Programming

New Austrian Coding School

week 8 - exercises

Exercise 1

Develop the following code:

1. Create a list of words. Make sure you use *new ArrayList* to have a mutable list.
2. Use the *removeIf* method to remove every word that ends with the letter *e*.
3. Use the *replaceAll* method to transform every word in their uppercase version.
4. Use the *removeIf* method to remove every word that has less than six letters.
5. Use the *forEach* method to print every word one by one.

```
Part 1:
["somewhere", "scenario", "table", "cable", "glass", "backpack", "mouse"]
Part 2:
[scenario, glass, backpack]
Part 3:
[SCENARIO, GLASS, BACKPACK]
Part 4:
[SCENARIO, BACKPACK]
Part 5:
SCENARIO
BACKPACK
```

Exercise 2

Develop the following code:

1. Create the *NumberFilter* class.
2. Add to it the *getEven* method that receives a list of numbers and returns a list with only its even numbers.
3. Add to it the *getOdd* method that receives a list of numbers and returns a list with only its odd numbers.
4. Write tests for both methods.

Rules:

- You're not allowed to use the *removeIf* method of any kind of data structure.
- You're not allowed to duplicate code.
- Use a *Predicate<Integer>* to avoid code duplication.

Exercise 3

Develop the following code:

- Create the *KeywordFinder* class.
- Add to it the *findElegant* method that receives a sentence and returns the list of all its *elegant* words.
- Add to it the *findPlayful* method that receives a sentence and returns the list of all its *playful* words.
- Write tests for both methods.

Rules:

- An *elegant* word is a word that starts with *ele*.
- A *playful* word is a word that ends with *ful*.
- You're not allowed to duplicate code.
- Use a *Predicate<String>* to avoid code duplication.

```
Input for findElegant:
“The elephant is lifted eleven floors easily with the help of an electricity elevator”
Output:
["elephant", "eleven", "electricity", "elevator"]
```

```
---

Input for findPlayful:
“The rightful heir of the powerful king had an awful accident playing with a colorful bear”
Output:
["rightful", "powerful", "awful", "colorful"]
```

Exercise 4

Develop the following code:

1. Create the *PowerLevelScouter* class.
2. Add to it the *scout* method that receives a name and returns its power level as a number. The power is calculated by summing up the *ASCII* code value of each letter.
3. Add to it the *scoutEnhanced* method that receives a name and returns its power level as a number. The power is calculated by transforming the name to lowercase and then summing up the *ASCII* code value of each letter.
4. Write tests for both methods.

Rules:

- You’re not allowed to duplicate code.
- Use a *Function<String, Integer>* to avoid code duplication.

```
Input for scout:
“Susana”
Output:
619

---

Input for findPlayful:
Input for scout:
“Susana”
Output:
651
```

Exercise 5

Find on the internet or create yourself an exercise where it makes sense to use a *Predicate* and develop it.

Exercise 6

Find on the internet or create yourself an exercise where it makes sense to use a *Function* and develop it.