

TERVEZÉSI MINTÁK

3. forduló



A kategória támogatója: IBM

Ismertető a feladathoz

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

Érdemes 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: 00:00/05:00

Elért pontszám: 0/4

1. feladat 0/1 pont

A tervezési minták milyen esetben növelik az alkalmazás teljesítményét?

Válasz

- ☐ Minden minta növeli a teljesítményt
- ☐ A mintáknak nem céljuk a teljesítmény növelése, de előfordulhat, mint járulékos előny
- ☒ Csak néhány mintának célja a teljesítmény növelése
Ez a válasz helyes, de nem jelölted meg.
- ☐ Semelyik minta nem növeli a teljesítményt

Magyarázat

Bármelyik minta képes lehet teljesítenynövelésre a megfelelő környezetben, de vannak kifejezetten a teljesítmény növelését célzó minták.

2. feladat 0/1 pont

Mely állítások igazak a Prototype mintára?

Válaszok

- ☒ Leegyszerűsíti a bonyolultan létrehozható objektumok példányosítását
Ez a válasz helyes, de nem jelölted meg.
- ☒ Az alapgondolata egy létező objektum másolása
Ez a válasz helyes, de nem jelölted meg.
- ☐ Az osztály alapú öröklést használó nyelvekben bevezeti a prototipikus öröklést
- ☐ A létjogosultsága lecsökkent azoknál a nyelveknél, amik pointerek helyett referenciákat használnak
- ☐ Egyik sem

Magyarázat

Az öröklés modellje és a pointerek/referenciák használata olyan nyelvi sajátosságok, amik függetlenek a tervezési mintáktól, így ez a két válasz hamis.

3. feladat 0/1 pont

Az alkalmazásunkban szeretnénk támogatni a műveletek visszavonását. Mely minták alkalmasak erre?

Válaszok

- ☒ Command
Ez a válasz helyes, de nem jelölted meg.
- ☒ Memento
Ez a válasz helyes, de nem jelölted meg.
- ☐ Reverter
- ☐ State
- ☐ Egyik sem

Magyarázat

Reverter minta nem létezik.

A State lényege az alkalmazás viselkedését az aktuális állapottól függővé tenni, nem az állapotok tárolásának kezelése.

A Command segítségével kéréseket tudunk objektumokba csomagolni, amik így akár visszavonhatóak is lehetnek.

A Memento az aktuális állapot mentésére és visszatöltésére szolgál.

4. feladat 0/1 pont

Adott a következő kódrészlet:

```
class Writer {  
    void writeLine(String line) {  
        // implementation  
    }  
}
```

A célunk az Adapter mintával a Writer szolgáltatásainak elérhetővé tétele a Printer interface-en keresztül:

```
interface Printer {  
    void println(String line);  
}
```

Eddig jutottunk:

```
class Adapter /* 1 */ {  
    // 2  
}
```

Hogyan egészítsük ki a fenti kódot, hogy elérjük a célunkat?

Válaszok

☒ Az /* 1 */ helyére:

```
extends Writer implements Printer
```

A // 2 helyére:

```
@Override  
public void println(String line) {  
    writeLine(line);  
}
```

Ez a válasz helyes, de nem jelölted meg.

☒ Az /* 1 */ helyére:

```
implements Printer
```

A // 2 helyére:

```
Writer writer = new Writer();  
  
@Override  
public void println(String line) {  
    writer.writeLine(line);  
}
```

Ez a válasz helyes, de nem jelölted meg.

☐ Az /* 1 */ helyére:

```
extends Writer
```

A // 2 helyére:

```
Printer printer = new Printer();

@Override
void writeline(String line) {
    printer.println(line);
}
```

☐ Az /* 1 */ helyére:

```
extends Writer implements Printer
```

A // 2 helyére:

```
@Override
void writeline(String line) {
    println(line);
}
```

☐ Egyik sem

Magyarázat

A Printer interface-t mindenképpen meg kell valósítani, mert enélkül nem történik meg az adaptálás. Ezért a println() metódust is kötelezően felül kell definiálni, mert absztrakt.

Attól függően, hogy class vagy object adaptert használunk, a Writer képességeinek újrahasználása örökléssel vagy kompozícióval történik.



[Legfontosabb tudnivalók](#) [Kapcsolat](#) [Versenyszabályzat](#) [Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos