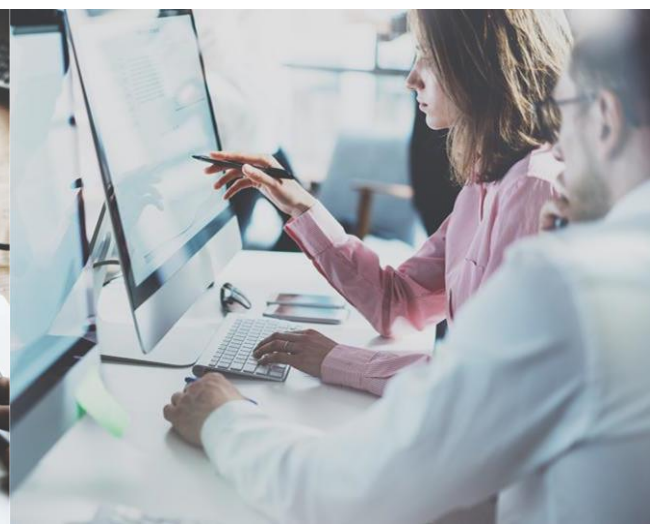




Szkolenia Komputerowe

**MICHAŁ KRUCZOWSKI
DZIEDZICZENIE**



Dziedziczenie



- ❑ Dziedziczenie to uzyskiwanie dostępu przez klasy dziedziczące do pól i metod klas nadrzędnych.

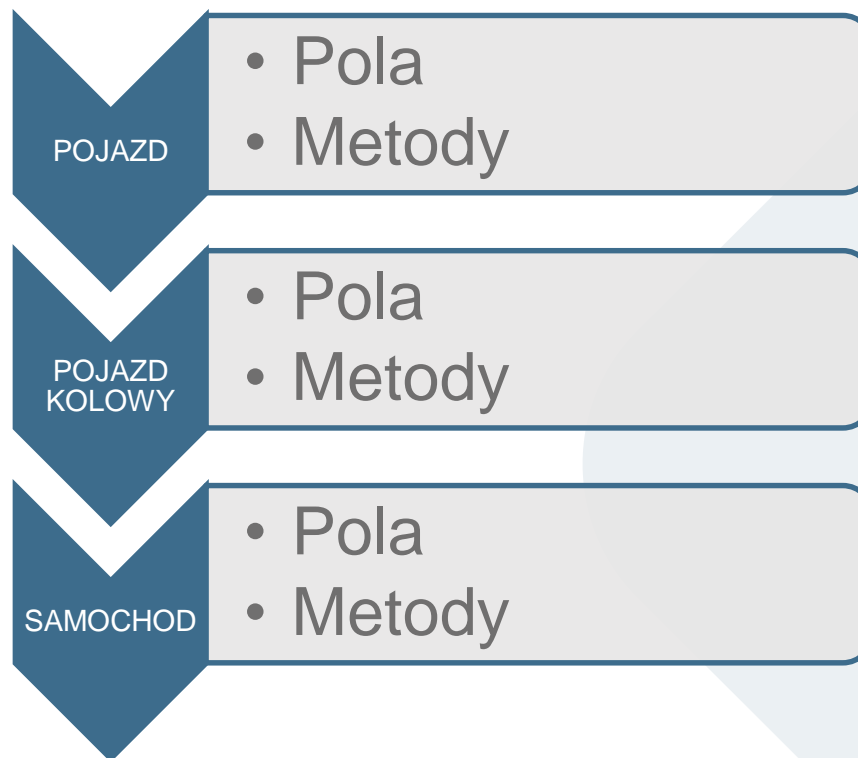
```
public class Pojazd{  
    ...  
}  
public class Samolot extends Pojazd{  
    ...  
}
```



Dziedziczenie



- ❑ Dziedziczenie to uzyskiwanie dostępu przez klasy dziedziczące do pól i metod klas nadrzędnych.



Mechanizmy powiązane z dziedziczeniem



- ☐ Dziedziczenie pól klasowych
- ☐ Dziedziczenie metod klasowych
- ☐ Wywoływanie metod klasy nadrzędnej (konstrukcja `super()`)
- ☐ Przysłanianie metod klasy nadrzędnej

- ☐ Dziedziczenie, a specyfikatory dostępu
- ☐ Wielodziedziczenie?!

- ☐ Polimorfizm
- ☐ Abstrakcja
- ☐ Enkapsulacja



Dziedziczenie pól klasowych



```
public class Vehicle {  
    public Engine engine;  
    public double fuel;  
}  
  
public class WheeledVehicle extends Vehicle {  
    public int wheels;  
}  
  
public class Car extends WheeledVehicle{  
  
}
```

Dziedziczenie metod klasowych



```
public class Vehicle {  
    public void startEngine() {  
        System.out.println("Started...");  
    }  
}  
  
public class WheeledVehicle extends Vehicle {  
    public void run() {  
        System.out.println("GO...")  
    }  
}  
  
public class Car extends WheeledVehicle{  
  
}
```

Przesłanianie metod klasy nadrzędnej



```
public class Vehicle {  
    public void startEngine() {  
        System.out.println("Started...");  
    }  
}  
  
public class WheeledVehicle extends Vehicle {  
    public void startEngine() {  
        System.out.println("GO...")  
    }  
}  
  
public class Car extends WheeledVehicle{  
  
}
```

Konstrukcja super() w metodzie i konstruktorze



- ❑ Wywołanie konstruktora klasy nadrzędnej

```
public class Vehicle {  
    String brand;  
    public Vehicle(String brand) {  
        this.brand = brand;  
    }  
}  
  
public class WheeledVehicle extends Vehicle {  
    public WheeledVehicle(String brand, model) {  
        super(brand);  
        this.model = model;  
    }  
}
```

- ❑ Wywołanie metody klasy nadrzędnej

```
super.MethodName([args]);
```


Dziedziczenie a specyfikatory dostępu

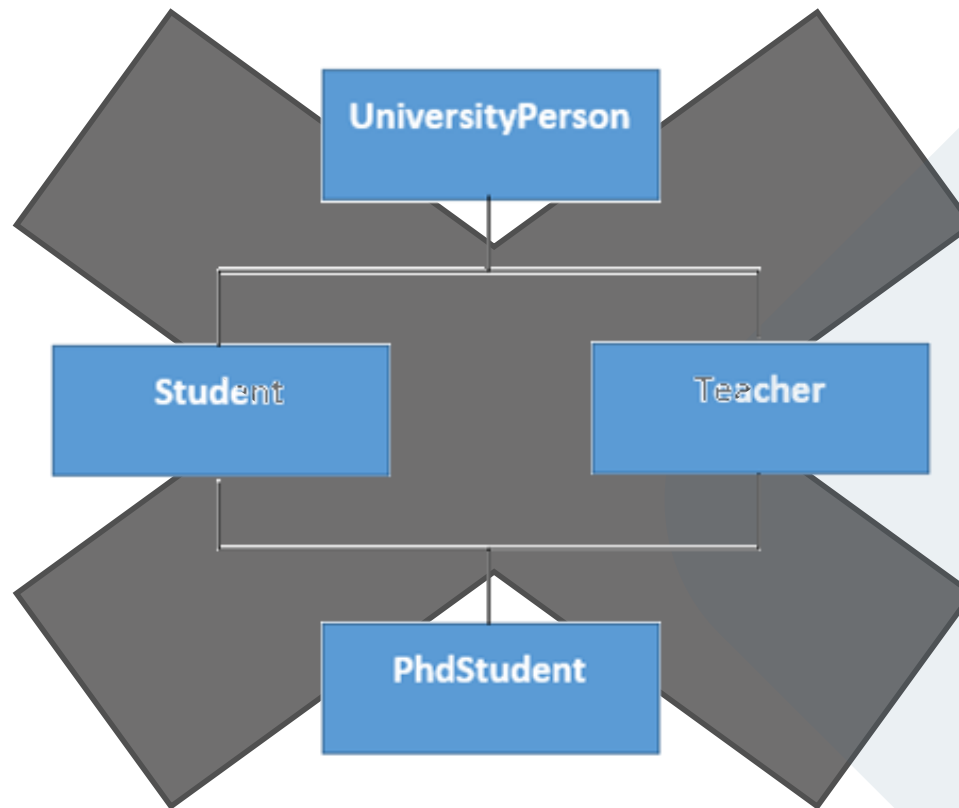


☐ Wszystkie składowe klasy mogą być dziedziczone zgodnie z założeniami specyfikującymi dostęp:

- | | |
|------------------------------------|------------------|
| <input type="checkbox"/> public | dostęp ogólny |
| <input type="checkbox"/> protected | dostęp pakietowy |
| <input type="checkbox"/> private | dostęp w klasie |



w Javie nie istnieje wielodziedziczenie



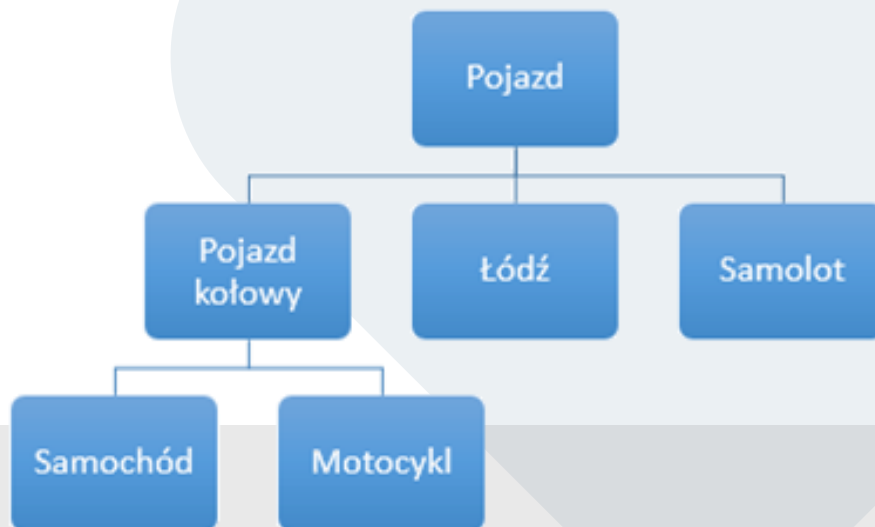


```
Vehicle vehicle = new Vehicle("BMW");
```

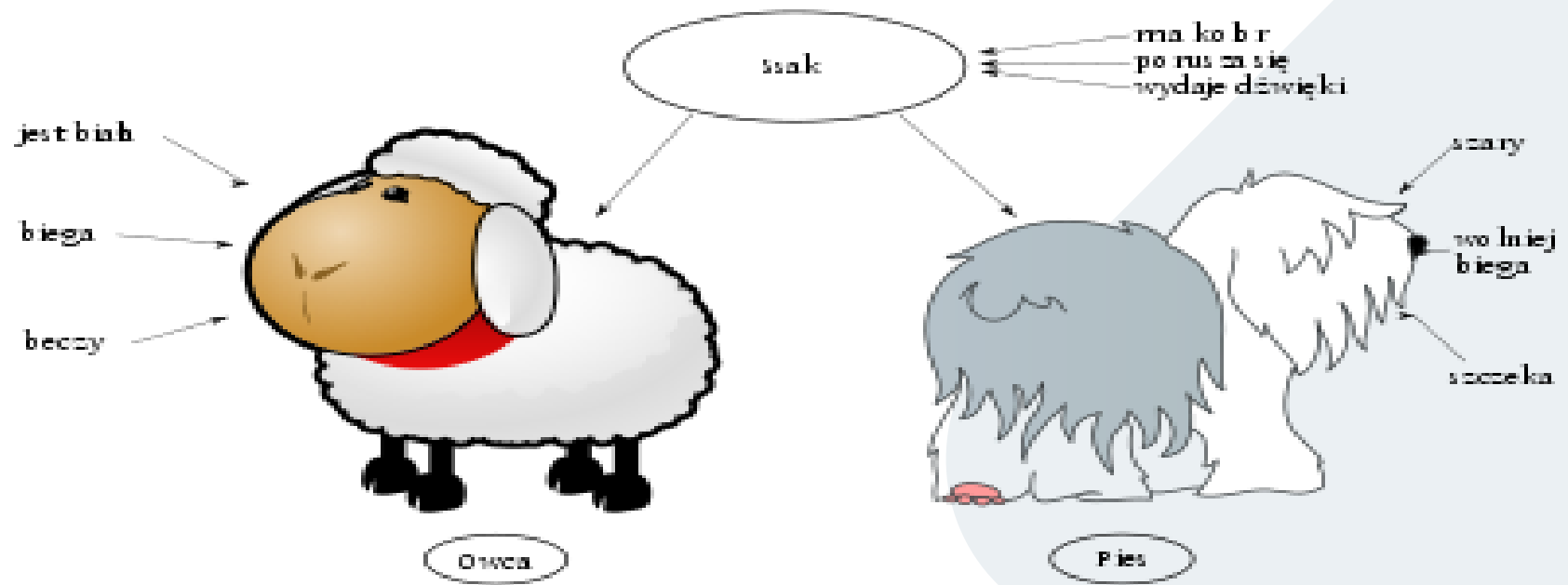
```
WheeledVehicle wVehicle = new WheeledVehicle("BMW", "X5");
```

POLIMORFIZM – wielopostaciowość

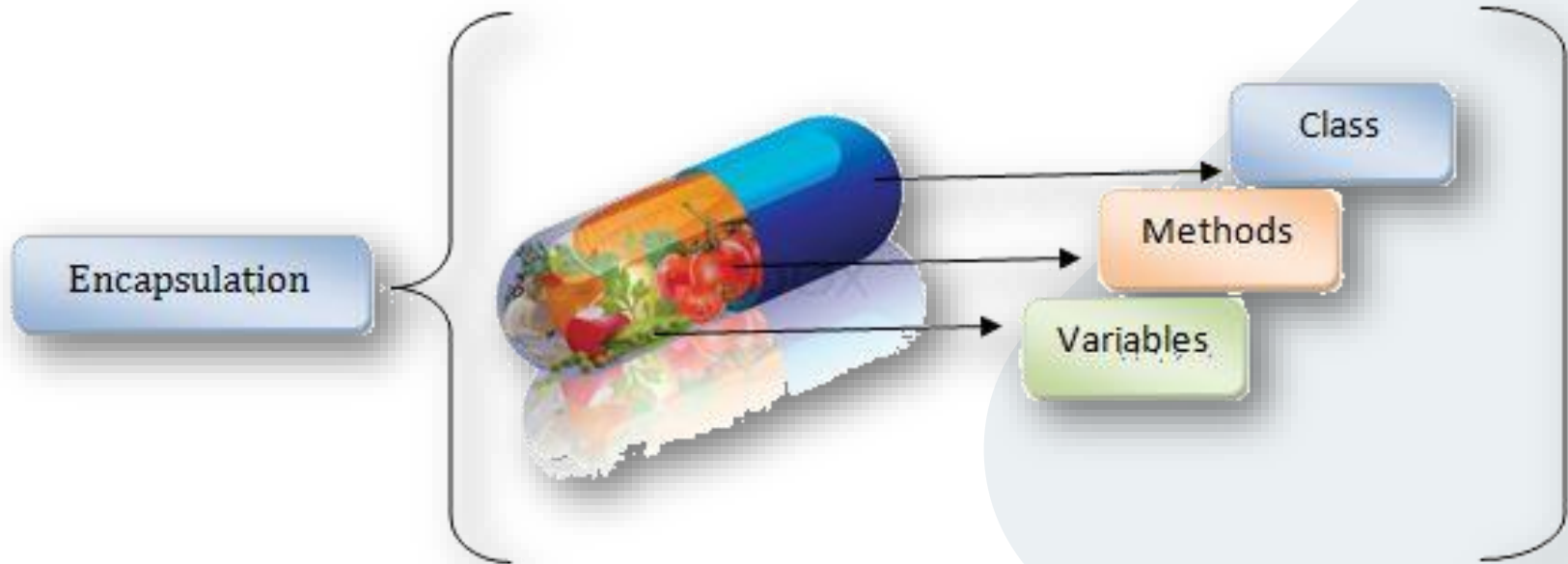
```
Vehicle wVehicle = new WheeledVehicle("BMW", "X5");
```



Abstrakcyjne spojrzenie na OOP



Abstrakcyjne spojrzenie na OOP





DZIĘKUJĘ ZA UWAGĘ



WYKONAWCASZKOLENIA:

software
development
academy



Szkolenia
Komputerowe



BIURO PROJEKTU

Toruńska Agencja Rozwoju Regionalnego S.A.

ul. Włocławska 167, 87-100 Toruń

56 699 54 89

szkoleniakomputerowe@tarr.org.pl

www.szkoleniakomputerowe.tarr.org.pl

Projekt „CERTYFIKOWANE SZKOLENIA KOMPUTEROWE dla osób dorosłych z województwa kujawsko-pomorskiego”
współfinansowany ze środków Europejskiego Funduszu Społecznego w ramach Regionalnego Programu Operacyjnego
Województwa Kujawsko-Pomorskiego na lata 2014 – 2020