

## 7. Tervezési minták I. (Vermek és Fák)

Határidő ápr 8, 23:59

Pont 10

Kérdések 10

Időkorlát Nincs

Engedélyezett próbálkozások 5

[Kvíz kitöltése újra](#)

### Próbálkozások naplója

	Próbálkozás	Idő	Eredmény
MEGTARTOTT	<a href="#">2. próbálkozás</a>	4 perc	7.25 az összesen elérhető 10 pontból
LEGUTOLSÓ	<a href="#">2. próbálkozás</a>	4 perc	7.25 az összesen elérhető 10 pontból
	<a href="#">1. próbálkozás</a>	79 perc	6.5 az összesen elérhető 10 pontból

⚠ A helyes válaszok el vannak rejtve.

Ezen próbálkozás eredménye: **7.25** az összesen elérhető 10 pontból

Beadva ekkor: ápr 8, 15:33

Ez a próbálkozás ennyi időt vett igénybe: 4 perc

#### 1. kérdés

1 / 1 pont

Milyen célt szolgál az egyke (singleton) tervezési minta?

- ☐ Memória szivárgás megszüntetését
- ☐ Futási idő javítását
- ☐ Kód-redundancia megszüntetését
- ☒ Memória pazarlás mérséklését

Helytelen

**2. kérdés****0 / 1 pont**

Milyen szoftvertervezési célt fejez ki az Open-Closed elv?

☐

A fejlesztőknek az alkalmazások készítésekor nyitottnak kell lenni a felhasználói igények befogadására, de miután a specifikációt rögzítették, nem szabad teret adni a tervezés és a megvalósítás során a specifikációtól eltérő felhasználói igényeknek.

☒

Azt, hogy az adatbázisok kulcsát meg kell őrizni.

☐

Egy szoftver későbbi módosítása során a már meglévő osztályokat, modulokat, metódusokat elég legyen csak új elemekkel kiegészíteni, meglévőkön változtatni ne kelljen.

☐

A perzisztenciához használt háttérfájlokat mindig meg kell nyitni a használat előtt, és le kell zárni azt követően.

**3. kérdés****1 / 1 pont**

Mi a stratégia tervezési minta lényege?

☒

Biztosítja, hogy egy metódus részét alkotó tevékenység rugalmasan megválasztható, lecserélhető legyen

☐

Olyan osztály-hierarchiát definiál, ahol az alosztályok egykék, azaz legfeljebb egy példány hozható létre belőlük



Segítségével rugalmasan változtatható egy objektum szerkezete: attól függően, hogy a kapcsolódó interfész melyik implementációjának példányát aggregáljuk



Hasonló tevékenységeknek kapcsolatát: származtatási hierarchiáját mutatja meg

#### 4. kérdés

1 / 1 pont

Mi az interfész szegregáció lényege?



Két objektum sohase kerüljön közvetlen kapcsolatba, csak egy interfész közvetítésével.



Az azonos felelősségi körhöz tartozó, de egymás alternatíváját kínáló objektumok osztályait elkülönítjük a kód többi részétől és azokat egy közös interfészből származtatjuk.



A származtatási hierarchiák azon osztályainak, amelyekből származtatunk, de amelyeknek nincs ősz osztályuk, mindig interfészeknek kell lenniük.



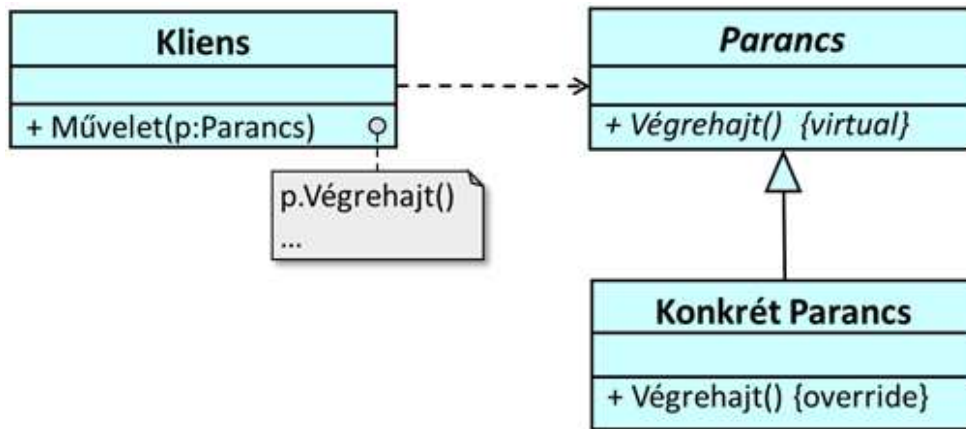
Az alkalmazásban bevezetett interfészeket a céljuk és fajtájuk szerint külön-külön csomagokba kell szervezni.

Helytelen

#### 5. kérdés

0 / 1 pont

Mi olvasható ki az alábbi tervből?



Egy KonkrétParancs objektum Végrehajt() metódusa felhasználja egy Kliens objektumnak a Művelet() metódusát.



A Kliens típusú objektumok akár a Parancs osztály, akár az abból származtatott KonkrétParancs osztály Végrehajt() műveletét használhatják.



Minden Kliens típusú objektum kapcsolatban áll legalább egy Parancsból származtatott osztály objektumával.



A Kliens típusú objektumok Művelet() metódusa a paraméterként megkapott parancs objektumtól függ.

## 6. kérdés

1 / 1 pont

Mi a különbség az absztrakt osztály és az interfész között?



Az interfészből mindig származtatni kell egy osztályt, az absztrakt osztályból nem.



Nincs közöttük különbség.



Az absztrakt osztálynak lehet olyan metódusa, amelyiknek definíciója is van.



Az interfésznek lehetnek adattagjai is.

## 7. kérdés

1 / 1 pont

Hasonlítsa össze az sablonfüggvény és a stratégia tervezési mintákat?



Az állapot tervezési minta a kliens objektum egy adattagjának (állapotának) lehetséges értékeit írja le, és metódusokat biztosít ezen értékek megváltoztatásához; a stratégia tervezési minta egymással helyettesíthető algoritmusokat definiál a kliens számára.



Mindkét tervezési minta kiszervezi a kliens objektum egy résztvevénységét egy másik objektumba azért, hogy ezt a résztvevénységet futási időben rugalmasan változtathassuk meg.



Ezen tervezési minták egyaránt támogatják a SOLID elvek közül a nyitott-zárt, az interfész szegregáció és a függőség megfordításának elvét.



Az állapot tervezési minta kompozícióval köt egy interfészt a kliens osztályhoz, a stratégia tervezési mintában ez a kapcsolat lehet aggregáció, vagy akár egyszerű függőség is.

## 8. kérdés

1 / 1 pont

Milyen tevékenység objektumot kellene tervezni annak a feladatnak a megoldására, amely kiválogatja egy bináris fa csúcsaiban tárolt egész számok közül a páros számokat, és azokat egy sorozatba fűzi?

☐

```
class Assort: IAction
{
    public void Exec(Node node)
    {
        if (node.Value % 2 == 0) return this;
    }
}
```

☒

```
class Assort: IAction
{
    public List<int> seq { get; private set; }
    public void Exec(Node node)
    {
        if (node.Value % 2 == 0) seq.Add(node.Value);
    }
}
```

☐

```
class Assort: IAction
{
    public void Exec(Node node)
    {
        if (node.Value % 2 == 0) return node.Value;
    }
}
```

☐

```
class Assort: IAction
{
    public List<int> seq { get; private set; }
    public void Exec(BinTree t)
    {
        foreach( node in t ) if (node.Value % 2 == 0) seq.Add(node.Value);
    }
}
```

Részleges

9. kérdés

0.25 / 1 pont

Milyen tevékenység objektumot kellene tervezni annak a feladatnak a megoldására, amely eldönti, hogy egy bináris fa csúcsaiban tárolt egész számok között csupa páros szám van-e?



```
class Decision : IAction
{
    public bool All { get; private set; } = true;
    public void Exec(Node node)
    {
        All = All && node.Value % 2 == 0;
    }
}
```



```
class FoundOddNumber : Exception { }
class Decision : IAction
{
    public bool All { get; private set; } = true;
    public void Exec(Node node)
    {
        All = All && node.Value % 2 == 0;
        if(!All) throw new FoundOddNumber();
    }
}
```



```
class FoundOddNumber : Exception { }
class Decision : IAction
{
    public bool All { get; private set; } = true;
    public void Exec(Node node)
    {
        if(node.Value % 2 != 0) { All = false; throw new FoundOddNumber
        ("); }
    }
}
```

```
class Decision : IAction
{
    public bool All { get; private set; } = true;
    public void Exec(Node node)
    {
        if(node.Value % 2 != 0) { All = false; }
    }
}
```

☐

## 10. kérdés

1 / 1 pont

Milyen tevékenység objektumot kellene tervezni annak a feladatnak a megoldására, amely egyrészt kiválogatja egy bináris fa csúcsaiban tárolt egész számok közül a páros számokat, és azokat egy sorozatba fűzi, másrészt eldönti, hogy egy bináris fa csúcsaiban tárolt egész számok között csupa páros szám van-e?

```
class DecisionAssortment : IAction
{
    public List<int> seq { get; private set; }
    public bool All { get; private set; } = true;
    public void Exec(Node node)
    {
        if(node.Value % 2 == 0) { All = true; seq.Add(node.Value); }
        else { All = false; }
    }
}
```

☐

```
class FoundOddNumber : Exception { }
class DecisionAssortment : IAction
{
    public List<int> seq { get; private set; }
    public bool All { get; private set; } = true;
    public void Exec(Node node)
    {
        All = node.Value % 2 == 0;
        if(All) seq.Add(node.Value);
        else throw new FoundOddNumber();
    }
}
```

☐



☐ class DecisionAssortment : IAction  
{  
    public List<int> seq { get; private set; }  
    public bool All { get; private set; } = true;  
    public void Exec(Node node)  
    {  
        All = All && node.Value % 2 == 0;  
        if(All) { seq.Add(node.Value);}  
    }  
}

☒ class DecisionAssortment : IAction  
{  
    public List<int> seq { get; private set; }  
    public bool All { get; private set; } = true;  
    public void Exec(Node node)  
    {  
        if(node.Value % 2 == 0) seq.Add(node.Value);  
        else All = false;  
    }  
}

Kvízeredmény: **7.25** az összesen elérhető 10 pontból