

# Wyższa Szkoła Informatyki Stosowanej i Zarządzania

pod auspicjami Polskiej Akademii Nauk

## WYDZIAŁ INFORMATYKI

Kierunek INFORMATYKA

Studia I stopnia (dyplom inżyniera)

---



## Język Java – laboratorium 6

dr inż. Łukasz Sosnowski  
[lukasz.sosnowski@wit.edu.pl](mailto:lukasz.sosnowski@wit.edu.pl)  
[sosnowsl@ibspan.waw.pl](mailto:sosnowsl@ibspan.waw.pl)  
[l.sosnowski@dituel.pl](mailto:l.sosnowski@dituel.pl)

[www.lsosnowski.pl](http://www.lsosnowski.pl)



## **Część 1 – Rozliczenie pracy domowej**



# **Prezentacja 1**

## **org.apache.commons.lang3.StringUtils – przegląd metod, przykłady użycia**



## **Prezentacja 2**

# **org.apache.commons.lang3.math.NumberUtils– przegląd metod, przykłady użycia**



## **Pakiet: pl.wit.lab4**

### **Omówienie rozwiązania**

- Realizacja Przewodnik9.pdf
- Plik tutorial9.txt



## **Pakiet: pl.wit.lab5**

### **Omówienie rozwiązania**

- Realizacja Przewodnik10.pdf
- Pliki ilustracyjne:
- LambdaTutorial.java
- LambdaTutorialTest.java



## Pakiet dla dzisiejszego laboratorium to:

```
package pl.wit.lab5.p2;  
package pl.wit.lab5.p3;  
package pl.wit.lab5.p4;  
package pl.wit.lab6;
```

dla klas biznesowych oraz testów jednostkowych  
jednak w odpowiednich podkatalogach katalogu *src*:  
*main* i *test* odpowiednio,



## **Część 2 – Obsługa wejścia i wyjścia**





**Pakiet:** `package pl.wit.lab4;`, **Katalog UBI: 1**

## **Przykłady operacji wejścia wyjścia na plikach z wykładu**

Pliki ilustracyjne:

- `InputOutputTest.java`
- `plik1.txt`
- `data1.txt`



## **Część 3 – strumienie**

### **Przykłady z wykładu**



## Tworzenie strumieni – przypomnienie z wykładu

- Strumień może być utworzony z kolekcji - za pomocą metody *stream()*.
- Strumień może być utworzony z tablicy - z użyciem klasy *Arrays* posiadającą metodę *stream()*.
- Strumień może być utworzony z pliku – metoda *lines()* z klasy *FileReader*
- Strumienie dla typów prostych mogą tworzone być za pomocą odpowiednich klas *IntStream*, *LongStream* i *DoubleStream* oraz odpowiednich metod typu: *of(wartosci)*, *generate()* lub dla strumieni całkowitoliczbowych *range(start, end)*.
- Strumień może powstać w wyniku wykonania wyrażenia regularnego oraz metody *splitAsStream()*.
- Pusty strumień – metoda *empty()*.



## Operacje na strumieniach zwracające nowy strumień

- Filtrowanie z użyciem metody *filter(predykat)*. Metoda zwraca nowy strumień zawierający tylko te elementy, które spełniają predykat.
- Odwzorowywanie – zmiana typu elementu strumienia poprzez utworzenie nowego typu elementu na bazie przetwarzanych elementów. Typowa metoda to *map()*, ale dostępne są również metody dla typów prostych: *mapToInt()*, *mapToLong()*, *mapToDouble()* lub *mapToObj()* dla mapowania na obiekty.
- Limitowanie z użyciem metody *limit(rozmiar)* - zwraca strumień ograniczony do zadanej liczby elementów.
- Wykonywanie akcji na elemencie poprzez metodę *peek()* pozwalającą wykonać dodatkową operację z użyciem elementu (bez jego modyfikacji) oraz zwrócić strumień o tych samych elementach.



## Operacje kończące – przypomnienie z wykładu

- Redukcja – zwrócenie wartości typu innego niż strumień, obliczonej na podstawie elementów strumienia. Do wykonania redukcji została przygotowana metoda *reduce(identityVal, BinaryOperator<T> accumulator)*.
- Zliczanie – z użyciem metody *count()*.
- Element największy – z użyciem metody *max()*.
- Element najmniejszy – z użyciem metody *min()*.
- Tworzenie kolekcji – z użyciem metody *collect()* oraz Klasy *Collectors* zawierającej m. in metody: *toList()*, *toSet()*.
- Wykonanie akcji kończącej na każdym elemencie poprzez wywołanie metody *forEach()*
- Tworzenie tablicy – z użyciem metody *toArray()*.
- *Inne: findFirst, findAny, noneMatch, allMatch, anyMatch(), itd.*



**Pakiet:** `package pl.wit.lab5.p3;` **Katalog UBI: 2**

## **API strumieni – przykłady z wykładu**

Pliki ilustracyjne:

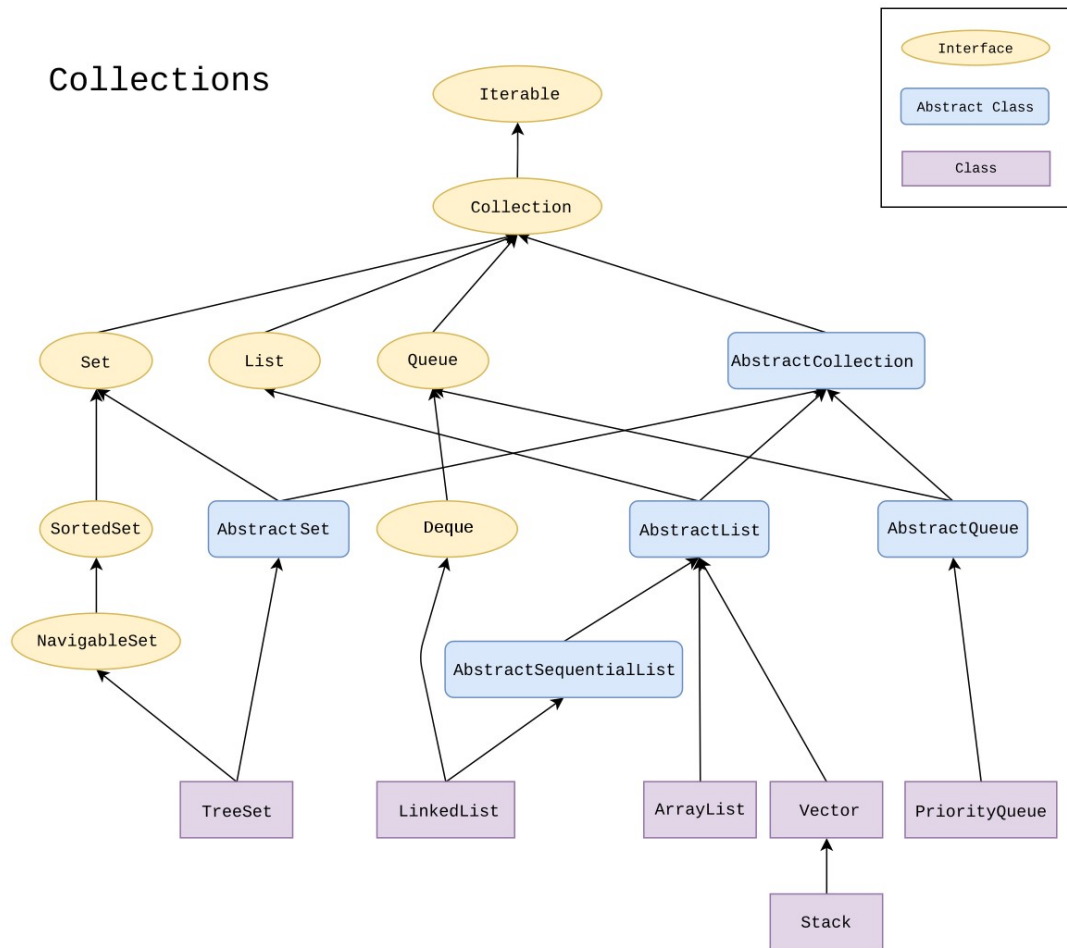
- StreamTest.java



## Część 4 – klasy kolekcji



# Hierarchia



\* - źródło w [https://en.wikipedia.org/wiki/Java\\_collections\\_framework](https://en.wikipedia.org/wiki/Java_collections_framework)





## Iterowanie kolekcji – przypomnienie z wykładu

- Dla przypadku braku modyfikacji kolekcji stosujemy pętlę typu for-each, aby przejść przez wszystkie elementy kolekcji.
- Pętla rozszerzona w postaci:  

```
for(TypElementuKolekcji elem:ObiektKolekcji){  
    // przetwarzanie  
}
```
- W przypadku konieczności dokonywania modyfikacji typu usunięcie lub wstawienie (dla listy) stosujemy Iterator lub ListIterator:  

```
Iterator<TypElementuKolekcji> it = col.iterator();  
while(it.hasNext()){  
    it.next();}
```



**Pakiet:** `package pl.wit.lab6;`, **Katalog UBI: 3**

## **API strumieni – przykłady z wykładu**

Pliki ilustracyjne:

- `MyCollections.java`
- `MyCollectionsTest.java`



**Pakiet:** `package pl.wit.lab5.p4;`, **Katalog UBI: 4**

**Zadanie 1 domowe dla wszystkich:**

**Przewodnik 11 – klasy i interfejsy kolekcji**

Pliki do zadania:

- CollectionsTutorial.java
- CollectionsTutorialTest.java



## **Część 5 – programowanie wielowątkowe**

### **Wybrane przykłady z wykładu**



**Pakiet:** `package p1.wit.lab5.p2;;` **Katalog UBI: 5**

**Programowanie wielowątkowe – przykłady z wykładu.**

**Podstawowe aspekty wielowątkowości:**

Pliki ilustracyjne:

- FirstThread.java
- SimpleThread.java,
- SimpleThread2.java
- Threads1Test.java



**Pakiet:** `package p1.wit.lab5.p2;`, **Katalog UBI: 6**

**Programowanie wielowątkowe – przykłady z wykładu.**

**Podstawowe własności i metody**

Pliki ilustracyjne:

- MultiplyArr.java
- SecondThread.java
- ThirdThread.java,
- Goods.java, Producer.java, Consumer.java
- Threads2Test.java



**Pakiet:** `package p1.wit.lab5.p2;`, **Katalog UBI: 7**

**Programowanie wielowątkowe – przykłady z wykładu.**

**Zaawansowane techniki wielowątkowości.**

Pliki ilustracyjne:

- `ArrayIntSum.java`
- `CountDownLatchThread.java`
- `Threads3Test.java`



**Pakiet:** `package p1.wit.lab5.p3;`, **Katalog UBI: 8**

## **Zadanie 2 domowe dla wszystkich:**

**Przewodnik 12 – programowanie wielowątkowe, licznik dla plików w katalogu**

Pliki do zadania:

- CounterMultiThread.java
- CounterMultiThreadTest.java





## **Część 6 – Praca domowa**



## Praca domowa:

- Przewodnik nr 11
- Przewodnik nr 12

## Prezentacja na temat:

- java.time API cz.1 – omówienie klas **LocalDate**, **LocalTime**, **LocalDateTime** z przykładami użycia
- java.time API cz.2 - omówienie klas **ZoneDateTime**, **Period**, **Duration** z przykładami użycia

# Wyższa Szkoła Informatyki Stosowanej i Zarządzania

pod auspicjami Polskiej Akademii Nauk

## WYDZIAŁ INFORMATYKI

Kierunek INFORMATYKA

Studia I stopnia (dyplom inżyniera)

---



**Dziękuję za uwagę!**