.NET MICROSERVICES

2. forduló

A kategória támogatója: DXC Technology

Ismertető a feladatlaphoz

Kérjük, hogy a feladatlap indítása előtt mindenképp olvasd el az alábbi útmutatót:

> Helyezéseket a 4. forduló után mutatunk, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.
>
> A feltűnően rövid idő alatt megoldott feladatlapok kizárást vonnak maguk után, bármilyen más gyanús esetben fenntartjuk a jogot a forduló érvénytelenítésére!

Jó versenyzést kívánunk! Ez most már nem lesz olyan könnyű. :)

1. feladat 2 pont

Válaszd ki a record-okra vonatkozó helyes állításokat!

Válasz

- ( ) A class-okkal megegyező viselkedésű, de kevesebb kóddal akar egy construktorral megvalósítható adatosztály.

- Két record változó akkor egyenlő, ha azonos objektumra referálnak.
- ToString()-re a publikus property-ket és a record típus nevét fogja kiírni.
- Ebből is lehet leörökölni a record struct kulcsszó megadásával.
- A fentiek közül egyik sem.

2. feladat 1 pont

Mi az Options Pattern fő célja az alkalmazás fejlesztése során?

Válasz

- Az Options Pattern segítségével lehetőség van különböző adatbázis műveletek végrehajtására.
- Az Options Pattern olyan struktúrát definiál, amely lehetővé teszi a futás közbeni állapot változtatását.
- Az Options Pattern segítségével könnyedén kezelhetjük és konfigurálhatjuk az alkalmazás beállításait, például egy appsettings.json fájlból.
- Az Options Pattern a kód modulárisabbá tételét szolgálja a kódbázis minden részében.

3. feladat 1 pont

Mi a "docker-compose.yaml" fájlban a "volumes" szekció szerepe?

Válasz

- Adatok tartós tárolását teszi lehetővé a hoston.
- Elkülönített hálózati szegmens definiálását teszi lehetővé.
- Környezeti változók definiálását teszi lehetővé.
- A service replikáinak számosságát jelöli.

4. feladat 0 pont

Egy valós idejű webalkalmazást fejlesztesz, amely időzített eseményeket kezel aszinkronitás és delegáltak segítségével. Az alkalmazásban szükség van az események időzítésére és azok végrehajtására előre

meghatározott időpontokban.

Az alábbi kód egy példát mutat egy időzített eseménykezelő delegáltra:

```
public delegate Task TimedEventHandler(object sender, EventArgs e);
```

Az alkalmazásban olyan időzített eseményeket akarsz létrehozni, amelyek előre meghatározott időpontokban hívják meg az eseménykezelő delegáltat.

Melyik kódrészlet mutatja be helyesen egy aszinkron időzített esemény kezelését a megadott delegálttal?

Válasz

```
public async Task SetTimedEvent(TimedEventHandler handler, TimeSpan delay)
{
        await Task.Delay(delay);
        await handler.Invoke(this, EventArgs.Empty);
}
```

```
public async Task SetTimedEvent(TimedEventHandler handler, TimeSpan delay)
{
        var task = Task.Delay(delay);
        task.Wait();
        await handler(this, EventArgs.Empty);
}
```

```
public async Task SetTimedEvent(TimedEventHandler handler, TimeSpan delay)
{
        Task.Run(async () =>
        {
                await Task.Delay(delay);
                await handler(this, EventArgs.Empty);
        }).Wait();
}
```

```
public async Task SetTimedEvent(TimedEventHandler handler, TimeSpan delay)
{
        await Task.Delay(delay);
        handler.Invoke(this, EventArgs.Empty);
}
```

5. feladat 4 pont

Mi a következő kód kimenete?

```
var couldParse = Enum.TryParse("7", out Days day);
if (couldParse)
{
    Console.WriteLine(day);
}

public enum Days
{
    Monday = 0,
    Tuesday = 1,
    Wednesday = 2,
    Thursday = 3,
    Friday = 4,
    Saturday = 5,
    Sunday = 6,
}
```

Válasz

- ○ Nincs kimenete, mert couldParse értéke false lesz.
- ○ Nincs kimenete, mert a Enum.TryParse exception-t dob.
- ○ Monday
- ○ Sunday
- ○ 7

Megoldások beküldése