

.NET FEJLESZTÉS C# NYELVEN

3. forduló



A kategória támogatója: DXC Technology

Ismertető a feladathoz

A 3.forduló feladatait a hosszú hétvége miatt kivételesen szerda (11.02.) éjfélig tudod megoldani!

Érdeemes ebben a fordulóban is játszani, mert a következő forduló kezdetekor, 11.03-án 18 órától kiosztjuk az 1.-2.-3. fordulóban megszerzett badgeket!

A verseny közben az alábbi teljesítményeket díjazzuk:

- fordulógyőztes
- átlagnál jobb időeredmény
- átlag feletti pontszám
- hibátlan forduló

Szeretnénk rá felhívni figyelmedet, hogy az egyszer megkapott badge-eket nem vonjuk vissza, akkor sem, ha esetleg az adott fordulóban a visszajelzések alapján változások vannak.

Jó játékot!

Felhasznált idő: 00:00/10:00

Elért pontszám: 0/15

1. feladat 0/5 pont

Az alábbi metódus hívások közül melyek **hibásak**?

```
class CarManager
{
    public bool ReserveCar(DateTime forecastDate, ReserveOptions options)
    {
        /* implementation */
    }
}

class ReserveOptions
{
    private string color { get; set; }

    public ReserveOptions()
    {
        this.color = "Red";
    }

    public ReserveOptions(string color)
    {
        this.color = color;
    }
}

var manager = new CarManager();
```

Válaszok



manager.ReserveCar(DateTime.Now(), new ReserveOptions());

Ez a válasz helyes, de nem jelölted meg.



manager.ReserveCar(DateTime.Now, new());



manager.ReserveCar(DateTime.Now, new ReserveOptions());



manager.ReserveCar(DateTime.Now, new ReserveOptions() { color = "Black" });

Ez a válasz helyes, de nem jelölted meg.



manager.ReserveCar(DateTime.Now, new ReserveOptions("Black"));



manager.ReserveCar(DateTime.Now, new("Black"));

Magyarázat

2. feladat 0/10 pont

Vizsgáld meg az alábbi kódrészleteket és a hozzá tartozó állításokat!

A feladat során feltételezheted, hogy .NET 6-ban íródott.

```
//Program.cs
IHost host = Host.CreateDefaultBuilder(args)
    .ConfigureServices(services =>
    {
        services.Configure<HostOptions>(options =>
        {
            options.BackgroundServiceExceptionHandler =
                BackgroundServiceExceptionHandler.Ignore;
        });
        services.AddHostedService<DoerService>();
    })
    .Build();
await host.RunAsync();

//DoerService.cs
public sealed class DoerService : BackgroundService
{
    protected override async Task ExecuteAsync(CancellationToken
stoppingToken)
    {
        await Task.Delay(1);
        throw new NotImplementedException("When I grow up, I will be a
doer.");
    }
}
```

Állítások:

1. A host szerviz terminálni fog, mert a DoerService futtatásakor kezeletlen exception van.
2. A host szerviznek nem lesz kezeletlen exceptionje a DoerService háttérszervíztől, mert ezt állítja a *BackgroundServiceExceptionHandler* kapcsoló.

Válaszok

- ☐ Az első állítás első tagmondata igaz.

☒ Az első állítás második tagmondata igaz.
Ez a válasz helyes, de nem jelölted meg.

☐ Az első állítás mindkét tagmondata igaz, helytelen a következtetés.

☐ Az első állítás mindkét tagmondata igaz, helyes a következtetés.

☐ Az első állítás mindkét tagmondata hamis.

☒ Az második állítás első tagmondata igaz.
Ez a válasz helyes, de nem jelölted meg.

☐ Az második állítás második tagmondata igaz.

☐ Az második állítás mindkét tagmondata igaz, helytelen a következtetés.

☐ Az második állítás mindkét tagmondata igaz, helyes a következtetés.

☐ Az második állítás mindkét tagmondata hamis.

Magyarázat

A host szervíz nem kapja meg, de ki lesz logolva. A kapcsolóval az állítható, hogy terminálást hívódjon-e a főszervízre.

<https://docs.microsoft.com/en-us/dotnet/core/compatibility/core-libraries/6.0/hosting-exception-handling>



Legfontosabb tudnivalók [↗](#)

Kapcsolat [↗](#)

Versenyszabályzat [↗](#)

Adatvédelem [↗](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE 

Megjelenés

 Világos 