

# Exercises on ArrayLists

Today we learned how to use ArrayLists instead of normal Arrays. Let's first practice how to use them with Strings. Afterwards we can work with Lists of objects of classes we have written ourselves.

## Exercise 1: Save and search

Write a program which uses a Scanner to read an unknown number of lines of text from the user. This means one could enter no line of text at all just as well as hundreds of lines, we simply don't know it.

Save each line as an element of an ArrayList. Stop reading and saving lines of text if the user enters the word "Stop".

Now ask the user for a word or phrase he wants to search for (only one line). Then search for this String in the ArrayList and print out every line which *contains* the search term.

After that is done, terminate the program.

### Example:

The following sample console output shows how an execution of the program could look like:

Please enter lines of text which you want to store:

An object is a software bundle of related state and behavior. Software objects are often used to model the real-world objects that you find in everyday life.

A class is a blueprint or prototype from which objects are created.

A package is a namespace for organizing classes and interfaces in a logical manner. Placing your code into packages makes large software projects easier to manage.

Stop

Please enter a search term:

object

An **object** is a software bundle of related state and behavior. Software **objects** are often used to model the real-world **objects** that you find in everyday life.

A class is a blueprint or prototype from which **objects** are created.

## Exercise 2: Classroom

Now it is time to use ArrayLists to store objects of your own classes.

Write a class Student whose instances represent students. A student has

- registration number
- age (in years)
- name
- gender("male","female")
- nationality ("stateless", "Iran", "Austria", "Afghanistan", ...)
- study („Informatics“, „Engineering“, „Architecture“, „Electronics“)

Remember to choose appropriate data types for all fields.

Write two constructors: One without parameter, one with a parameter for each attribute.  
Write the corresponding getter and setter methods.

In the class containing the main method do the following:

- Write two methods *filterStudentsByGender* and *filterStudentsByStudy*, which take an ArrayList of Students and a *gender* or *study* as a parameters and return a new ArrayList, which contains only those students which have the requested *gender* or *study*.
- Write methods *sumOfAges* and *averageOfAges*, which take a List of Students and return the sum of their ages or the average of them, respectively.

In the main method:

- Define an ArrayList of Students
- Write a loop (Scanner) for entering each student.
- After the last student has been entered, print the average age of all students of *Informatics* and the sum of all students of *Electronics*.

Hint: you can manage this last point by combining the methods written above. Do not reinvent the wheel here.