Week 2
Getting Started in IntelliJ and Java

Comment

The purpose of this practical sheet is to introduce you to `IntelliJ IDEA`, the course-supported IDE. IntelliJ should be installed on all the lab computers, and can be downloaded on your personal machines as well. Details on this can be found in the course `IntelliJ Guide` on Blackboard.

1 Basic Project

You will need to create a project in order to write code in your IDE, and the IntelliJ Guide on Blackboard provides instructions on how to accomplish this. Use this guide to help you create a class which can print out "Hello, World!" (not the complicated version from lectures). There is also a skeleton program in the lecture slides which may be of help.

2 Java Documentation

Documentation for all standard Java classes can be found at https://docs.oracle.com/en/java/javase/11/docs/api/index.html. If you are using it frequently on your own machine, it may be worthwhile downloading the documentation. Also note that googling documentation will often give results from Java 8. While we are not using this version, often the documentation for classes you are likely to encounter will be virtually identical in Java 11.

Locate the documentation for the `String` class (in the `java.lang` package which is in the `java.base` module). **Some methods in this class will be helpful in the following exercise**.

3 Palindrome Exercises

A palindrome is a string which reads the same forwards as backwards, such as `"AaaA"` or `"madamimadam"`. Before attempting to write code, work out on paper how you would check if a string is a palindrome, then write the following methods. All four methods check whether a string is a palindrome, however each uses a different technique. You will need a main method in your class in order to test that they work. You should test at least the following strings: `"AaA", "A", ""` (should return `true`) and `"Abbb"` (should return `false`).

1. `public static boolean isPalindrome1(String word)` — Use a `for` loop.
2. `public static boolean isPalindrome2(String word)` — Use a `while` loop.
3. `public static boolean isPalindrome3(String word)` — Implement it recursively (with no helper methods). Think about how you can reduce the problem to a smaller one. If x...x is a palindrome, then what must be true about ...? How could we extract ...?
4. `public static boolean isPalindrome4(String word)` — Implement it recursively with the following helper function:

```
public static boolean helper(String word, int i)
```
What the helper would do is up to you.