

Mandag d. 1. maj 2023

For the following exercises make a class with a main() method. Add to the class the methods mentioned in the exercises (make all the methods static), and test the added methods in the main() method. The first exercise must be solved in your table group. Discuss and decide how you most beneficial for all can split up in two pairs and solve both exercises by pair programming.

Group exercise

In this exercise you must use a Player class. Make a player class with the following fields: name, height, weight, scoredGoals. The class must have a constructor that initializes all the fields, and get methods for all four fields (and a toString() method).

In the test class make an ArrayList of Player with about 6-8 Players.

a) Write a *linear* search method that returns a player with a given number of scored goals found in an ArrayList of players given as parameter. Return null, if no such player is found.

The header of the method:

```
public Player findPlayerLinear(ArrayList<Player> players, int score)
```

b) Write a *binary* search method that returns a player with a given number of scored goals found in a *sorted* ArrayList of players (sorted in descending order according to scored goals) given as parameter. Return null, if no such player is found.

The header of the method:

```
public Player findPlayerBinary(ArrayList<Player> players, int score)
```

Exercise 1

Write a linear search method that returns, whether an uneven number exists in an array. The method must return true or false, and take an array of integer numbers as parameter. Test the method.

Exercise 2

Write a linear search method that finds the first number belonging to the interval [10;15]. The method must return the number found in the interval, and take an array of integer numbers as parameter. If a number in the interval is not found, the method must return -1. Test the method.

If the array is [7, 56, 34, 3, 7, 14, 13, 4], the method should return 14.

Exercise 3

Write a linear search method that returns true, if two adjacent numbers are the same. The method must take an array of integer numbers as parameter.

Test the method.

If the array is [7, 9, 13, 7, 9, 13], the method must return false. If the array is [7, 9, 13, 13, 9, 7], the method must return true.

Write another method that returns true, if the same number exists in n adjacent places. The method must take an array of integer numbers and the number n as parameters.

Test the method.

Exercise 4

Write a method that returns all indexes of a given String in a given ArrayList<String>.

The header of the method:

```
public static ArrayList<Integer> findAllIndices(ArrayList<String> list, String target)
```

Exercise 5 *

Write a search method, named repeatedChars(), with two parameters: a string s, and a positive int k.

The method must return a boolean indicating whether the same character is found in k adjacent positions in the string. This is an advanced searching.

Examples:

repeatedChars("vnhstxxxaby",3) returns true, because the character x is found in 3 adjacent places in the string.

repeatedChars("vnhstxxxaby",4) returns false.