COMP122 Week 4

Javadoc and Version Control

UNIVERSITY OF LIVERPOOL

Dr. Patrick Totzke
totzke@liverpool.ac.uk

https://liverpool.instructure.com/courses/59716

Your Questions?

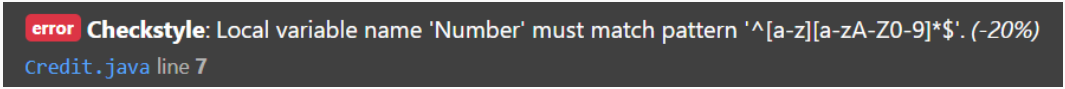
Q: I'm having a lot of troubles keeping up to speed and learning java. What is the advisable thing to do?

A:

1. (re)watch my videos
2. ask questions/discuss with TAs and others (on canvas)
3. practice. a lot. Use the exercises and autochecker to get feedback on your attempts.

Q24

Q: I have found that the check style section consistently gives strange and unhelpful suggestions like the one below.



A: This reports that the variable name is unconventional.

Q: Do we need to submit a report or line-by-line documentations?

A: No.

Q: When I run CodeGrade on my submission, some checks fail but when I run it I get the expected result. Do I get points?

A: No.

Q: Are there more resources for A1 part 2?

A:

1. Google “Caesar frequency analysis”
2. John Fearnley has recorded a video for COMP105 (ask fellow students nicely)

Q: I'm done with A1. What now?

A: Great! Make sure your code works with any integer shift as expected.

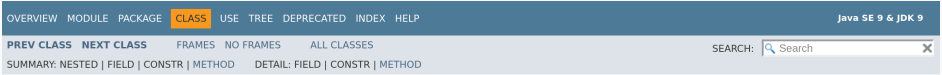
Q: Can I contact the TAs out of lab times?

A: Yes, post on the Canvas forum.

No, I do not give out their contacts nor do I expect them to work out of paid hours.

Javadoc

Ever wondered how these were generated?



OVERVIEW MODULE PACKAGE CLASS USE TREE DEPRECATED INDEX HELP Java SE 9 & JDK 9

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES SEARCH: Search

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Module java.base
Package java.util

Class Arrays

java.lang.Object
  java.util.Arrays

```
public class Arrays
extends Object
```

This class contains various methods for manipulating arrays (such as sorting and searching). This class also contains a static factory that allows arrays to be viewed as lists.

The methods in this class all throw a `NullPointerException`, if the specified array reference is null, except where noted.

The documentation for the methods contained in this class includes brief descriptions of the *implementations*. Such descriptions should be regarded as *implementation notes*, rather than parts of the *specification*. Implementors should feel free to substitute other algorithms, so long as the specification itself is adhered to. (For example, the algorithm used by `sort(Object[])` does not have to be a MergeSort, but it does have to be *stable*.)

This class is a member of the Java Collections Framework.

Since:
1.2

Method Summary

All Methods | Static Methods | Concrete Methods

Modifier and Type	Method	Description
static <T> List<T>	asList(T... a)	Returns a fixed-size list backed by the specified array.

Javadoc

... is a command-line tool to generate API docs as seen e.g. on `oracle.com` directly from your source code.

- designated "javadoc comments", `/** comment */`.
- a comment documents a `class`, `interface`, `method`,..., if it appears directly before.
- special markup tags inside these comments are interpreted by javadoc. For example `@author`,`@param`,...
- the structure (Class Hierarchy) is automatically extracted.

Javadoc Examples

A javadoc comment documenting class `java.awt.Window`; taken from `https://docs.oracle.com/javase/8/docs/technotes/tools/unix/javadoc.html`

```
/**
 * A class representing a window on the screen.
 * For example:
 * <pre>
 *     Window win = new Window(parent);
 *     win.show();
 * </pre>
 *
 * @author  Sami Shaio
 * @version 1.13, 06/08/06
 * @see     java.awt.BaseWindow
 * @see     java.awt.Button
 */
class Window extends BaseWindow {
    ...
}
```

Javadoc Examples

A javadoc comment documenting `String.charAt()`; taken from

`https://docs.oracle.com/javase/8/docs/technotes/tools/unix/javadoc.html`

```
/**
 * Returns the character at the specified index. An index
 * ranges from <code>0</code> to <code>length() - 1</code>
 *
 * @param     index the index of the desired character.
 * @return    the desired character.
 * @exception StringIndexOutOfRangeException
 *              if the index is not in the range <code>0</code>
 *              to <code>length()-1</code>
 * @see       java.lang.Character#charValue()
 */
public char charAt(int index) {
   ...
}
```

Javadoc – Command-line invocation

The binary is part of the JavaSE, so you should already have it by now.

To generate docs, say for all files in a directory `src`, and write them to directory `docs`, you'd invoke javadoc like this.

```
$> javadoc -d docs -author solution/*java
```

(the `-author` parameter causes it not to drop the `@author` tag). **DEMO?**

For more documentation, I recommend

- javadoc's man-page (see `man javadoc`)
- `https://docs.oracle.com/javase/8/docs/technotes/tools/unix/javadoc.html`
- `https://en.wikipedia.org/wiki/Javadoc`

Version Control

Version Control?

Have you ever found yourself...

- using the undo/redo functions of your editor while programming?
- copied whole files back and forth to "safely" keep an older version?
- collaborated on source code?
- worked on independent features simultaneously?
- wondered when and why you made some code change?

Version Control to the rescue!

People have long addressed these things using software tools, which help organize code and provide standard solutions/workflows.

Version Control Systems

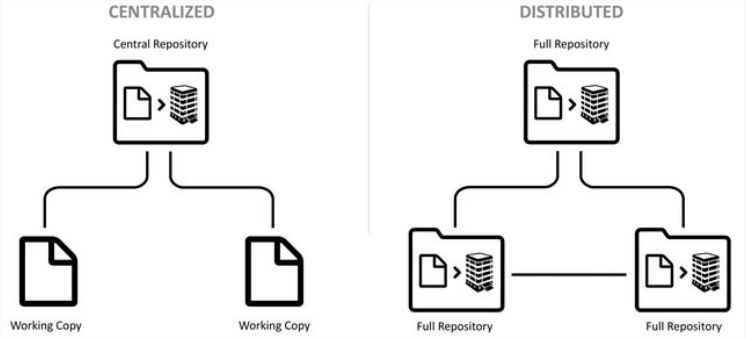
Version Control Systems (VCS) = Source Code Management (SCM) tools = Revision Control Systems (RCS) usually provide:

- A long-term history of every file
- Branching and merging of code
- Traceability: you can annotate code changes
- Rich tool support, such as IDE integration, command-line tools, GUIs and dedicated websites (svn-web,github, bitbucket, gitlab)

Local v Centralized v Distributed VCSs

CVS, SVN

mercurial, bazaar, git



©Sofia Feist https://www.researchgate.net/publication/316553817_Collaborative_Algorithmic-based_Building_Information_Modelling

Start using git

Initial setup

```
$> git config --global user.name "John Doe"
$> git config --global user.email johndoe@example.com
```

Creating a repository

```
$> cd /home/user/my_project
$> git init
```

Making a commit = save a snapshot

```
$> git add *.java
$> git commit -a -m "This introduces amazing feature x"
```

Getting help: Try `git help` or `man git` or check out `https://git-scm.com/`

GitHub and GitLab

`github.com` and `gitlab.com` are popular hosting services for git repositories, both with extra bells and whistles. Try them out?

The department runs a local GitLab instance at `https://gitlab.csc.liv.ac.uk`, (use your MWS login).

Do not publish your COMP122 solutions!

Schedule for Wednesday

There will be no lecture on Thu. 23 February.

1. Do the lab exercises (especially "strings")
2. Work on A1 and submit by Friday 24.

Summary of Week 4

We looked at...

- Javadoc
- Version Control

Next Week: Block 2!

- Objects
- Classes!