

# **JAVA ZAAWANSOWANA: NOWOŚCI W JAVIE 8 I 9**

# PROGRAMOWANIE FUNKCYJNE

# PARADYGMAT PROGRAMOWANIA FUNKCYJNEGO

Paradygmat programowania funkcyjnego, w odróżnieniu od paradygmatu obiektowego określa, że najważniejszym i jedynym narzędziem do realizacji programu są funkcje. Funkcje mogą przyjmować na wejściu także funkcje oraz zwracać funkcje jako wynik.

# INTERFEJSY FUNKCYJNE

Są to interfejsy, które posiadają tylko i wyłącznie jedną metodę abstrakcyjną (interfejsy SAM), np. interfejs `Runnable`, `Callable`.

```
@FunctionalInterface
public interface Presenter {
    void show(String text);
}
```

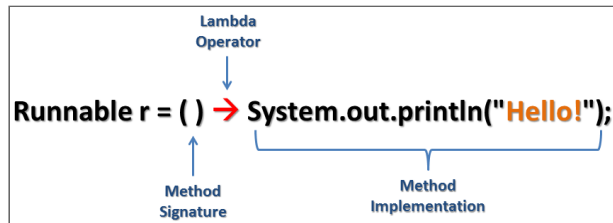
## DEFAULT

Metoda domyślna jest to metoda interfejsu, która zawiera domyślną implementację. Dzięki tej nowej składni, w ramach interfejsu funkcyjnego możemy deklarować więcej niż jedną metodę, z zachowaniem zasady pojedynczej metody abstrakcyjnej.

```
public interface Compare {  
    int max(int[] values);  
  
    default int min(int a, int b) {  
        return a < b ? a : b;  
    }  
}
```

# WYRAŻENIA LAMBDA

Pozwalają traktować klasę anonimową jak zwykłą funkcję, znacząco skracając przy tym składnię samego zapisu.

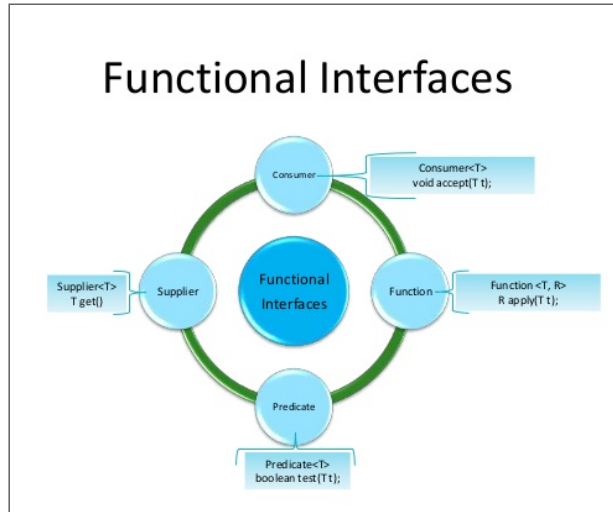


# INTERFEJSY FUNKCYJNE W JAVIE 8

Java 8 wprowadza wiele przydatnych interfejsów funkcyjnych, które można wykorzystywać m.in. w Stream API. Wszystkie interfejsy znajdują się w pakiecie `java.util.function`. Są to między innymi:

- `Supplier`
- `Function<T, R>`
- `Consumer`
- `UnaryOperator<T, T>`
- `Predicate`

# INTERFEJSY FUNKCYJNE W JAVIE 8





## OPTIONAL

Obiekt kontener, który może zawierać wartość inną niż null lub nie zawierać obiektu wcale. Jeśli wartość jest obecna, metoda `isPresent()` zwraca `true`. Jeśli nie ma żadnej wartości, obiekt jest traktowany jako pusty, a `isPresent()` zwraca `false`. Zapewnione są dodatkowe metody, które zależą od obecności lub braku zawartej wartości, takie jak `orElse()` (zwraca wartość domyślną, jeśli nie ma żadnej wartości) i `ifPresent()` (wykonuje akcję, jeśli wartość jest obecna).

```
Movie movie = null;
Optional<Movie> movieContainer =
Optional.ofNullable(movie);

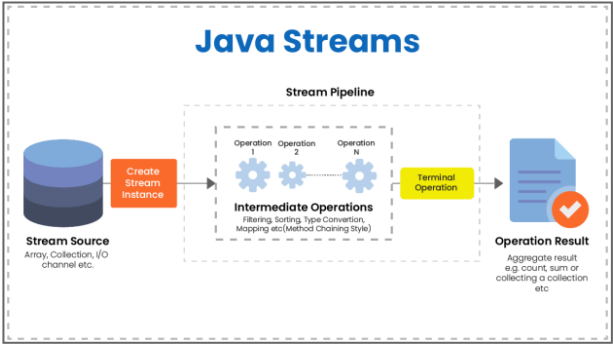
movieContainer.ifPresent(System.out::println)
;
```

# STREAM API

Mechanizm przetwarzania źródła danych (kolekcji, tablic, plików) w Javie:

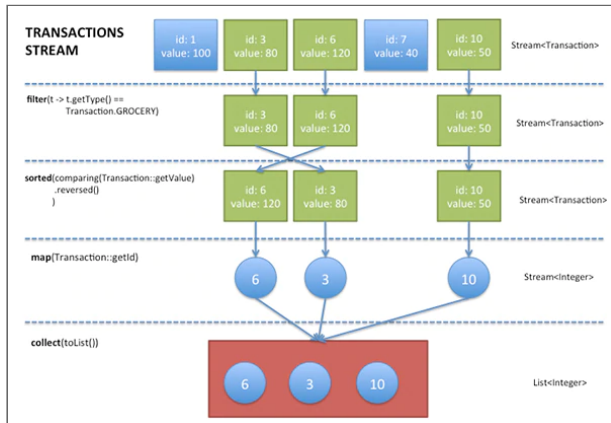
- reprezentuje sekwencyjny zestaw elementów i pozwala na wykonywanie różnych operacji na tych elementach
- strumienie umożliwiają w łatwy sposób zrównoleglanie operacji, przetwarzanie danych może być dużo szybsze

# STREAM API



# STREAM API - PRZETWARZANIE DANYCH

Przetwarzanie danych w Stream API realizowane jest za pomocą metod pośrednich, które w dowolny sposób mogą wpływać na strumień danych. Metody końcowe mogą zwracać rezultat przetwarzania danych.



# JAVA 9 MODUŁY

# MODUŁY

Zostały wprowadzone w wersji Javy 9. To grupa ściśle powiązanych pakietów i zasobów z nowym plikiem deskryptora modułu. Można powiedzieć, że jest to abstrakcja „pakietu pakietów Java”.

