pontos szövege.

Ebben a metódusban azt teszteljük, hogy a Felhasználó entitás a FelhasználóException kivétel-t dobja-e.

```
@Test(expected = FelhasznaloException.class)
public void tesztVezetekNevKivetel() throws FelhasznaloException {
    System.out.println(x: "tesztVezetekNevKivetel");

    String vezetekNev = "";
    boolean eredmeny = Felhasznalo.vezetekNevEllenorzes(vezetekNev);

    assertTrue(condition: eredmeny);
}
```









Ezekben a metódusokban a Felhasználó vezetéknevéhez köthető kivételeket teszteljük.

```
@Rule  
public ExpectedException thrown = ExpectedException.none();  
  
@Test  
public void tesztVezetekNevUres() throws FelhasznaloException {  
    System.out.println(x: "tesztVezetekNevUres");  
  
    thrown.expect(type:FelhasznaloException.class);  
    thrown.expectMessage(substring: "A felhasználó vezetékeve nem lehet üres!");  
    String vezetekNev = "";  
    boolean eredmeny = Felhasznalo.vezetekNevEllenorzes(vezetekNev);  
  
    assertTrue(condition:eredmeny);  
}  
  
@Test  
public void tesztVezetekNevHosszu() throws FelhasznaloException {  
    System.out.println(x: "tesztVezetekNevHosszu");  
  
    thrown.expect(type:FelhasznaloException.class);  
    thrown.expectMessage(substring: "A felhasználó vezetékeve nem lehet 100 karakternél hosszabb!");  
  
    String vezetekNev = "ELeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee";  
    boolean eredmeny = Felhasznalo.vezetekNevEllenorzes(vezetekNev);  
  
    assertTrue(condition:eredmeny);  
}  
  
@Test  
public void tesztKeresztNevSpecialis() throws FelhasznaloException {  
    System.out.println(x: "tesztKeresztNevSpecialis");  
  
    thrown.expect(type:FelhasznaloException.class);  
    thrown.expectMessage(substring: "A felhasználó keresztnéve nem tartalmazhat speciális karaktert!");  
  
    String vezetekNev = "Elek?";  
    boolean eredmeny = Felhasznalo.keresztNevEllenorzes(keresztNev: vezetekNev);  
  
    assertTrue(condition:eredmeny);  
}
```



Ezekben a metódusokban a Felhasználó emai címéhez köthető kivételeket teszteljük.

```
@Test  
public void tesztEmailUres() throws FelhasznaloException {  
    System.out.println(x: "tesztEmailUres");  
  
    thrown.expect(type:FelhasznaloException.class);  
    thrown.expectMessage(substring:"A felhasználó emailje nem lehet üres!");  
  
    String email = "";  
    boolean eredmény = Felhasznalo.emailEllenorzes(email);  
  
    assertTrue(condition:eredmény);  
}  
  
@Test  
public void tesztEmailHosszu() throws FelhasznaloException {  
    System.out.println(x: "tesztEmailHosszu");  
  
    thrown.expect(type:FelhasznaloException.class);  
    thrown.expectMessage(substring:"A felhasználó emailje nem lehet 100 karakternél hosszabb!");  
  
    String email = "kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk@gmail.com";  
    boolean eredmény = Felhasznalo.emailEllenorzes(email);  
  
    assertTrue(condition:eredmény);  
}  
  
@Test  
public void tesztEmailValos() throws FelhasznaloException {  
    System.out.println(x: "tesztEmailValos");  
  
    thrown.expect(type:FelhasznaloException.class);  
    thrown.expectMessage(substring:"A felhasználó emailjének tartalmaznia kell a @ karaktert!");  
  
    String email = "kkgmail.com";  
    boolean eredmény = Felhasznalo.emailEllenorzes(email);  
  
    assertTrue(condition:eredmény);  
}
```

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×		
	Tests passed: 100,00 %	tesztEmailUres
	The test passed. (0,059 s)	
com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×		
	Tests passed: 100,00 %	tesztEmailHosszu
	The test passed. (0,053 s)	
com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×		
	Tests passed: 100,00 %	tesztEmailValos
	The test passed. (0,074 s)	

Ezekben a metódusokban a Felhasználó jelszávához köthető kivételeket teszteljük.

```
@Test
public void tesztJelszoUres() throws FelhasznaloException {
    System.out.println(x: "tesztJelszoUres");

    thrown.expect(type:FelhasznaloException.class);
    thrown.expectMessage(substring:"A felhasználó jelszava nem lehet üres!");

    String jelszo = "";
    boolean eredmeny = Felhasznalo.jelszoEllenorzes(jelszo);

    assertTrue(condition:eredmeny);
}

@Test
public void tesztJelszoHosszu() throws FelhasznaloException {
    System.out.println(x: "tesztJelszoHosszu");

    thrown.expect(type:FelhasznaloException.class);
    thrown.expectMessage(substring:"A felhasználó jelszava nem lehet 100 karakternél hosszabb!");

    String jelszo = "kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk123456789**";
    boolean eredmeny = Felhasznalo.jelszoEllenorzes(jelszo);

    assertTrue(condition:eredmeny);
}

@Test
public void tesztJelszoRovid() throws FelhasznaloException {
    System.out.println(x: "tesztJelszoRovid");

    thrown.expect(type:FelhasznaloException.class);
    thrown.expectMessage(substring:"A felhasználó jelszava nem lehet 8 karakternél rövidebb!");

    String jelszo = "asd123";
    boolean eredmeny = Felhasznalo.jelszoEllenorzes(jelszo);

    assertTrue(condition:eredmeny);
}

@Test
public void tesztJelszoSzam() throws FelhasznaloException {
    System.out.println(x: "tesztJelszoSzam");

    thrown.expect(type:FelhasznaloException.class);
    thrown.expectMessage(substring:"A felhasználó jelszávanak tartalmaznia kell számot!");

    String jelszo = "asdASDqwe*";
    boolean eredmeny = Felhasznalo.jelszoEllenorzes(jelszo);

    assertTrue(condition:eredmeny);
}

@Test
public void tesztJelszoBetu() throws FelhasznaloException {
    System.out.println(x: "tesztJelszoBetu");

    thrown.expect(type:FelhasznaloException.class);
    thrown.expectMessage(substring:"A felhasználó jelszávanak tartalmaznia kell betűt!");

    String jelszo = "123789456*";
    boolean eredmeny = Felhasznalo.jelszoEllenorzes(jelszo);



    assertTrue(condition:eredmeny);
}

@Test
public void tesztJelszoSpecialis() throws FelhasznaloException {
    System.out.println(x: "tesztJelszoSpecialis");







    thrown.expect(type:FelhasznaloException.class);
    thrown.expectMessage(substring:"A felhasználó jelszávanak tartalmaznia kell speciális karaktert!");

    String jelszo = "asdASD123q";
    boolean eredmeny = Felhasznalo.jelszoEllenorzes(jelszo);

    assertTrue(condition:eredmeny);
}
```

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
 Tests passed: 100,00 %	tesztJelszoUres
 The test passed. (0,062 s)	



com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
 	tesztNevUres
 The test passed. (0,073 s)	
com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
 	tesztNevHosszu
 The test passed. (0,057 s)	




Ebben a metódusban a Játék árához köthető kivételt teszteljük.

```
@Test
public void tesztArNegativ() throws JatekException {
    System.out.println(x: "tesztArNegativ");

    thrown.expect(type: JatekException.class);
    thrown.expectMessage(substring: "A játék ára nem lehet kisebb, mint 0!");

    int ar = -1;
    boolean eredmény = Jatek.arEllenorzes(ar);

    assertTrue(condition: eredmény);
}
```

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
 	tesztArNegativ
 The test passed. (0,053 s)	




Ebben a metódusban a Játék leírásához köthető kivételt teszteljük.

```
@Test
public void tesztLeirasUres() throws JatekException {
    System.out.println(x: "tesztLeirasUres");

    thrown.expect(type: JatekException.class);
    thrown.expectMessage(substring: "A játék leírása nem lehet üres!");

    String leiras = "";
    boolean eredmény = Jatek.leirasEllenorzes(leiras);

    assertTrue(condition: eredmény);
}
```

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
 	tesztLeirasUres
 The test passed. (0,039 s)	



Ebben a metódusban a Játék képéhez köthető kivételt teszteljük.

```
@Test
public void tesztKepUres() throws JatekException {
    System.out.println(x: "tesztKepUres");

    thrown.expect(type: JatekException.class);
    thrown.expectMessage(substring: "A játék képe nem lehet üres!");

    String kep = "";
    boolean eredmeny = Jatek.kepEllenorzes(kep);

    assertTrue(condition: eredmeny);
}
```

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
 Tests passed: 100,00 %	tesztKepUres
 The test passed. (0,067 s)	



Ebben a metódusban a Játék korhatárához köthető kivételt teszteljük.

```
@Test
public void tesztKorhatar() throws JatekException {
    System.out.println(x: "tesztKorhatar");

    thrown.expect(type: JatekException.class);
    thrown.expectMessage(substring: "A játék korhatára nem térhet el a PEGI számoktól!");

    int korhatar = 11;
    boolean eredmeny = Jatek.korhatarEllenorzes(korhatar);

    assertTrue(condition: eredmeny);
}
```

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
 Tests passed: 100,00 %	tesztKorhatar
 The test passed. (0,089 s)	



Ezekben a metódusokban a Játék akciójához köthető kivételeket teszteljük.

```
@Test
public void tesztAkcioNegativ() throws JatekException {
    System.out.println(x: "tesztAkcioNegativ");

    thrown.expect(type: JatekException.class);
    thrown.expectMessage(substring: "A játék akciója nem lehet kisebb, mint 0!");

    int akcio = -1;
    boolean eredmeny = Jatek.akcioEllenorzes(akcio);

    assertTrue(condition: eredmeny);
}

@Test
public void tesztAkcio() throws JatekException {
    System.out.println(x: "tesztAkcio");

    thrown.expect(type: JatekException.class);
    thrown.expectMessage(substring: "A játék akciója nem lehet nagyobb, mint 100!");

    int akcio = 101;
    boolean eredmeny = Jatek.akcioEllenorzes(akcio);

    assertTrue(condition: eredmeny);
}
```

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
Tests passed: 100,00 %	tesztAkcioNegativ
The test passed. (0,07 s)	

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
Tests passed: 100,00 %	tesztAkcio
The test passed. (0,077 s)	

Ebben a metódusban a Játék mennyiségraktáron értékéhez köthető kivételt teszteljük.

```
@Test
public void tesztMennyisegraktaronNegativ() throws JatekException {
    System.out.println(x: "tesztMennyisegraktaronNegativ");

    thrown.expect(type: JatekException.class);
    thrown.expectMessage(substring: "A játék mennyiségraktaronja nem lehet kisebb, mint 0!");

    int mennyisegraktaron = -1;
    boolean eredmeny = Jatek.mennyisegraktaronEllenorzes(mennyisegraktaron);

    assertTrue(condition: eredmeny);
}
```

com:CodeCraftersWebshop:war:1.0-SNAPSHOT (Unit) ×	
Tests passed: 100,00 %	tesztMennyisegraktaronNegativ
The test passed. (0,053 s)	



A futtató osztály eredménye bizonyítja a tesztek sikerességét:

tesztEmailUres  
tesztJelszoBetu  
tesztJelszoSzam  
tesztJelszoUres  
tesztEmailValos  
tesztJelszoRovid  
tesztKeresztNevUres  
tesztVezetekNevUres  
tesztEmailHosszu  
tesztVezetekNevHosszu  
tesztKeresztNevHosszu  
tesztJelszoSpecialis  
tesztJelszoHosszu  
tesztVezetekNevSpecialis  
tesztKeresztNevSpecialis  
tesztVezetekNevKivétel

Tesztelt osztály: com.codecrafterswebshop.Model.FelhasznaloTest  
Futtatott tesztek száma: 16  
Sikeres tesztek száma: 16  
Hibás tesztek száma: 0

tesztNevUres  
tesztNevHosszu  
tesztKorhatar  
tesztAkcio  
tesztKepUres  
tesztAkcioNegativ  
tesztLeirasUres  
tesztMennyisegraktaronNegativ  
tesztNevKivétel  
tesztArNegativ

Tesztelt osztály: com.codecrafterswebshop.Model.JatekTest  
Futtatott tesztek száma: 10  
Sikeres tesztek száma: 10  
Hibás tesztek száma: 0