Programming Basics
Prof. Dr Silke Lechner-Greite
WiSe 2021 / 2022

Exercise sheet 5 – Arrays & algorithms

Task 1 – Prime number test

A prime number is a number that can only be divided by 1 and by the number itself.
Write a method that tests whether a given number is a prime number. Test this method with a few selected numbers.

Task 2 - Letter histogram

Count the number of times each character (i.e. letter) occurs within a string.
Example: Anna: 2 x a, 2 x n.
Write a method that counts the characters within a string. Test this method with a few selected strings.

Task 3 – Add an element to an array

As you know, the size of an array is not changeable. However, sometimes it is necessary to append an element to an array.
Write a method that gets an array and an element passed as parameters. The method should append the given element to the end of the given array.
For example, append([1, 2, 3], 4) returns the array [1, 2, 3, 4]
Test the method using appropriate data.

Task 4 – Binary search

A binary search can be used to quickly find an element in pre-sorted arrays.

From Wikipedia:
Binary search works on sorted arrays. Binary search begins by comparing an element in the middle of the array with the target value. If the target value matches the element, its position in the array is returned. If the target value is less than the element, the search continues in the lower half of the array. If the target value is greater than the element, the search continues in the upper half of the array. By doing this, the algorithm eliminates the half in which the target value cannot lie in each iteration.

Write a method that gets a sorted array and the element to be searched for as input, and returns the index of that element. The method should return -1 if the element is not present in the array.

Test the method using appropriate data.

Task 5 – Earnings calculator

This example is from the lecture on Chapter 5 – Arrays:

```java
public class EarningsCalculator {

    public static double earnings(double hours, double wage, double factor) {
        return hours <= 8.0
                ? hours * wage
                : (8.0 + factor * (hours - 8.0)) * wage;
    }

    public static void main(String[] args) {
        final double wage = 15.0;
        final double factor = 1.15;
        double hoursMon = 8.0;
        double hoursTue = 8.0;
        double hoursWed = 9.0;
        double hoursThur = 9.0;
        double hoursFri = 6.0;
        double hoursSat = 8.0;

        double total =
                earnings(hoursMon, wage, factor) +
                earnings(hoursTue, wage, factor) +
                earnings(hoursWed, wage, factor) +
                earnings(hoursThur, wage, factor) +
                earnings(hoursFri, wage, factor) +
                earnings(hoursSat, wage, factor);
        System.out.println(total);
    }
}
```

1. What does the return statement in the method earnings(.) mean? How could this be written differently?
2. Based on the concept of arrays: modify the given programme so that:
    a. The hours are stored in an array.
    b. The total hourly wage is determined by means of the forEach loop and the method earnings(.) and stored in a variable, and then output to the console.
    c. Expand your programme to include the method append(.) from Task 3, and change this into double values and double arrays.
    d. Expand your programme to include a method remove() that reduces a double array by one element, which is specified by an index in the parameter list.
3. Test your programme for different hour arrays by appending more hours, and also by deleting hours at a specific position. Output the relevant information to the console.