Part 8

Grouping data using hash maps

Learning Objectives

- You know how to use a list as a hash map's value
- You know how to categorize data using a hash map

A hash map has at most one value per each key. In the following example, we store the phone numbers of people into the hash map.

```
HashMap<String, String> phoneNumbers = new HashMap<>();
phoneNumbers.put("Pekka", "040-12348765");

System.out.println("Pekka's Number " + phoneNumbers.get("Pekka"));

phoneNumbers.put("Pekka", "09-111333");

System.out.println("Pekka's Number " + phoneNumbers.get("Pekka"));
```

Sample output

```
Pekka's number: 040-12348765
Pekka's number: 09-111333
```

What if we wanted to assign multiple values to a single key, such as multiple phone numbers for a given person?

Since keys and values in a hash map can be any variable, it is also possible to use lists as values in a hash map. You can add more values to a single key by attaching a list to the key. Let's change the way the numbers are stored in the following way:

```
HashMap<String, ArrayList<String>> phoneNumbers = new HashMap<>();
```

Each key of the hash map now has a list attached to it. Although the new command creates a hash map, the hash map initially does not contain a single list. They need to be created separately as needed.

```
HashMap<String, ArrayList<String>> phoneNumbers = new HashMap<>();

// let's initially attatch an empty list to the name Pekka
phoneNumbers.put("Pekka", new ArrayList<>());

// and add a phone number to the list at Pekka
phoneNumbers.get("Pekka").add("040-12348765");
// and let's add another phone number
phoneNumbers.get("Pekka").add("09-111333");

System.out.println("Pekka's numbers: " + phoneNumbers.get("Pekka"));
```

Sample output

```
Pekka's number: [040-12348765, 09-111333]
```

We define the type of the phone number as `HashMap<String, ArrayList<String>>`. This refers to a hash map that uses a string as a key and a list containing strings as its value. As such, the values added to the hash map are concrete lists.

```
// let's first add an empty ArrayList as the value of Pekka
phoneNumbers.put("Pekka", new  ArrayList<>());

// ...
```

We can implement, for instance, an exercise point tracking program in a similar way. The example below outlines the `TaskTracker` class, which involves user-specific tracking of points from tasks. The user is represented as a string and the points as integers.

```
public class TaskTracker {
    private HashMap<String, ArrayList<Integer>> completedExercises;
```

```
    public TaskTracker() {
        this.completedExercises = new HashMap<>();
    }

    public void add(String user, int exercise) {
        // an empty list has to be added for a new user if one has not already
        this.completedExercises.putIfAbsent(user, new ArrayList<>());

        // let's first retrieve the list containing the exercises completed by
        ArrayList<Integer> completed = this.completedExercises.get(user);
        completed.add(exercise);

        // the previous would also work without the helper variable as follows
        // this.completedExercises.get(user).add(exercise);
    }

    public void print() {
        for (String name: completedExercises.keySet()) {
            System.out.println(name + ": " + completedExercises.get(name));
        }
    }
}
```

```
TaskTracker tracker = new TaskTracker();
tracker.add("Ada", 3);
tracker.add("Ada", 4);
tracker.add("Ada", 3);
tracker.add("Ada", 3);

tracker.add("Pekka", 4);
tracker.add("Pekka", 4);

tracker.add("Matti", 1);
tracker.add("Matti", 2);

tracker.print();
```

Sample output

```
Matti: [1, 2]
Pekka: [4, 4]
Ada: [3, 4, 3, 3]
```

Programming exercise:

Dictionary of many translations

Points

/

Your assignment is to create the class `DictionaryOfManyTranslations`. In it can be stored one or more translations for each word. The class is to implement the following methods:

- `public void add(String word, String translation)` adds the translation for the word and preserves the old translations.
- `public ArrayList<String> translate(String word)` returns a list of the translations added for the word. If the word has no translations, the method should return an empty list.
- `public void remove(String word)` removes the word and all its translations from the dictionary.

It's probably best to add the translations to an object variable that is of the type `HashMap<String, ArrayList<String>>`

An example:

```
DictionaryOfManyTranslations dictionary = new DictionaryOfManyTranslations(
dictionary.add("lie", "maata");
dictionary.add("lie", "valehdella");

dictionary.add("bow", "jousi");
dictionary.add("bow", "kumartaa");

System.out.println(dictionary.translate("lie"));
dictionary.remove("bow");
System.out.println(dictionary.translate("bow"));
```

Sample output

```
[maata, valehdella]
[]
```

Error loading exercise details

Programming exercise:

Storage facility (2 parts)

Points 2/2

NB! By submitting a solution to a part of an exercise which has multiple parts, you can get part of the exercise points. You can submit a part by using the 'submit' button on NetBeans. More on the programming exercise submission instructions: Exercise submission instructions.

Part 1: Adding items and examining contents

Your task is creating a class called `StorageFacility` that can be used to keep track of storage units and their contents. The class is to implement the following methods:

- `public void add(String unit, String item)` adds the parameter item to the storage unit that is also given as a parameter.
- `public ArrayList<String> contents(String storageUnit)` returns a list that contains all the items in the storage unit indicated by the parameter. If there is no such storage unit or it contains no items, the method should return an empty list.

Here's an example:

```
StorageFacility facility = new StorageFacility();
facility.add("a14", "ice skates");
facility.add("a14", "ice hockey stick");
facility.add("a14", "ice skates");

facility.add("f156", "rollerblades");
facility.add("f156", "rollerblades");

facility.add("g63", "six");
facility.add("g63", "pi");

System.out.println(facility.contents("a14"));
System.out.println(facility.contents("f156"));
```

Sample output

```
[ice skates, ice hockey stick, ice skates]
[rollerblades, rollerblades]
```

Part 2: Listing the units and removing from unit

- Now the class `StorageFacility` contains the functionality to add an item to a storage unit and to list the contents of a unit. Next add the possibilities to remove an item from a storage unit and to list all the units.
- `public void remove(String storageUnit, String item)` removes the given item from the given storage unit. NB! Only removes one item — if there are several items with the same name, the rest still remain. If the storage unit would be empty after the removal, the method also removes the unit.
- `public ArrayList<String> storageUnits()` returns the names of the storage units as a list. NB! Only the units that contain items are listed.

An example:

```
StorageFacility facility = new StorageFacility();
facility.add("a14", "ice skates");
facility.add("a14", "ice hockey stick");
facility.add("a14", "ice skates");

facility.add("f156", "rollerblades");
facility.add("f156", "rollerblades");

facility.add("g63", "six");
```

```
facility.add("g63", "pi");

facility.remove("f156", "rollerblades");

System.out.println(facility.contents("f156"));

facility.remove("f156", "rollerblades");

System.out.println(facility.storageUnits());
```

Sample output

```
[rollerblades]
[a14, g63]
```

The order of the storage units in the output may be different from this example.

Exercise submission instructions

How to see the solution

You have reached the end of this section! Continue to the next section:

→ 5. Fast data fetching and grouping information

Remember to check your points from the ball on the bottom-right corner of the material!

In this part:

4. Grouping data using hash maps

5. Fast data fetching and grouping information

 Source code of the material

This course is created by the Agile Education Research -research group of the University of Helsinki.

Credits and about the material.




HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI


MASSIIVISET AVOIMET VERKKOKURSSIT
MASSIVE OPEN ONLINE COURSES · MOOC.FI