

Tworzenie pakietów

1. Po co są pakiety?
2. Jak tworzymy pakiety?
3. Zasięg klas (słowo kluczowe **protected**)
4. Interfajsy (implement)

Stwórz nowy projekt w IntelliJ, w którym stworzysz następujące pakiety:

elektryk.twoje_imie_nazwisko – pakiet główny z całym kodem programu

elektryk.twoje_imie_nazwisko.utils – w którym będą znajdowały się klasy odpowiedzialne za zapis i odczyt danych

elektryk.twoje_imie_nazwisko.model – w którym będą znajdowały się klasy przechowujące dane

Jeśli przed klasą użyjesz słowa kluczowego **protected** to klasy tej będzie można używać tylko w obrębie **pakietu** w którym się ona znajduje.

Interface:

Tworzymy tak samo jak klasę - różnica w zapisie:

```
public interface CarImp {  
  
}
```

Używanie:

```
public class Car implements CarImp{  
  
}
```

Klasa Car implementuje (używa) interfejsu CarImp.

Interfejsy opisują co dana klasa ma zrobić, czyli jaką metodę **musi** posiadać dana klasa. Jest to bardzo przydatne jeśli tworzymy kilka różnych klas implementujących dany interface, wtedy nasze IDE (lub kompilator jak piszemy w notatniku) wymusi na nas stworzenie odpowiednich metod. Jest to też używane w **POLIMORFIZMIE** oraz w wyrażeniach **LAMBDA**.

Dobra zrobmy coś więcej:

```
public interface CarImp {  
    void startEngine();  
}
```

Wszystkie metody zawarte w interfejsie **domyślnie są publiczne** więc w tym przypadku można ominąć słowo kluczowe **public**, nie jest potrzebne.

Teraz w naszej klasie która implementuje powyższy interface pojawi się błąd:

```
public class Car implements CarImp{
|
}

```

Class 'Car' must either be declared abstract or implement abstract method 'startEngine()' in 'CarImp'

Implement methods Alt+Shift+Enter More actions... Alt+Enter

Należy więc klasę Car zmienić (zaimplementować brakującą metodę):

```
public class Car implements CarImp{
    @Override //nadpisanie metody z interface
    public void startEngine() {
    }
}
```

I już możemy napisać co nasza metoda `startEngine()` ma robić.

Można powiedzieć że interface to taki szkielet klasy, nakładka na klasę wymuszająca stworzenie metod znajdujących się w interfejsie.

Ćwiczenia do wykonania:

Zróbmy sobie następujący szkielet danych i klas w aplikacji:

Program będzie miał możliwość zapisywania danych w pliku lub do bazy danych, stworzymy do tego interface:

```
public interface SaveDataImpl {
    boolean saveData(String data);
}
```

Czyli będziemy zapisywać dane typu **String**. (można oczywiście zrobić bardziej uniwersalną metodę ale to już jest zaawansowane programowanie)

Teraz tworzymy klasy:

```
public class SaveDataDatabase implements SaveDataImpl{
}
public class SaveDataFile implements SaveDataImpl{
}
```

Następnie implementujemy potrzebne metody:

```
public class SaveDataDatabase implements SaveDataImpl{
    @Override
    public boolean saveData(String data) {
        return false;
    }
}
public class SaveDataFile implements SaveDataImpl{
    @Override
    public boolean saveData(String data) {
        return false;
    }
}
```

Od teraz wszystkie programy będziemy pisali z wykorzystaniem INTERFEJSÓW!

3 zadania

1. Zrób kalkulator tak aby metoda oblicz znajdowała się w interface a klasa Suma, Różnica itp. implementowały ten interface.
2. Planety. Każda planeta może posiadać atmosferę bądź nie. Można sprawdzić czy na planecie jest życie. Stwórz odpowiednią nadklasę dla wszystkich planet. Stwórz interface posiadający jedną metodę sprawdzającą czy na planecie jest życie. Stwórz obiekt Ziemia dziedziczący po Planecie (nadklasie) oraz implementujący interface z metodą sprawdzającą czy na planecie jest życie.
3. Każde ciało może się poruszać lub być w spoczynku. Stwórzmy program z wykorzystaniem interfejsu, który będzie pozwalał na poruszenie, zatrzymanie i sprawdzenie czy obiekt się porusza czy też stoi w miejscu.

Program ma posiadać:

- Klasę Body która będzie implementowała interfejs BodyImp, oraz posiadała pole do przechowywania stanu obiektu (czy się porusza, czy stoi w miejscu).
- Interfejs BodyImp będzie posiadał szkielety metod do poruszania, zatrzymywania i sprawdzania czy ciało się porusza.