ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Java alapú webfejlesztés

Gyűjtemények

Stream API

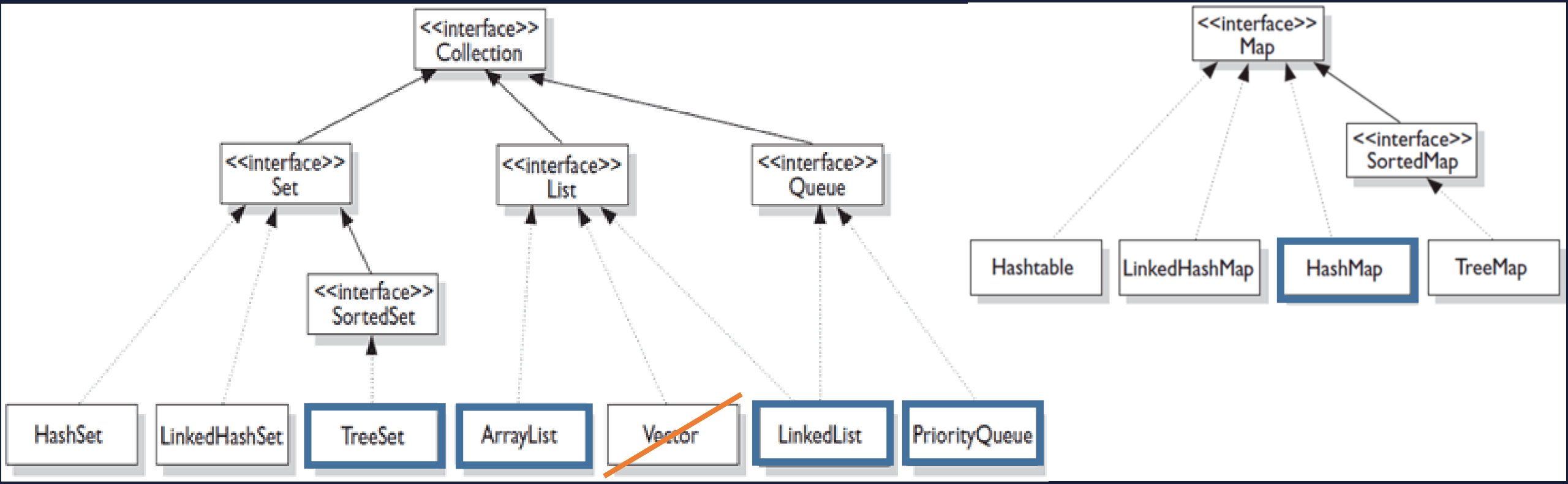
Kivételkezelés

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Java Collections Framework

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Collections Framework



Forrás: https://www.w3resource.com/java-tutorial/java-collections.php

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Collection<E>

Általános gyűjtemény metódusok

- add, remove, clear
- isEmpty, size
- contains
- stream
- toArray

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

List<E>

- Változó méretű lista, különböző implementációk
  - ArrayList: tömb alapú, LinkedList: láncolt lista
  - Vigyázat! Az Arrays.asList() fix méretű listát készít!
- Metódusai: Collection metódusok +
  - get, set
  - indexOf, lastIndexOf

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Queue<E>

- Sor: egy irányban vagy mindkét irányban feldolgozható
  - utóbbi a Deque<E>: double ended queue

	Sikertelenség esetén kivételt dob	**Sikertelenség esetén speciális értéket ad vissza**
Beszúrás	add(e)	offer(e)
Kiolvasás és Törlés	remove()	poll()
Kiolvasás	element()	peek()

Set<E>

- Halmaz: ugyanaz az elem nem szerepelhet többször
  - azaz nem lehet olyan e1 és e2, hogy e1.equals(e2), illetve e1.compareTo(e2)==0
- Metódusai: Collection metódusok (a halmazműveletek is)
  - Unió: s1.addAll(s2)
  - Metszet: s1.retainAll(s2)
  - Különbség: s1.removeAll(s2)

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

TreeSet<E> osztály

- Faként implementált **rendezett** halmaz, garantálja a log(n) költségű beszúrást/törlést
- Metódusai: Set metódusok +
  - first/last
  - lower/higher
  - headSet/tailSet

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Map<K, V>

- Kulcs-érték párok gyűjteménye
- NEM a Collection leszármazottja, de kínál Collection típusú hozzáférést a tartalmához
  - entrySet; keySet, values
- További hasznos metódusok
  - put, putIfAbsent
  - get, getOrDefault
  - containsKey, containsValue

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Stream API

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Stream API

- Szekvenciák pipeline elvű feldolgozása (mint a Linq)
  - a közbülső fázisokat nem tároljuk
- Stream<T> interfész: objektum stream
  - collection-ből a .stream() metódusával kaphatjuk
  - 3 primitív típusból is lehetséges stream-et készíteni (IntStream, LongStream, DoubleStream interfészek)

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Stream API

- Köztes műveletek
  - visszatérési értékük stream (de lehetséges, hogy más típusú elemekből álló stream!)
  - egymás után láncolhatók
- Lezáró (terminális) műveletek
  - nem streamet ad vissza, hanem primitív értéket vagy gyűjteményt vagy akár semmit (forEach)
  - egy pipeline egy lezáró műveletet tartalmazhat, a végén
  - lezárás után a stream-re újabb köztes művelet nem hívható

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Stream API: filter

Szűrés (Linq: Where)

- lambda kifejezés formájában adhatjuk meg a feltételt
  - a lambda operátor itt `->`
- példa:

```
people.stream().filter(p->p.getAge()>40)
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Stream API: map

Leképezés (Linq: Select)

- lambda kifejezés formájában adhatjuk meg az elemek átalakításának módját
- példa:

```
people.stream().map(p->p.getName().toUpperCase())
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Stream API: sorted

Rendezés (Linq: OrderBy)

- lambda kifejezés helyett Comparator.comparing-et is használhatunk
- példa:

```
people.stream()
      .sorted(Comparator.comparing(Person::getAge)
            .reversed())
```

metódus referencia

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Stream API: collect

Redukció (nincs egy konkrét Linq megfelelője)

- Lezáró művelet
- Paramétere: számos lehetőség a Collectors osztályban
  - Collectors.toList()
  - Collectors.joining(", ")
  - Collectors.summingInt(Person::getAge)
  - Collectors.groupingBy(p->p.getAge())

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Stream API

További gyakran használt lezáró műveletek:

- forEach
- count
- findFirst, findAny
  - Optional típust ad vissza
- min, max
  - általános stream-nél Comparator paraméterrel
- sum, average, min, max
  - szám streameknél paraméter nélkül, többnyire optional eredmény

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Kivételkezelés

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Kivételkezelés

- Nagyrészt ugyanúgy működik, mint C#-ban
  - try, catch, finally, throw
  - a C# -os using blokk megfelelője: try-with-resources

- Fontos különbség: **kötelező kivételkezelés**

Kivételkezelés

- Ellenőrzött (kötelezően kezelendő) kivételek
  - Az Exception osztály leszármazottai
  - Kötelező valahol a programban elkapni:
    - vagy elkapjuk a meghívás helyén
    - vagy a metódus deklarációban jelezni kell, hogy ez a metódus ilyen kivételt dobhat

```
public void writeFooFile() throws IOException {....}
```

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Kivételkezelés

- Nem ellenőrzött kivételek
  - A RuntimeException osztály leszármazottai
  - Nem kötelező őket elkapni, de el lehet
  - Nem ajánlott saját kivételosztályokhoz

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Források

- Dr. Szénási Sándor előadás prezentációja és példái
- Collections Tutorial https://docs.oracle.com/javase/tutorial/collections/index.html
- Package java.util.stream https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html
- The Java 8 Stream API Tutorial https://www.baeldung.com/java-8-streams