Due March 3rd at 12:00 NOON

Tutorial 4

W22 – COMP 1006/1406 – Introduction to Computer Science II

Objectives

Introduce and use **javadocs** and java **Enum** classes.

Notes

Question 1 introduces you to Java **Enum** classes. You are asked to read and run some code to gain some understanding of enums and then write a method that uses them.

Question 2 introduces you to the javadoc tool to make html documentattion for class APIs. You will use a command line (cmd/powershell or terminal) to run the javadoc program to generate the API for a provided class. used in Assignment 1. You will also add javadoc comments to your class from Question 1 in this tutorial.

Submit your updated UseMonthEnumType.java file.

QUESTION 1

An **Enum** (enumeration) type is a special Java class to hold constants (names). By convention, constants in Java are always labelled in FULL CAPS. Essentially, objects of the enum class are each of the possible constants.

See the **Month.java** enum type provided. We can use this Month enum when we need to use months.

Java provides some pre-built methods in every **enum** class. These include:

- **toString()** will return a string representation of the constant's name.
- **values()** will return an array containing all possible constants (in the order that they appear in the enum definition.
- **valueOf(String)** will return the enum object corresponding to the input string (if possible).

Run the **UseMonthEnumType** program to see how these methods used.

Add a static method to the UseMonthEnumType program. The method should look like

```
public static int days(Month month)
```

The method will have a **switch** statement that will determine the number of days in the provided (input) month. The function will then return this number. Don't worry about leap years (i.e., FEBRUARY will always have 28 days in this exercise).

Assigned Reading

https://docs.oracle.com/javase/tutorial/java/javaOO/enum.html

Note: In Assignment 2, the Temperature class used the Scale class of constant (String) names of temperature scales. Using an enum type would have been a better design decision.

More Reading

http://tutorials.jenkov.com/java/enums.html

QUESTION 2

In this problem, you will use the **javadoc** program (that is part of the JDK installation) from the command line (shell).

First, check that you can access the javadoc program. In a terminal, type

```
javadoc --version
```

It should return `javadoc XX` (if you installed the JDK for Java XX). If not, you will need to make sure you have the program installed and that your path is set to find it. See the end of the tutorial document for help with this.

Look at the code in the provided **Find** class. You will notice that the comments might look slightly different than normal Java comments. The commenting used is in the Javadoc format. What is Javadoc? From wiki, we see

> *Javadoc is a documentation generator from Oracle Corporation for generating API documentation in HTML format from Java source code. The HTML format is used to add the convenience of being able to hyperlink related documents together.*

An **API** is the **application programming interface**. This is the interface that allows other code to use a given class. It is the public methods that allow us to interact with the data in an object.

In order to generate the *html* code for the API, we will use the **javadoc** program from the command line (using cmd or terminal). In the directory where you compile and run your code for this tutorial, type

```
javadoc -d docs Find.java -author -version
```

This will create a new directory called **docs** in your current directory (where Find.java is located). Inside this new directory, you will find the *html* code for the documentation for the API for the Find class. In the windows file viewer, or equivalent in MAC OS, click on the **index.html** file. This will open the file in your default web browser and show you the API documentation for the Find class.

What happened?

- `-d docs` specifies where to put the html files (for the API)
- `Find.java` specified which file to generate javadocs for (use *.java for all java files)
- `-author -version` specifies to list the author and version (if specified in the java files).

How do you write javadoc comments? In your code, you can add a special comment block just **before** a class, attribute, method or constructor declaration. This block of comments will be used in the generated API to describe whatever it is that this comment block comes before. There are some special tags that the Javadoc tool will read and use in this comment block.

Here are some simple rules

- The comment block must start with **`/**`** (two stars instead of 1) and end with **`*/`** (Using /* … */ is a normal comment block and not a javadoc comment block.)

- For methods, each input argument should have an associated **@param** tag describing that input. (We can add pre-conditions on the argument here.)
- For non-void methods, the **@return** tag is used to describe the output (and any post-conditions).
- You can use basic HTML tags to help format the text. For example, `<code>main</code>` will format *main* in code format. Use `<p>` to start a new paragraph (with a blank line).

- The **@author** tag will list the author of the class. (This may or may not be used in practice since version control systems like git will have this information instead.
- The @**version** tag will list the version of code. The @**since** tag can used to indicate when a change has first existed.

Add some Javadoc comments to the method you created in the UseEnumMonthType class. Run the javadoc program to view your comments.

More Reading

- https://www.codeproject.com/Articles/658382/Basic-Javadoc-guide

- https://www.oracle.com/ca-en/technical-resources/articles/java/javadoc-tool.html

Javadoc Help

If you installed Java 17 for this course this semester and it automatically set the path so that you can run java and javac, it might not have set the path so that javadoc can run. Following these instructions should help solve this problem.

https://docs.oracle.com/en/java/javase/15/install/installation-jdk-microsoft-windows-platforms.html#GUID-96EB3876-8C7A-4A25-9F3A-A2983FEC016A