

TESZTAUTOMATIZÁLÁS

6. forduló



A kategória támogatója: EPAM

RENDELKEZÉSRE ÁLLÓ IDŐ:

45:00

Ismertető a feladathoz

Felhasznált idő: 01:36/45:00

Elért pontszám: 0/46

1. feladat 0/12 pont

Mit fog kiírni az alábbi TypeScript kód (A TypeScript teljesen default konfigurációt használ)?

```
class Car {
  public engineName: string;
  public numberOfWheels: number;

  public constructor(engineName: string, numberOfWheels: number) {
    this.engineName = engineName;
    this.numberOfWheels = numberOfWheels;
  }
}

class Bicycle {
  public numberOfWheels: number;

  public constructor(numberOfWheels: number) {
    this.numberOfWheels = numberOfWheels;
  }
}

type Vehicle = Car | Bicycle;
```

```
const vehicles = [new Car('V8', 4), new Bicycle(2), new Bicycle(2)]
```

```
vehicles.forEach(vehicle => console.log(`The value: ${JSON.stringify(vehicle)}, the type ${typeof vehi
```

Válasz



The value: {"engineName":"V8","numberOfWheels":4}, the type object

The value: {"numberOfWheels":2}, the type object

The value: {"numberOfWheels":2}, the type object



The value: [object Object], the type object

The value: [object Object], the type object

The value: [object Object], the type object



The value: [object Object], the type Car

The value: [object Object], the type Bicycle

The value: [object Object], the type Bicycle



The value: {"engineName":"V8","numberOfWheels":4}, the type Car

The value: {"numberOfWheels":2}, the type Bicycle

The value: {"numberOfWheels":2}, the type Bicycle

Magyarázat

A TypeScript fordításkor eltörli az összes típus adatot, így mindennek a típusa itt 'object' marad. A `JSON.stringify` metódus pedig az object-ből rendes string-et csinál amit ki tudunk írni.

2. feladat 0/12 pont

Adott az alábbi Python kód:

```
import pytest

@pytest.mark.parametrize("x", [2,1])
@pytest.mark.parametrize(
    "y",
    [2, pytest.param(3, marks=pytest.mark.xfail)]
)
```

```
def test_foo(x,y):  
    assert x == y
```

Hogyan fog kinézni a tesztesetek összegzése a teszt futtatás végén pytest 6.1.1-es verziót használva (alapértelmezett beállításokkal)?

Válasz

- ☐ 1 passed, 1 xfailed
- ☒ 1 failed, 1 passed, 2 xfailed
- ☐ 1 passed, 1 failed
- ☐ 2 failed, 1 passed, 1 xfailed

Magyarázat

Pytest használatakor a paraméterek összes variációját a '@pytest.mark.parametrize' többszörösítésének segítségével határozhatjuk meg, így a jelenlegi példában 2*2, azaz 4 teszteset lesz lefuttatva a következő értékekkel:

1. x=2 y=2
2. x=1 y=2
3. x=2 y=3
4. x=1 y=3

A 'marks=pytest.mark.xfail'-al jelezhetjük, hogy egy adott tesztesetnél várhatóan el fog bukni az adott eset, így az eredményben 'xfailed' tesztesetként jelenik meg.

3. feladat 0/12 pont

Tekintsük az alábbi Java kódrészletben, Selenium page object-hez készült osztály vázát egy weboldal kezdőoldalához:

```
public class HomePageObject {  
    private static final String HOME_PAGE_URL = "https://www.myhomepage.com";  
  
    _____ //1. részlet jön ide  
    private WebElement registrationButton;  
  
    private WebDriver webDriver;  
  
    _____ //2. részlet jön ide  
}
```

Bővítsük úgy az osztályt, hogy:

1. A selenium megtalálja ID selector alapján a regisztrációs gombot a weboldalon
2. Az osztály példányosításakor inicializálódjon a WebElement értékkel való feltöltése
3. Legyen pontosan egy különálló metódus ami képes az adott weboldalra az URL-je alapján navigálni, de mást nem csinál

4. Legyen pontosan egy különálló metódus ami képes rákattintani a regisztrációs gombra, de mást nem csinál.

Mely kódrészleteket kell beírunk a 2 alávonással jelölt _____ helyére, hogy minden feltétel teljesüljön?

Tegyük fel, hogy a gomb selectorának értékét ismerjük.

Válasz

☐ 1. részlet:

```
@RunWith
```

2. részlet:

```
public HomePage(final WebDriver webDriver) {  
    PageFactory.initElements(webDriver, this);  
    this.webDriver = webDriver;  
}  
  
public void navigateToHomePage() {  
    webDriver.load(HOME_PAGE_URL);  
}  
  
public void clickOnRegistrationButton() {  
    registrationButton.click();  
}
```

☐ 1. részlet:

```
@FindBy(id = "idOfTheButton")
```

2. részlet:

```
public HomePage(final WebDriver webDriver) {  
    webDriver.findElements(By.id("idOfTheButton"));  
    this.webDriver = webDriver;  
}  
  
public void navigateToHomePage() {  
    webDriver.get(HOME_PAGE_URL);  
    registrationButton.click();  
}
```

☐ 1. részlet:

```
@RunWith(id = "idOfTheButton")
```

2. részlet:

```

public HomePage(final WebDriver webDriver) {
    webDriver.findElement(By.id("idOfTheButton"));
    this.webDriver = webDriver;
}

public void navigateToHomePage() {
    webDriver.load(HOME_PAGE_URL);
}

public void clickOnRegistrationButton() {
    registrationButton.click();
}

```

1. részlet:

```
@FindBy(id = "idOfTheButton")
```

2. részlet:

```

public HomePage(final WebDriver webDriver) {
    PageFactory.initElements(webDriver, this);
    this.webDriver = webDriver;
}

public void navigateToHomePage() {
    webDriver.get(HOME_PAGE_URL);
}

public void clickOnRegistrationButton() {
    registrationButton.click();
}

```

Magyarázat

Ahhoz, hogy a Selenium megtalálja a weboldalon egy WebElement értékét, meg kell adnunk a WebElement változó selectorát. Ezt a @FindBy annotációval adjuk meg.

Ahhoz, hogy a WebElement futáskor meg is kapja a selector által megtalált értéket, a Page objectet inicializálni kell. Ezt teszi meg a konstruktorban a PageFactory.initElements() a WebDriver segítségével.

Két különálló metódust kér a feladat az oldal betöltéséhez és a kattintáshoz.

Melyik leírás illik legjobban a kék-zöld (blue-green) telepítésre?

Válasz

- ☐ Az alkalmazás különböző verziói futnak kísérleti jelleggel, melyeket a felhasználók elkülönített csoportjai érnek el. A végső éles verzió kiválasztása a definiált metrikák elérése alapján történik.
- ☐ Az alkalmazás új verzióját a felhasználók folyamatosan növekvő részének teszik elérhetővé, ezzel arányosan frissítik az alkalmazás futó példányait.
- ☒ Két külön környezet áll rendelkezésre, az egyikben a régi verzió fut, a másikra az új verziót telepítik. Validálás és minőségbiztosítás után a forgalmat átirányítják az új verzióra.
- ☐ Az alkalmazás futó példányait kötegekben (batch) frissítik az új verzióra, miközben meghatározott számú példány végig elérhető állapotban van.

Magyarázat

Az első válasz: A/B telepítés

A második válasz: Canary telepítés

Az utolsó válasz: Rolling telepítés

[Legfontosabb tudnivalók](#)

[Kapcsolat](#)

[Versenyszabályzat](#)

[Adatvédelem](#)

© 2022 Human Priority Kft.

KÉSZÍTETTE

Megjelenés

 Világos 