

Specyfikatory dostępu

Pakiety

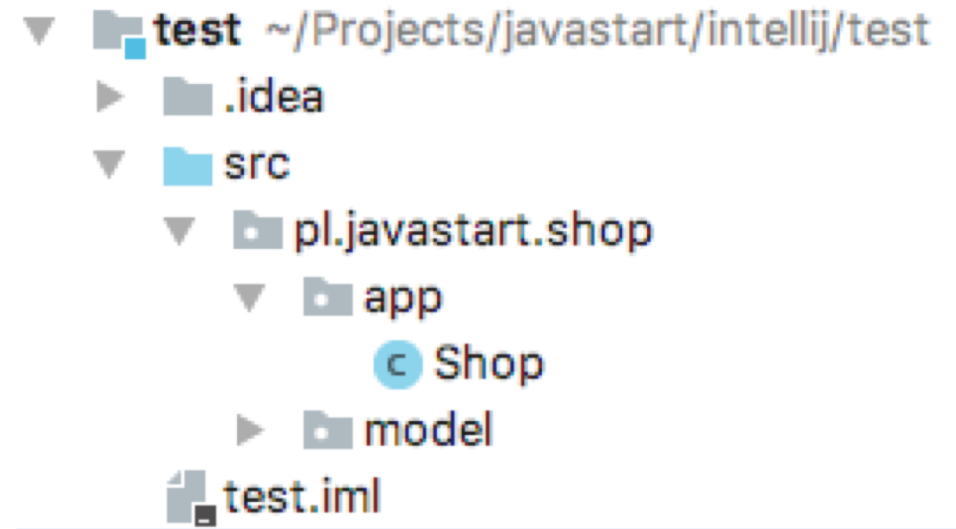
dostępne specyfikatory / modyfikatory

JavaBeans

Pakiety

Pakiety, reprezentowane przez foldery / subfoldery w naszym projekcie i pozwalają m.in. na:

- lepszą organizację kodu – logiczne grupowanie klas
- unikanie konfliktów nazw (klasy o takich samych nazwach)
- pozwalają ograniczyć dostęp do wybranych klas lub ich składowych
- nazewnictwo
com.example.projekt.pakiet



Pakiety (demo)

W celu wskazania w jakim pakiecie znajduje się dana klasa, w nagłówku należy dodać informację ***package nazwa.naszego.pakietu***

```
package pl.javastart.app;

public class Shop {
    //...
}
```

Odwołując się do klasy z innego pakietu, musimy ją zaimportować poprzez ***import nazwa.pakietu.NazwaKlasy***

```
package pl.javastart.app;
import pl.javastart.model.Product;

public class Shop {
    public static void main(String[] args) {
        Product prod = new Product();
    }
}
```

Specyfikatory dostępu (demo)

Składowe klas (pola, metody, konstruktory) w Javie możemy oznaczyć jednym z poniższych specyfikatorów, ograniczając możliwość odwołania się do nich.

private – dostęp do składowej tylko w ramach klasy

protected – dostęp także z innych klas w ramach tego samego pakietu

public – dostęp ogólny z różnych pakietów

brak specyfikatora jest (prawie) identyczny z ograniczeniem **protected**

```
package pl.javastart.model;

public class Product {

    private String name;
    private String producer;
    private int amount;

    public Product(String n, String p, int a) {
        name = n;
        producer = p;
        amount = a;
    }

    protected void showInfo() {
        System.out.println(name + " "
            + producer + " "
            + amount);
    }
}
```

Specyfikacja JavaBeans i enkapsulacja/hermetyzacja

- JavaBeans jest standardem definiowania klas w Javie, który może być wymagany przez niektóre biblioteki.
- W praktyce większość klas powinna maksymalnie ograniczać dostęp do swoich składowych, a publicznie udostępniać tylko elementy, z których chcemy, aby inni korzystali (API)
- Najczęściej konwencja wykorzystywana jest w stosunku do klas reprezentujących dane (model danych)
- JavaBeans powinny posiadać:
 - bezparametrowy konstruktor
 - zestaw metod set i get dla pól klasy (w JavaBeans nazywanych właściwościami / properties)
- Klasa może oczywiście posiadać dodatkowe metody i konstruktory

JavaBeans przykład (demo)

```
public class Product {  
  
    private String name;  
    private String producer;  
  
    public Product() {  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getProducer() {  
        return producer;  
    }  
  
    public void setProducer(String producer) {  
        this.producer = producer;  
    }  
}
```

```
public class Shop {  
    public static void main(String[] args) {  
        Product prod = new Product();  
        prod.setName("Mleko");  
        prod.setProducer("Mlekowita");  
        System.out.println(prod.getName()  
            + " " + prod.getProducer());  
    }  
}
```

Ćwiczenie

Na początku stwórz program składający się z dwóch klas:

- Book – posiadającą trzy publiczne pola: title, author, isbn i metodę showInfo() wyświetlającą informacje o danej książce.
- Library – z metodą main, w której utworzysz przykładową książkę i wyświetlisz o niej informacje

Następnie utwórz dwa pakiety:

- com.example.library.model
- com.example.library.app

Klasę Book przenieś do pakietu model i przekształć ją do formy JavaBeans, a klasę Library przenieś do pakietu app i napraw w niej wszelkie błędy powstałe w międzyczasie