El?z? - Fel - Következ?

Tartalomjegyzék

Örökl?désr?l

Többszörös örökl?dés

Egy osztály nem tud egynél több osztályból örökl?dni, vagyis az alábbi hibás:

```
public class A
{
    ...
}

public class B
{
    ...
}

public class C extends A, B
{
   // compile error
}
```

De lehet több interface-t implementálni:

```
public interface A
{
    ...
}

public interface B
{
    ...
}

public class C implements A, B
{
    // mind A és B metódusait is implementálni kell.
}
```

Részleges implementálás

Lehet egy **interface**-t részben is implementálni, ha az implementáció maga absztrakt osztály.

```
public interface A
{
    ...
}

public abstract class B implements A
{
    // A bizonyos metódusait implementálhatom, bizonyosakat nem
}

public class C extends B
{
    // Itt a maradék metódust is implementálni kell.
}
```

Final

Egy *osztály is lehet **final*** (végleges), ekkor nem lehet bel?le örökl?dni! Ez akkor jó, ha garantálni akarom, hogy más ne tudjon olyan kódot az enyémhez illeszteni, ami (override segítségével) megváltoztatja az én kódomat.

```
public class A
{
    public float f(float x)
    {
         // számol valamit
    }
}

public class B extends A
{
    public float f(float x)
    {
         // számol más valamit
    }
}
```

Ekkor nem lehetek biztos abban, hogy az **A** típusú objektumaim az én függvényemet használják.

```
A a;
...
a.f(3.14f); // lehet B.f é A.f is
```

Ha **A** final, akkor a fentebbit kivédhetem.

Megjegyzés

Egy osztály nem lehet egyszerre absztrakt és final! Bár nem feltétlenül értelmetlen, de kérdéses, hogy mire is lenne jó.

Final method

```
public class A
{
    public final float f(float x)
    {
         // számol valamit
    }
}
```

Feladat

Legyen a `Funtion` és `Bijection` absztrakt. Függvénynek legyen `compute` metódusa, bijekciónak `inverse` metódusa.

```
Function <-- Bijection <-- Power
Function <-- Sin
Function <-- Cos
```

Vagy lehet absztrakt `Periodic` is.

```
Function <-- Periodic <-- Sin
```

El?z? - Fel - Következ?