

Interfejsy

<https://github.com/idzikpro/JavaCore/tree/master/src/main/java/pl/idzikpro/interfaces>

Interfejs to zestaw metod bez ich implementacji. Właściwa implementacja metod danego interfejsu znajduje się w klasie implementującej dany interfejs. Od Java 1.8 możemy jednak definiować domyślne metody interfejsu z ciałami. Dajemy wtedy przed nazwą metody słowo **default**. Po co to robić? W przypadku, gdy metoda często się powtarza w klasach implementujących ten interfejs.

Klasa może implementować wiele interfejsów ale dziedziczyć tylko z jednej klasy. W Javie występuje bowiem dziedziczenie jednokrotne. Gdy implementujemy interfejs, to musimy zaimplementować jego wszystkie metody. Metody w interfejsie muszą być **public abstract** i nie można tego zmieniać. Interfejsy mogą być również używane jako pola składowe klasy.

Domyślnie wszystkie składowe(tzw zmienne) interfejsu są stałymi publicznymi przypisanymi do interfejsu więc słowa kluczowe **public static final** mogą zostać pominięte.

```
public interface Voice {  
    public void saySth();  
}
```

```
public class Cat implements Voice {  
    @Override  
    public void saySth() {  
        System.out.println("Miau");  
    }  
}
```

```
List<Voice> voiceList = Arrays.asList(  
    new Cat(),  
    new Dog()  
);  
  
for (Voice animal : voiceList  
) {  
    animal.saySth();  
}
```

Dziedziczenie

Interfejsy mogą dziedziczyć po innych interfejsach. Taki interfejs musi zawierać wszystkie metody z tych odziedziczonych interfejsów.

Interfejs znacznikowy

Interfejs znacznikowy to interfejs, który nie ma żadnej metody. Służy on do oznaczenia kompilatorowi danej klasy, która implementuje taki interfejs. Przykładem takiego interfejsu jest interfejs **Serializable**.

Zalety stosowania

- ułatwiają integrację różnych fragmentów kodu.
- kod staje się łatwiejszy do modyfikacji. Interfejsy oddzielają komponenty programu i dlatego . elementy programu można z łatwością wymieniać.
- można porównać je do pilota do TV. Korzystając z TV przy użyciu pilota zauważymy jego możliwości.