

# TESZTAUTOMATIZÁLÁS

4. forduló



A kategória támogatója: EPAM

RENDELKEZÉSRE ÁLLÓ IDŐ:

30:00

## Ismertető a feladathoz

### Fontos információk

A forduló után a megoldások publikálásával együtt iránymutatásként elérhetőek lesznek a **helyezéssel kapcsolatos információk**, látni fogod, hogy a kategóriában a játékosok 20%, 40% vagy 60%-a közé tartozol-e épp.

Felhívjuk figyelmedet, hogy a következő, **5. fordulótól az egyes kategóriák csak a kijelölt napokon lesznek megoldhatóak 7-22 óra között**, érdemes letöltened a naptárat a [Kategóriáim](#) menüpontban.

Felhasznált idő: 01:32/30:00

Elért pontszám: 0/23

## 1. feladat 0/8 pont

Adott a következő DOM struktúra. Mely XPATH selectorokkal tudjuk kiválasztani csak a második "span"-eket a notes li elemekben?

```
<body>
  <li element = "notes">
    <span class = "label">Yaris</span>
    <span class = "label">Toyota</span>
  </li>
  <li element = "notes">
    <span class = "label">Astra</span>
    <span class = "label">Opel</span>
  </li>
  <li element = "notes">
    <span class = "label">Fabia</span>
```

```
        <span class = "label">Skoda</span>
    </li>
    <li element = "notes">
        <span class = "label">Samara</span>
        <span class = "label">Lada</span>
    </li>
    <li element = "notes">
        <span class = "label">Lanos</span>
        <span class = "label">Daewoo</span>
    </li>
    <li element = "notes">
        <span class = "label">Z4</span>
        <span class = "label">BMW</span>
    </li>
</body>
```

## Válaszok



```
//li[@element="notes"]/span[2]
```



```
//li[@element="notes"]/span
```



```
//li[@element="notes"]/span[last()]
```



```
//*[last()][name()='span']
```

## Magyarázat

Az első, harmadik és negyedik válasz helyes.

Az első az olyan span elemet fogja meg, amely egy li class-nak a gyereke és második span az adott szinten.

A második az első span gyerek elemet fogja meg, ezért ez helytelen.

A harmadik az olyan span elemet fogja meg, amely utolsó a szintjén és li elemnek a gyereke.

A negyedik válasz hasonló a harmadikhoz, csak nem nézi, hogy li elemben lévő span-ról beszélünk, de azok közül az utolsó fogja meg.

Melyik eszköz(ök) alkalmas(ak) önmagában API tesztelésre?

### Válaszok

- ☐ Selenium IDE
- ☒ Postman
- ☐ Cucumber
- ☒ Katalon Studio

### Magyarázat

Selenium IDE - UI tesztelői eszköz.

Cucumber - egy BDD tesztelői framework amely Gherkin nyelvet használ, de önmagában nem használható API tesztelésre, kombinálni kell más eszközökkel.

A Postman és Katalon studio pedig alkalmas API tesztelésre.

## 3. feladat 0/2 pont

Milyen státusz kód jelzi a kliensnek, ha olyan tartalmat szeretne elérni, amihez nincs jogosultsága?

### Válasz

- ☐ 401 Unauthorized
- ☒ 403 Forbidden
- ☐ 405 Method Not Allowed
- ☐ 406 Not Acceptable

### Magyarázat

A felsoroltak közül a 403 Forbidden státusz kód jelzi a kliensnek, ha jogosulatlan tartalmat szeretne elérni.

A 401-es kód az azonosító információk hiányát vagy invaliditását jelzi.

A 405 az adott operáció blokkolását jelzi.

A 406 pedig a beküldött adat "nem elfogadását".

## 4. feladat 0/4 pont

Mely reguláris kifejezések segítségével tudom kiválasztani **csak** a képfájlokat a következő listából?

tesztes.doc

tesztes.jpg

auto.xls

profil.txt

fontos.png

fejleszto.js

kep\_01.jpg

kep\_02.bmp

harcos.bmpz

### Válaszok

☐

`\S+.(jpg|bmp|png)`

☒

`\S+\. (jpg|bmp|png)$`

☐

`\s+\. (jpg|bmp|png)$`

☒

`\w+\. (jpg|bmp|png)$`

### Magyarázat

Az első reguláris kifejezésre azok a stringek illeszkednek, amelyek valamilyen nem szóközt vagy szóközöket tartalmaznak, majd ezután egy . karaktert és jpg, bmp vagy png kifejezés követi. Ez azért nem jó, mert erre illeszkedik a harcos.bmpz fájl is, ami nem kép fájl.

A második reguláris kifejezés ugyanazt tudja, mint az első, de van egy szigorítás a végén, hogy a jpg, bmp vagy png kifejezésnek kell zárnia a string-et. Emiatt ez már jó lesz a kép fájljainkhoz.

A harmadik reguláris kifejezés már rögtön rosszul indul, mivel szóköz karaktert vagy karaktereket kell tartalmaznia a szónak, hogy illeszkedjen. Emiatt nem fog illeszkedni egyik fájl név sem rá.

Az utolsó reguláris kifejezésre is illeszkedni fognak a kép fájlok, mivel betű, számjegy, vagy alulvonás karaktert vár és utána a megfelelő formátumokat zárásként.

## 5. feladat 0/2 pont

Az alábbi jellemezők közül melyik **nem** igaz a komponens tesztelésre?

### Válasz

- ☐ A tesztek a fejlesztéssel párhuzamosan készülnek, többnyire a fejlesztők által.
- ☐ Lehetőséget biztosítanak a hibák alacsonyabb szinten való megtalálására.
- ☒ Tesztelik, hogy a szoftver, mint egész a felhasználók elvárásainak megfelelően működik.
- ☐ Tesztelik, hogy a szoftver egymástól független részei jól működnek.

## Magyarázat

A harmadik válasszal ellentétben a komponens tesztek a szoftver egymástól független részeit tesztelik, ezért a harmadik válasz nem igaz a komponens tesztelésre vonatkozóan.

## 6. feladat 0/3 pont

Az alábbi válaszlehetőségek közül melyik indokolja leginkább egy teszteset felvételét az automata regressziós tesztkészletbe?

### Válasz

- ☒ Hogy minden szoftver frissítés esetén meggyőződjünk arról, hogy a tesztelt feature még mindig megfelelően működik.
- ☐ Hogy minél több hibát megtaláljunk a tesztelt feature-ben.
- ☐ Hogy minden új verzió kiadásakor biztosak lehessünk benne, hogy az alkalmazás hibamentes.
- ☐ Hogy az automata tesztheink minél hosszabb ideig fussanak, emiatt párhuzamosíthassuk a tesztek futását.

## Magyarázat

A helyes válasz:

Hogy minden szoftver frissítés esetén meggyőződjünk arról, hogy a tesztelt feature még mindig megfelelően működik.

A regressziós tesztelés esetén nem a feature-ben lévő új hibák megtalálása a cél (hiszen azokat jó esetben a feature kezdeti tesztelések vagy az automata teszt implementálásakor már megtaláltuk), hanem az, hogy minden szoftverfrissítésnél megbizonyosodjunk arról, hogy a tesztelt feature még mindig a követelményeknek megfelelően működik és más feature-ökben eszközölt módosítások nem okoztak nem várt hibát.

A "Hogy minden új verzió kiadásakor biztosak lehessünk benne, hogy a tesztelt feature hibamentes." válasz nem helyes, hiszen a tesztelés egyik alapelve, hogy sosem mondhatjuk teljes bizonyossággal, hogy egy alkalmazás hibamentes. A tesztelés ugyanis a hibák létét bizonyítja, nem azok hiányát.

A "Hogy az automata tesztheink minél hosszabb ideig fussanak, emiatt párhuzamosíthassuk a tesztek futását." válasz szintén nem helyes, hiszen az automata tesztek futási idejének növelése nem lehet cél sosem.

## 7. feladat 0/2 pont

Adott az alábbi Java kódrészlet:

```
public class MyClass {  
    public void checkValues(boolean A, boolean B, boolean C) {  
        if ((A || B) && !C) {  
            System.out.println("OK");  
        } else {  
            // Comment  
        }  
        if (A) {  
            System.out.println("NOT OK");  
        }  
    }  
};  
}
```

Legkevesebb hány darab tesztesettel érhetünk el 100%-os útvonal lefedettséget (path coverage) ?

### Válasz

- ☐ 1
- ☐ 2
- ☐ 3
- ☒ 4

### Magyarázat

Path coverage során az egymástól független 'utak' a fontosak (azaz minden az IF feltételek által meghatározott végrehajtási útnak és kombinációjának végre kell hajtódnia), ezért, hogy 100%-os path coverage-t érjünk el, a következő tesztesetek szükségesek:

1. A = true, B = true, C = false
2. A = false, B = true, C = true
3. A = false, B = true, C = false
4. A = true, B = true, C = true

