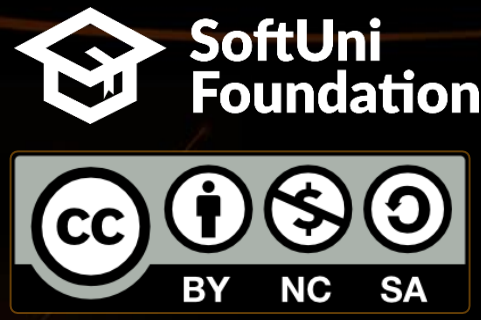
Sets, Maps

Java Collections API - Sets, Maps



SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Contents

Have a Question?

SoftUni Foundation

sli.do

#java-fund

Sets

HashSet<E>, TreeSet<E> and LinkedHashSet<E>

Sets in Java

SoftUni Foundation

- A set keeps unique elements
  - Provides methods for adding/removing/searching elements
  - Offers very fast performance
- `HashSet<E>`
  - The elements are randomly ordered
- `TreeSet<E>`
  - The elements are ordered incrementally
- `LinkedHashSet<E>`
  - The order of appearance is preserved

Sets Methods

- Initialization

```
HashSet<String> hash = new HashSet<String>();
```

  - For easy reading you can use diamond inference syntax

```
TreeSet<String> tree = new TreeSet<>();
```

- .size()
- .isEmpty()

```
HashSet<String> hash = new HashSet<>();
System.out.println(hash.size()); // 0
System.out.println(hash.isEmpty()); // True
```

HashSet<E> – add()

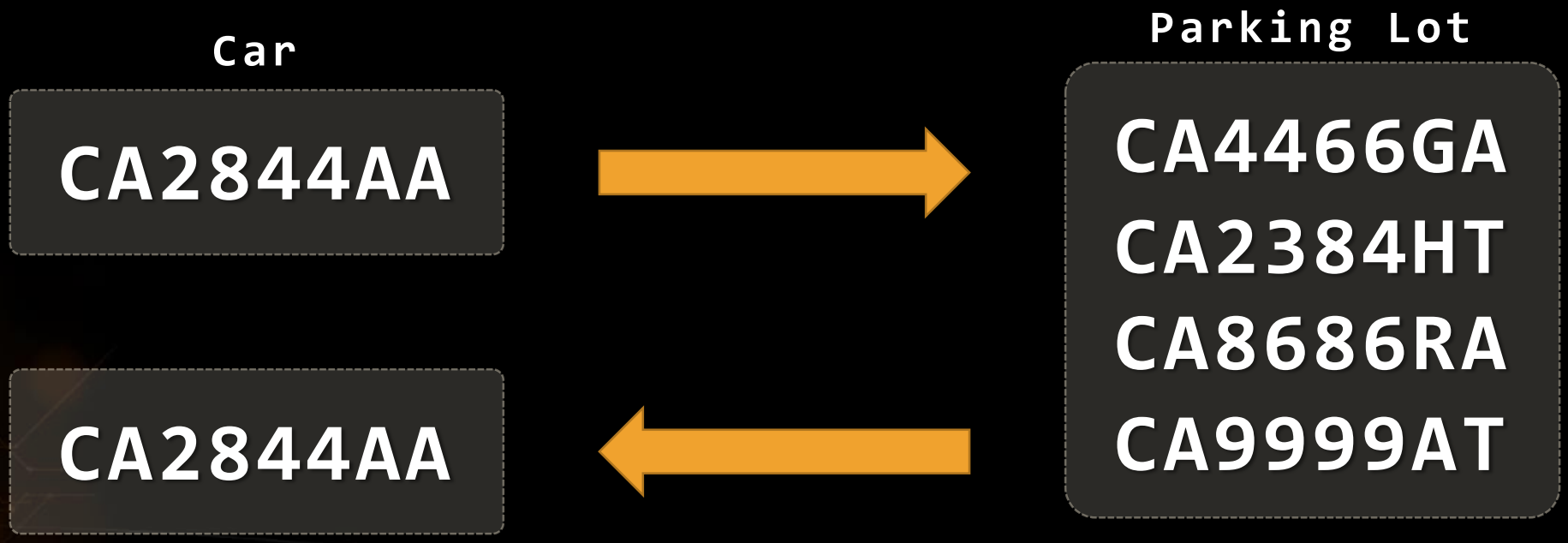
HashSet<E> – remove()

TreeSet<E> – add()

LinkedHashSet<E> – add()

Problem: Parking Lot

- Write a program that:
  - Record car number for every car that enter in parking lot
  - Remove car number when the car go out



Solution: Parking Lot

SoftUni Foundation

```
HashSet<String> parkingLot = new HashSet<String>();
while(true)
  String input = sc.nextLine();
  if (input.equals("END"))
    break;
  else
    String[] reminder = input.split(", ");
    if (reminder[0].equals("IN"))
      parkingLot.add(reminder[1]);
    else
      parkingLot.remove(reminder[1]);
```

Problem: SoftUni party

- Guests are two types:
  - Regular
  - VIPs – their tickets start with digit
- Until PARTY command, you will receive guest invitations
- Next until END command, you will receive a second list with guests that actually come to the party
- Find how many guests didn't came to the party
- Print all guests that didn’t came (VIPs first)

```
Reservation List
7IK9Yo0h
9NoBUajQ
Ce8vwPmE
SVQXQCbc
```

Solution: SoftUni party

SoftUni Foundation

```
HashSet<String> vip = new HashSet<String>();
TreeSet<String> regular = new TreeSet<String>();
while (true)
  String input = sc.nextLine();
  if (input.equals("PARTY"))
    break;
  else
    String sign = Character.toString(input.charAt(0));
    if (numbers.contains(sign))
      vip.add(input);
    else
      regular.add(input);
//TODO: Remove from guest, that came to party
regular.addAll(vip);
//TODO: Print results
```

Return true or false

Problem: "Voina" – Number Game

SoftUni Foundation

- "Voina" is similar to card game, but with numbers
- There are two players. Each one have 20 numbers (read from console, separated with single space)
- Each player can have only unique numbers
- "Voina" is round game, so every round each player bet his first number from deck.
- Player with bigger number win and place both numbers at the bottom of his deck
- Game and after 50 rounds or when any player have 0 numbers

Solution: SoftUni party

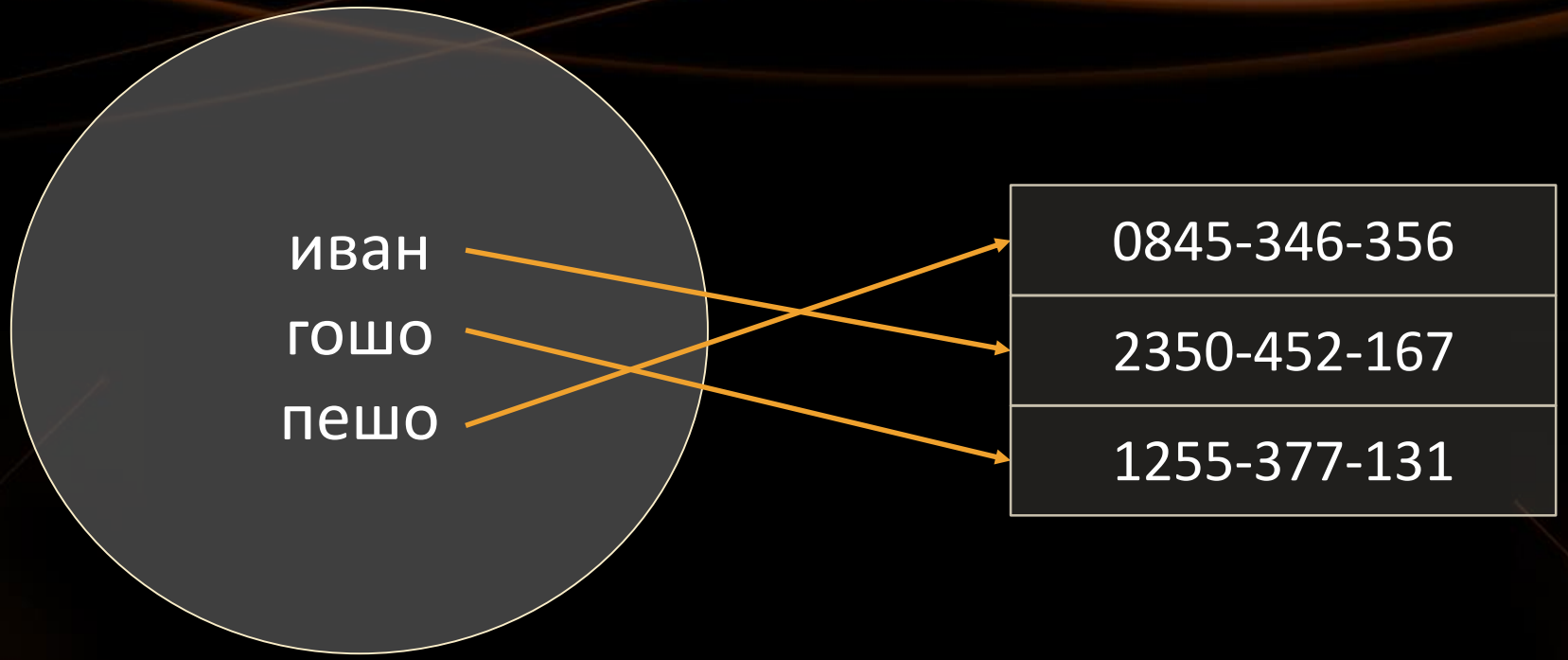
SoftUni Foundation

```
LinkedHashSet<Integer> firstPlayer = getPlayerNumbers();
LinkedHashSet<Integer> secondPlayer = getPlayerNumbers();
for (int i = 0; i < 50; i++) {
  int firstNumber = firstPlayer.iterator().next();
  firstPlayer.remove(firstNumber);
//TODO: get top number for second player
  if (firstNumber > secondNumber) {
    firstPlayer.add(firstNumber);
    firstPlayer.add(secondNumber);
  } else if (secondNumber > firstNumber)
//TODO: finish logic about second player win or draw
//TODO: print result
}
```

SoftUni Foundation

HashSet<E>, TreeSet<E> and LinkedHashSet<E>

Exercises in class



Associative Arrays

`HashMap<Key, Value>`

Associative Arrays (Maps)

SoftUni Foundation

- Associative arrays are arrays indexed by keys
  - Not by the numbers 0, 1, 2, …
- Hold a set of pairs `<key, value>`
- Traditional array

key	0	1	2	3	4
value	8	-3	12	408	33

- Associative array

key	value
John Smith	+1-555-8976
Lisa Smith	+1-555-1234
Sam Doe	+1-555-5030

Maps Methods

- Initialization

```
HashSet<String, Integer> hash = new HashSet<String>();
```

Type of keys

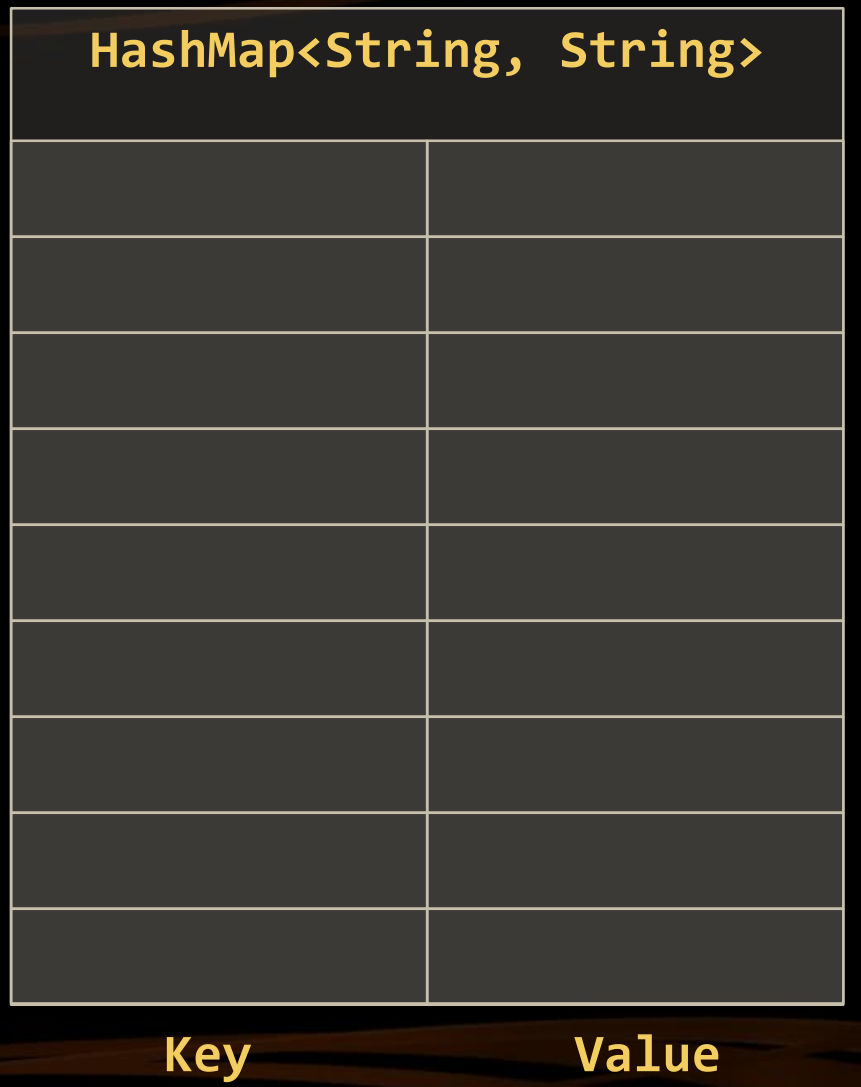
Type of values

- .size()
- .isEmpty()

```
HashSet<String> hash = new HashSet<>();
System.out.println(hash.size()); // 0
System.out.println(hash.isEmpty()); // True
```

HashMap<K, V> – put()

Hash Function



HashMap<K, V> – remove()

Pesho

Hash Function

HashMap<String, String>	
Gosho	0881-456-987
Pesho	0881-123-987
Alice	+359-899-55-592

Key Value

Looping Through Maps - Example

SoftUni Foundation

```
HashMap<String, Integer> vehicles = new HashMap<>();
vehicles.put("BMW", 5);
vehicles.put("Mercedes", 3);
vehicles.put("Audi", 4);
vehicles.put("BMW", 10);
for(String key: vehicles.keySet())
  System.out.println(key + " - " + vehicles.get(key));
```

Override first value

Return set of all keys

Return value for key

```
Audi - 4
Mercedes - 3
BMW - 10
```

Problem: Count Same Values in Array

- Write a program that counts in a given array of double values the number of occurrences of each value.

-2.5	4	4	3	-2.5	-5.5	4	3	3	-2.5	3

-2.5	3 – times
4	3 – times
-5.5	1 – times
3	4 – times

Solution: Count Same Values in Array

SoftUni Foundation

```
HashMap<String, Integer> result = new HashMap<>();
for (String number : input) {
  if (!result.containsKey(number)) {
    result.put(number, 1);
  } else {
    result.put(number, result.get(number) + 1);
  }
}
for (String key : result.keySet()) {
  System.out.println(key + " - " + result.get(key) + " times");
}
```

TreeMap<K, V> – put()

Alice	+359 888 123 987

Tree Map<String, String>	

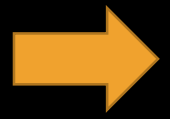
Key Value

Problem: Academy Graduation

SoftUni Foundation

- Write a program that:
  - Read list of students and their score for some courses
  - Print on console sorted list with average score for each student

Student	Java Advanced	Java OOP
Gosho	3.75	5
Mara	4.25	6
Pesho	6	4.5



Student	Average
Gosho	4,375
Mara	5,125
Pesho	7,25

Solution: Academy Graduation

SoftUni Foundation

```
TreeMap <String,Double[]> graduationList = new TreeMap<>();
for (int i = 0; i < numberOfStudents; i++) {
  String name = scanner.nextLine();
  String[] scoresStrings = scanner.nextLine().split(", ");
  Double[] scores = new Double[scoresStrings.length];

  for (int j = 0; j < scoresStrings.length; j++) {
    scores[j] = Double.parseDouble(scoresStrings[j]);
  }
  graduationList.put(name, scores);
}
//TODO print results
```

HashMap<K, V>, TreeMap<K, V>, LinkedHashMap<K, V>

SoftUni Foundation

- `size()` – the number of key-value pairs
- `keySet()` – a set of unique keys
- `values()` – a collection of all values
- Basic operations – `put()`, `remove()`, `clear()`
- Boolean methods:
  - `containsKey()` – checks if a key is present in the dictionary
  - `containsValue()` – checks if a value is present in the dictionary

SoftUni Foundation

Associative Arrays

Exercises in class

Summary

- `HashSet<E>`, `TreeSet<E>` and `LinkedHashSet<E>` hold unique elements and are very fast
- `HashMap<K, V>`, `TreeMap<K, V>` and `LinkedHashMap<K, V>` are an associative arrays where a value is accessed by its key

Java Advanced – Course Overview

SoftUni Foundation

Questions?

https://softuni.bg/courses/programming-fundamentals

License

SoftUni Foundation

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

- Attribution: this work may contain portions from
  - "Fundamentals of Computer Programming with Java" book by Svetlin Nakov & Co. under CC-BY-SA license

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

Software University

SoftUni Foundation

FORUM