

El?z? - Fel - Következ?

## Tartalomjegyzék

- 1 Abstract
- 2 Feladatok
  - ◆ 2.1 Síkbeli objektumok
  - ◆ 2.2 Point
  - ◆ 2.3 Immutable

### Abstract

Egy interface (lásd el?z? gyakorlat) egyetlen metódusa sincsen implementálva, adattagja is csak bizonyos tulajdonságú lehet. Egy **abstract** osztály azonban keverheti az implementált és implementálatlan metódusokat, adattagja is tetsz?leges lehet.

```
public abstract class A {
    protected float x_;
    public A() // default konstruktor
    {
        x_ = 0.0f;
    }
    public A(float x) // konstruktor
    {
        x_ = x;
    }
    public void SetX(float x)
    {
        x_ = x;
    }
    public abstract float f(); // nincsen implementálva
}
```

**abstract**-al illehetünk egy metódust, ekkor kötelez?en a bennfoglaló osztály is **abstract** kell legyen. De **abstract** osztálynak lehetnek *nem abstract*, azaz implementált metódusai.

**abstract** osztály nem példányosulhat, de bel?le származtatott osztály igen, ha minden absztrakt metódusát implementálja (már nem absztrakt többé).

```
public class B extends A {
    public B(float x) // konstruktor meghívja az ?osztály konstruktorát
    {
        super(x);
    }
    public float f()
    {
        return (float)Math.sin(x_);
    }
}
```

Ekkor **B** már nem absztrakt, lehet példányosítani:

```
A a = new B(3);
a.f(); // valójában a B.f fut le (override)
```

## Feladatok

### Síkbeli objektumok

Készítsunk egy **Point** osztályt, mely tartalmaz

- két koordinátát
- eltolás metódust
- setter/getter-t a koordinátákra

Készítsunk egy **PlanarObject** osztályt, mely tartalmaz

- **Point** tömböt, amiben koordináták vannak
- egy **translate** metódust, ami a koordinátáinak mindegyikét eltolja (a **Point.translate** metódussal)
- egy **area** metódust ami legyen absztrakt (még nem implementáljuk)

Örökl?djön a **PlanarObject** osztályból egy **Square** és egy **Triangle** osztály, melyek **nem absztraktok!**

### Point

#### *Halloween-i örökl?dés*

Legyen a Point osztály maga is egy *PlanarObject*!

### Immutable

Oldjuk meg örökl?déssel/interface-el hogy minden osztályunkból legyen egy **Immutable** el?tagú is, ami immutable (nem lehet eltolni, nincs setter-e).

```
ImmutablePoint <-- Point
ImmutablePlanarObject <-- PlanarObject
ImmutablePlanarObject <-- ImmutableSquare <-- Square
PlanarObject <-- Square
```

[El?z?](#) - [Fel](#) - [Következ?](#)