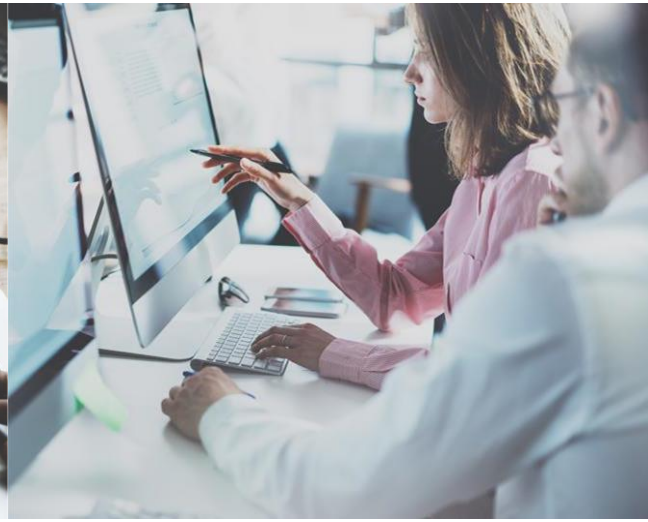
Szkolenia
Komputerowe

MICHAŁ KRUCZOWSKI
WYRAŻENIA LAMBDA & STREAM API





Fundusze Europejskie Program Regionalny
WOJEWÓDZTWO KUJAWSKO-POMORSKIE
Toruńska Agencja Rozwoju Regionalnego S.A.
Unia Europejska Europejski Fundusz Społeczny

Czym są wyrażenia lambda (*Lambda Expressions*)?

- Wyrażenie lambda jest pseudo-metodą którą można:
  - przypisać do zmiennej
  - wywołać
  - przekazać jako argument do innej metody.

- Składnia:

```
<lista parametrów> -> <ciało wyrażenia>
```

- Zastosowanie:
  - Operacje na kolekcjach.
  - Praca ze strumieniami.
  - Programowanie funkcyjne

Strumień (*STREAM API*)

- Strumień to sekwencja elementów na której możliwe jest wykonanie wielu operacji jednocześnie
- Strumienie są tworzone na podstawie źródła:
  - Tablica
  - Lista
  - Zbiór
  - itp.
- Operacje wykonywane na strumieniach:
  - Pośrednie - przetwarzające elementy strumienia
  - Kończące - zwracające wyniki

- Operacje strumieniowe mogą być wykonywane sekwencyjnie bądź równolegle

Operacje pośrednie

- .map() - przekształca każdy element strumienia za pomocą przekazanej funkcji
- .filter() - ogranicza zawartość strumienia do obiektów spełniających predykat
- .sorted() - przekształca strumień do postaci posortowanej
- .limit() - ogranicza strumień do podanego rozmiaru
- .distinct() - ogranicza strumień do unikalnych elementów

Operacje kończące

- .forEach(Consumer<T>) - wykonuje akcję z każdym elementem strumienia, zamykając go jednocześnie
- .collect() - zbiera wszystkie elementy pozostające w strumieniu i zwraca je w postaci determinowanej przez podany Collector
- .allMatch(), .anyMach(), noneMatch() - sprawdza czy w strumieniu obiekty które spełniają predykat
- .findFirst(), .findAny() - zwraca pierwszy/ dowolny element strumienia
- .count() - zlicza elementy strumienia
- .toArray() – zwraca elementy strumienia w postaci tablicy

Tworzenie strumienia

- Sortowanie sekwencyjne

```
collection.stream()
        .operacjePośrednie()
        .operacjaKończąca()
```

- Sortowanie równoległe

```
collection.parallelStream()
        .operacjePośrednie()
        .operacjaKończąca();
```

Metoda sorted()

- Rosnąco

```
array.stream()
      .sorted((o1, o2) -> o1.compareTo(o2))
      .forEach(System.out::print);
```

- Malejąco

```
array.stream()
      .sorted((o1, o2) -> o2.compareTo(o1))
      .forEach(System.out::print);
```

Klasa Comparator

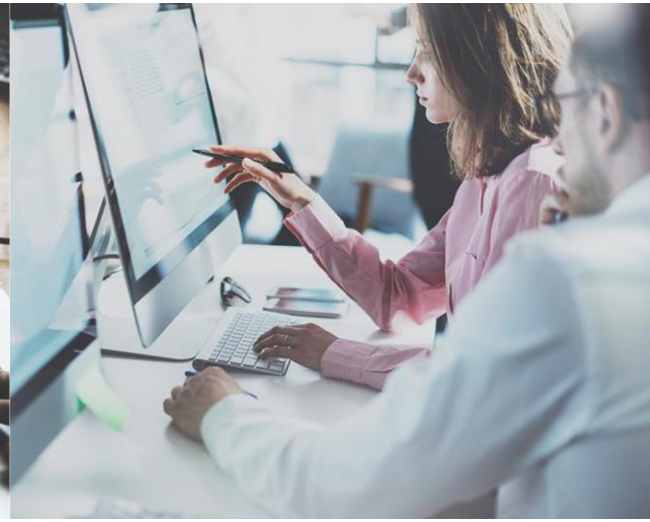
```
List<ModelClass> orderedList = list.stream()
                .sorted(
                        Comparator.comparing(
                        ModelClass::getField
                        ).reversed())
                 .collect(Collectors.toList());

orderedComments.forEach(System.out::print);
```

Szkolenia Komputerowe

MICHAŁ KRUCZOWSKI
OPTIONAL





Fundusze Europejskie Program Regionalny
WOJEWÓDZTWO KUJAWSKO-POMORSKIE
Toruńska Agencja Rozwoju Regionalnego S.A.
Unia Europejska Europejski Fundusz Społeczny

Czym jest Optional?

- Optional to kontener pomagający uniknąć wyjątku *NullPointerException*

- Pole typu Optional<Class>:
  - Przechowuje obiekty klasy Class
  - Może zawierać wartość
  - Może zawierać null

- Inicjalizacja

```
Optional<Class> optNotNull = Optional.of(notNull);
Optional<Class> optMayBeNull = Optional.ofNullable(mayBeNull);
Optional<Class> optEmpty = Optional.empty();
```

Optional na przykładach

```
Optional<String> optional = Optional.of(”XYZ");
optional.isPresent();                                   // true
optional.get();                                         // XYZ
optional.orElse("fallback");                            // XYZ
optional.ifPresent(s -> System.out.println(s));         // X



Product product = null;
Optional<Product> optional = Optional.ofNullable(product);
optional.isPresent();                                   // false
Optional
        .ifPresent(product -> System.out.println(product.getName()))
        .orElse(”BRAK");                                // BRAK
```

DZIĘKUJĘ ZA UWAGĘ

WYKONAWCA SZKOLENIA:

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

Fundusze Europejskie Program Regionalny

WOJEWÓDZTWO KUJAWSKO-POMORSKIE

Toruńska Agencja Rozwoju Regionalnego S.A.

Unia Europejska Europejski Fundusz Społeczny