

Nowoczesne języki programowania obiektowego

Dekorator



Dekorator (Decorator)

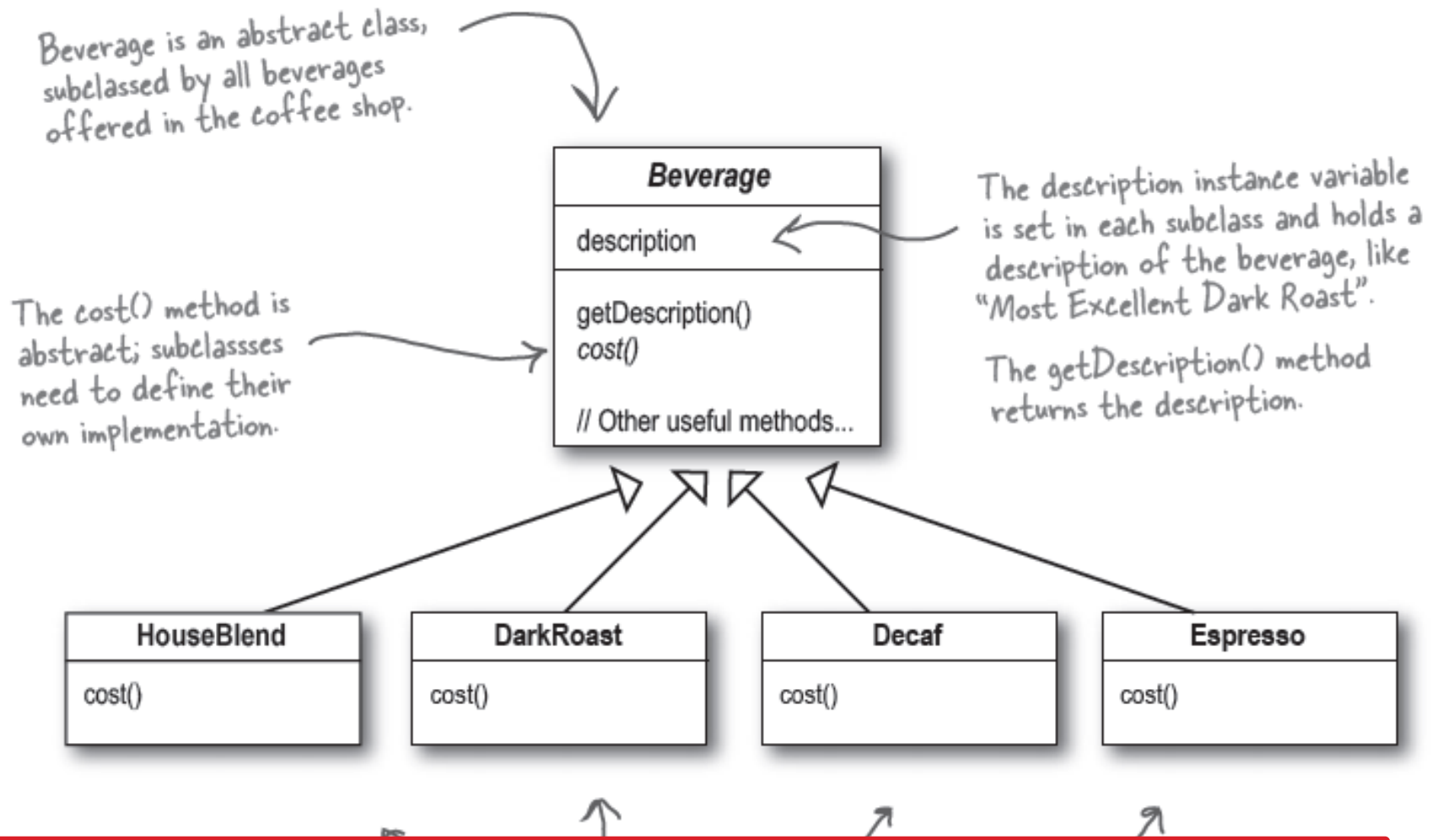
Cel

- ▶ Dodawanie funkcjonalności do obiektu w czasie pracy programu (at run time)
- ▶ Alternatywa dla standardowego procesu dziedziczenia (at compile time)

Zastosowania:

- ▶ Łatwe dodawanie zmian w programie.

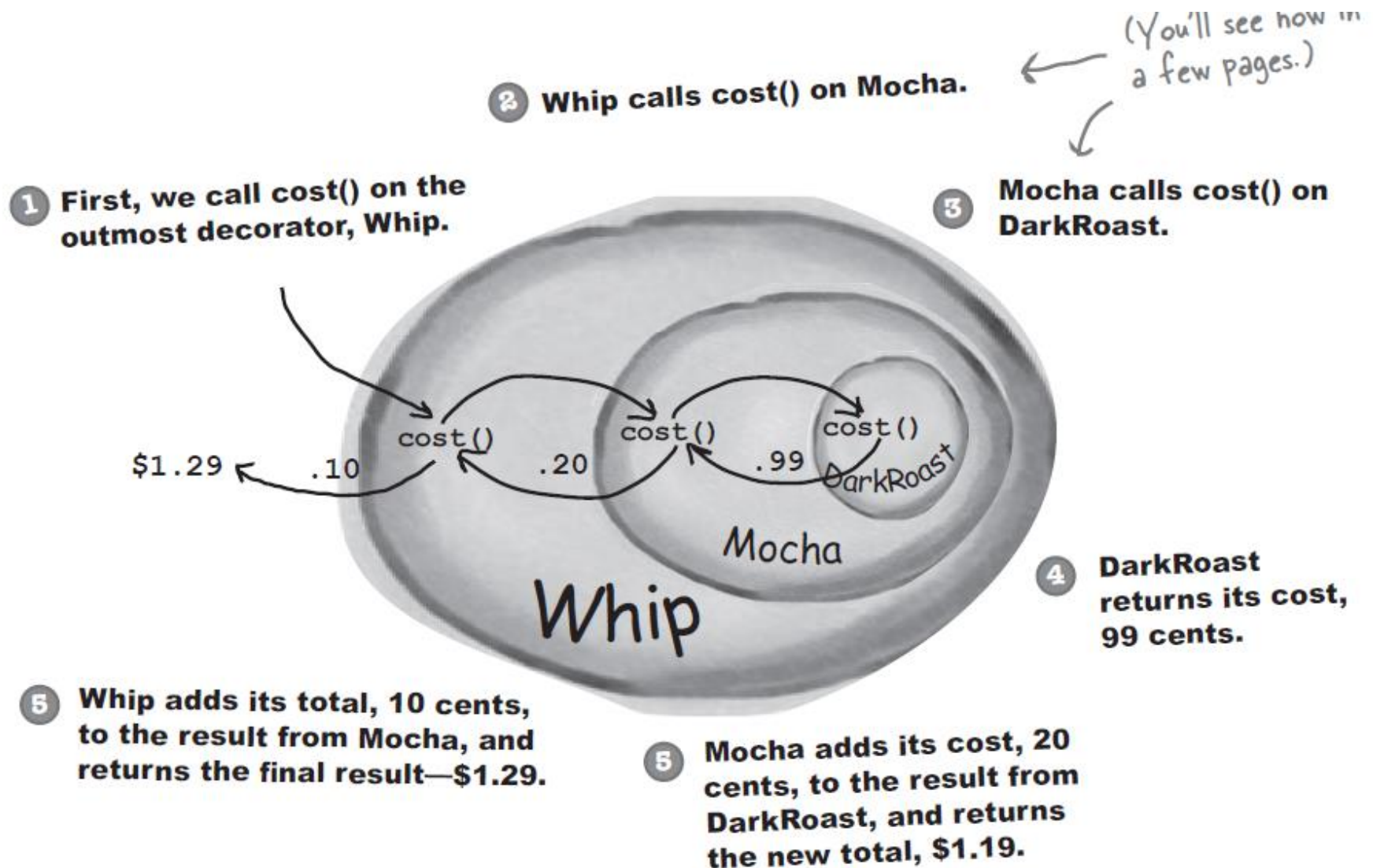
Bez zastosowania dekoratora



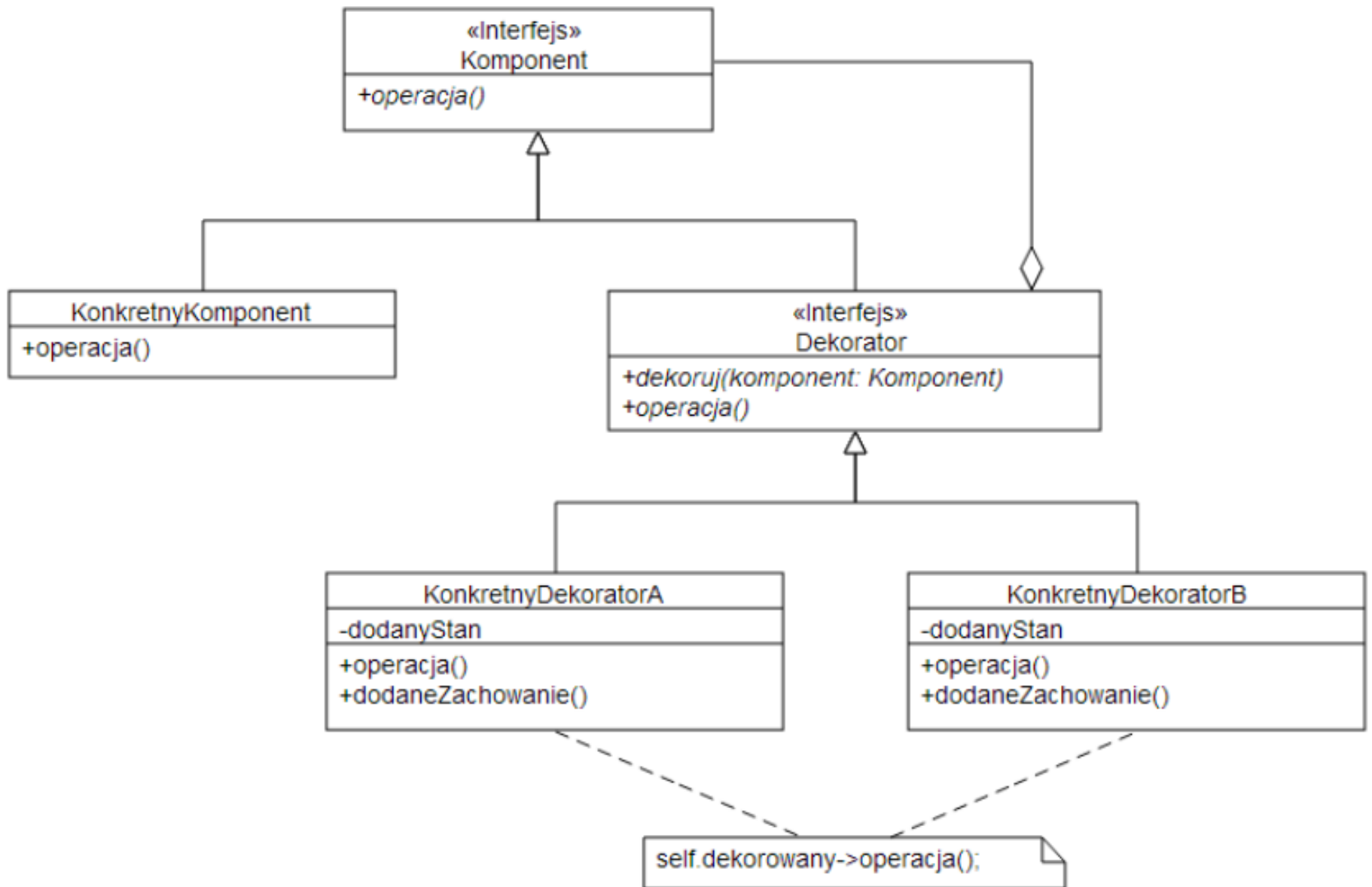
A jak chcemy jeszcze herbaty, szejki, pitne jogurty itp.? Mnóstwo klas?

Each subclass implements `cost()` to return the cost of the beverage.

Z zastosowaniem dekoratora

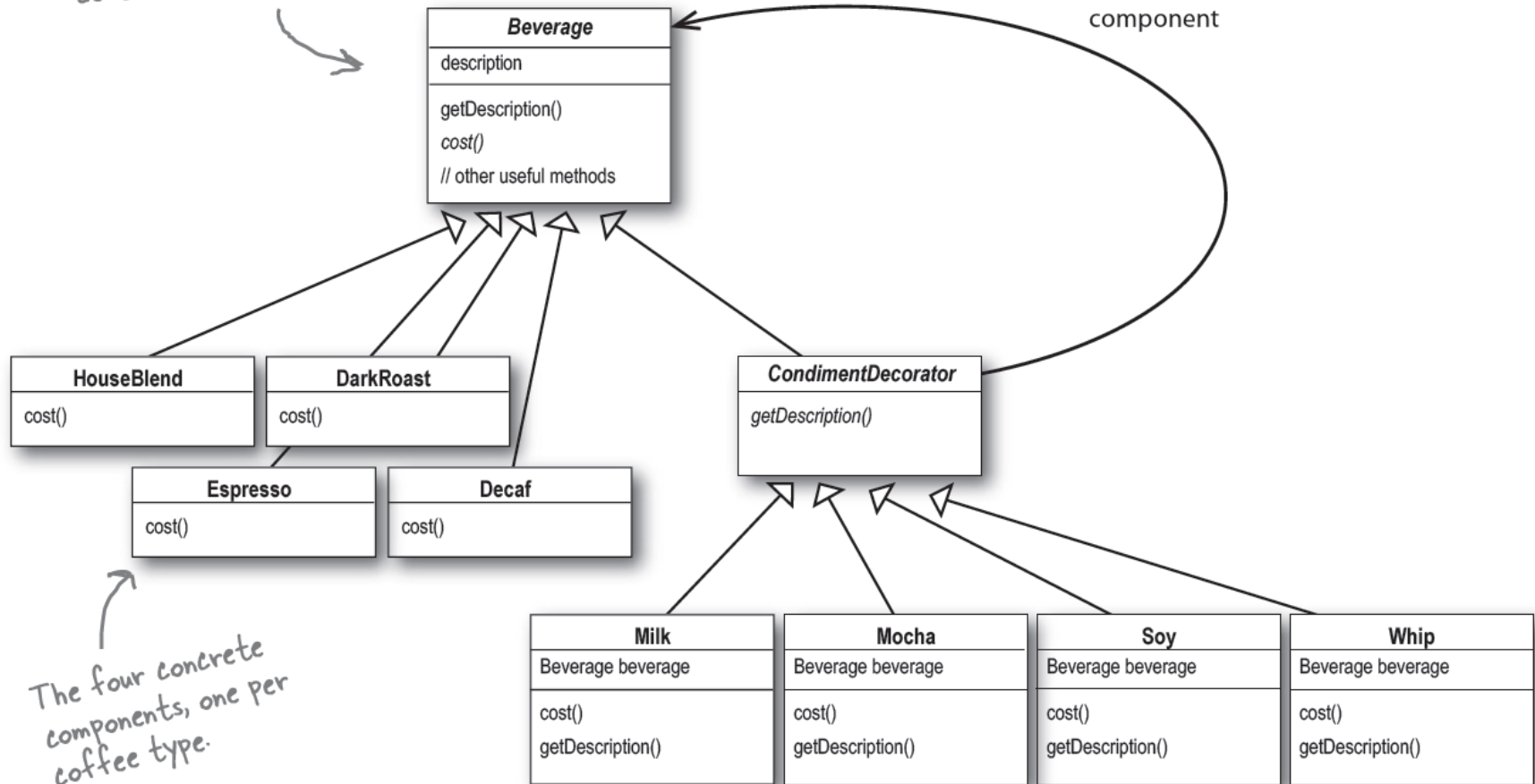


Dekorator – struktura



Dekorator – przykład

Beverage acts as our abstract component class.



The four concrete components, one per coffee type.

And here are our condiment decorators; notice they need to implement not only `cost()` but also `getDescription()`. We'll see why in a moment..

```
public abstract class Beverage {  
    String description = "Unknown Beverage";  
    public String getDescription() {  
        return description;  
    }  
    public abstract double cost();  
}
```

```
public abstract class CondimentDecorator extends Beverage {  
    public abstract String getDescription();  
}
```

```
public class Espresso extends Beverage {  
    public Espresso() {  
        description = "Espresso";  
    }  
    public double cost() {  
        return 1.99;  
    }  
}
```

Dekorator implementacja c.d.

```
public class Mocha extends CondimentDecorator {  
    Beverage beverage;  
    public Mocha(Beverage beverage) {  
        this.beverage = beverage;  
    }  
    public String getDescription() {  
        return beverage.getDescription() + ",  
Mocha";  
    }  
    public double cost() {  
        return .20 + beverage.cost();  
    }  
}
```

Inny przykład: [http://pl.wikisource.org/wiki/Dekorator_\(wzorzec_projektowy\)/kod](http://pl.wikisource.org/wiki/Dekorator_(wzorzec_projektowy)/kod)

Ćwiczenia

- ▶ Wykonaj prostą grę polegającą na symulacji zachowań drogowych
- ▶ Zdefiniuj klasy użytkowników drogi (pieszych, samochodów, rowerów).
- ▶ Zadbaj o ich dekorowanie z abstrakcyjnej klasy bazowej `UżytkownikDrogi`
- ▶ Dzięki dekoratorowi `Pieszcy` może stać się `Rowerzystą` (wystarczy, że wsiądzie na rower), itp.
- ▶ Każdy z obiektów danej klasy potrafi się poruszać po planszy (konsoli) z różną prędkością. Każdy z obiektów ma również swój odpowiedni znaczek.
- ▶ Obiekty poruszają się po planszy w sposób losowy, gra kończy się w momencie „kraksy”.
- ▶ Zadbaj o krokową pracę (aby można było podglądać każde stadium).