

1. Melyik láthatóság szint nem létezik C#-ban?
  - a) public
  - b) package
  - c) private
  - d) internal
2. A modell-nézet-nézetmodell-perzisztencia architektúrában melyik nem a modell felelőssége?
  - a) Az üzleti logika megvalósítása.
  - b) Az adatbázis sémájának leírása.
  - c) Jelzés küldése egy adattag értékének változásáról.
  - d) A kapott adatok validációja.
3. Mi az adatkötés (data binding)?
  - a) Adatbányászat, az alkalmazásban a nézet és a modell között történik.
  - b) Amikor az alkalmazásban található adattagok kezdeti értéket kapnak a modelltől.
  - c) Az adattagok automatikus validációja a nézeten.
  - d) A megjelenített és az üzleti logikában tárolt adat értesítés alapú összekötése.
4. Az alábbiak közül melyik megközelítési mód nem létezik entitás adatmodellek létrehozására?
  - a) database first
  - b) modell first
  - c) object first
  - d) code first
5. Az alábbiak közül melyik állítás helyes C# nyelvre?
  - a) A C# nyelv támogatja a többszörös öröklődést.
  - b) A C# nyelvben egy osztály implementálhat több interfészt.
  - c) A struktúra referencia szerint kezelt.
  - d) Az osztály alapértelmezett láthatósága public.
6. Melyik állítás nem teljesül modell-nézet architektúra esetén?
  - a) A felhasználó a modellel kommunikál, a modell és a nézet egymással.
  - b) A nézet tartalmazza a grafikus felhasználói felület megvalósítását, beleértve a vezérlőket és eseménykezelőket.
  - c) A modell nem függ a nézettől.
  - d) A modell tartalmazza a háttérben futó logikát.
7. Az erőforrások felszabadításával kapcsolatban melyik állítás hamis a C# nyelvre?
  - a) A referencia szerinti változók törlését a szemétyűjtő felügyeli.
  - b) A programozó a destruktorközvetlen meghívásával is felszabadíthatja a memóriaterületet.
  - c) A nem-menedzselt erőforrások felszabadítására akkor van lehetőségünk, ha az osztály megvalósítja az IDisposable interfészt és benne a Dispose() metódust.
  - d) A szemétyűjtő végrehajtása nem garantálható, így nő a memóriahasználat.
8. Melyik típusnak felel meg C#-ban az int kulcsszóval rövidített típus?
  - a) System.Int32
  - b) System.Int64
  - c) System.UInt32
  - d) Implementáció (fordító) függő

## 9. Az alábbi XAML leírások közül mely érvénytelen?

- a) `<Window [...]>`  
    `<Grid>`  
        `<Label>Hello World 1</Label>`  
        `<Label>Hello World 2</Label>`  
    `</Grid>`  
`</Window>`
- b) `<Window [...]>`  
    `<Canvas>`  
        `<Label>Hello World 1</Label>`  
        `<Label Canvas.Top="50">Hello World 2</Label>`  
    `</Canvas>`  
`</Window>`
- c) `<Window [...]>`  
    `<StackPanel>`  
        `<Label>Hello World 1</Label>`  
        `<Label>Hello World 2</Label>`  
    `</StackPanel>`  
`</Window>`
- d) `<Window [...]>`  
    `<Label>Hello World 1</Label>`  
    `<Label>Hello World 2</Label>`  
`</Window>`

10. Adott az alábbi `Person` típus és a `people` változó, amelynek típusa `List<Person>`, mi lesz a következő LINQ lekérdezés eredménye:

```
var result = people.Where(p => p.Age > 18)
                .Select(p => p.Name).OrderBy(p => p);
```

```
public struct Person
{
    public int Age { get; set; }
    public string Name { get; set; }
}
```

- a) Fordítási hiba.
- b) A tartalmazott nagykorú személyek objektumai, elődlegesen életkor, majd név szerint rendezve.
- c) A tartalmazott nagykorú személyek nevei, név szerint rendezve.
- d) A tartalmazott nagykorú személyek nevei, életkor szerint rendezve.
11. Mi az egységteszt?
- a) A teljes alkalmazás egy egységben történő tesztelése.
- b) Az Entity Framework egységeinek (entitásainak) tesztelése.
- c) Az alkalmazás komponensei (egységei) közötti integráció tesztelése.
- d) Egyes osztályok és objektumok önálló viselkedésének tesztelése.
12. Melyik nem a Windows Presentation Foundation (WPF) előnye?
- a) Széleskörű, cross-platform felhasználhatóság.
- b) A 3D grafikus kártyák kihasználásának lehetősége.
- c) A felület (nézet) deklaratív leírásának lehetősége.
- d) Háromrétegű (MVVM) architektúra támogatása.

13. Az alábbiak közül melyik állítás hamis?
- a) Az egyes rétegek között függőségek alakulnak ki, mivel felhasználják egymás funkcionalitását.
  - b) A függőség befecskendezés (dependency injection) jelentése, hogy a rétegek a függőségeknek csak az absztrakcióját látják, a konkrét megvalósítását külön adjuk át nekik.
  - c) A függőségeket úgy kell megvalósítani, hogy a konkrét megvalósítástól függenek.
  - d) A befecskendezésnek különböző módjai lehetnek (például: konstruktor, metódus).
14. Milyen programnyelvet vagy struktúrált leíró nyelvet használhatunk a WPF alkalmazás nézeteinek leírásához?
- a) XML
  - b) XAML
  - c) YAML
  - d) JavaScript
15. Az alábbi osztályok közül melyik .NET Framework osztály implementálja az INotifyCollectionChanged interfészt?
- a) List
  - b) Vector
  - c) Label
  - d) ObservableCollection