Exercises: Week 3

Introductory Programming 2020

Exercise 4.1

Open the music-organizer-v1 project from the exercise files and create a `MusicOrganizer` object. Store the names of a few audio files into it — they are simply strings. As we are not going to play the files at this stage, any file names will do.
Check that the number of files returned by `numberOfFiles` matches the number you stored. When you use the `listFile` method, you will need to use a parameter value of 0 (zero) to print the first file, 1 (one) to print the second, and so on. We shall explain the reason for this numbering in due course.

Exercise 4.2

What happens if you create a new `MusicOrganizer` object and then call `removeFile(0)` before you have added any files to it? Do you get an error? Would you expect to get an error?

Exercise 4.3

Create a `MusicOrganizer` and add two file names to it. Call `listFile(0)` and `listFile(1)` to show the two files. Now call `removeFile(0)` and then `listFile(0)`. What happened? Is that what you expected? Can you find an explanation of what might have happened when you removed the first file name from the collection?

Exercise 4.8

If a collection stores 10 objects, what value would be returned from a call to its `size` method?

Exercise 4.9

Write a method call using `get` to return the fifth object stored in a collection called items.

Exercise 4.10

What is the index of the last item stored in a collection of 15 objects?

Exercise 4.12

Write a method call to remove the third object stored in a collection called `dates`.

Exercise 4.13

Suppose that an object is stored at index 6 in a collection. What will be its index after the objects at index 0 and index 9 are removed?

Exercise 4.14

Add a method called `checkIndex` to the `MusicOrganizer` class. It takes a single integer parameter and checks whether it is a valid index for the current state of the collection. To be valid, the parameter must lie in the range `0` to `size()-1`.
If the parameter is not valid, then it should print an error message saying what the valid range is. If the index is valid, then it prints nothing. Test your method with both valid and invalid parameters. Does your method still work when you check an index if the collection is empty?

Exercise 4.15

Write an alternative version of `checkIndex` called `validIndex`. It takes an integer parameter and returns a boolean result. It does not print anything, but returns `true` if the parameter's value is a valid index for the current state of the collection, and `false` otherwise. Test your method with both valid and invalid parameters. Test the empty case too.

Exercise 4.16

Rewrite both the `listFile` and `removeFile` methods in `MusicOrganizer` so that they use your `validIndex` method to check their parameter, instead of the current boolean expression. They should only call get or remove on the `ArrayList` if `validIndex` returns `true`.