CMSC131 Fall 2021 (https://www.cs.umd.edu/class/fall2021/cmsc131-01XX-03XX/)

Project #5, PhotoManager (Fri, Oct 29)

Objectives

To practice class definition, use of a class by another class, ArrayLists and Collections class.

Overview

For this project you will implement the class Photo and PhotoManager. These classes support a photo manager.

For this project you can discuss the project with classmates, but you may not exchange code nor write code together. Your goal is to learn so you can be successful in internships and class exams. You do not need to include the names of people your discuss the project with. This project has been used in previous installments of the course. Some students may find implementations for this project online. If you use any such implementation you will be violating academic integrity rules and an appropriate academic case will be submitted which may result in an XF in the course.

Please do not post implementations of this project on the web where others can see them. This leads to academic integrity violations. Thank you for your cooperation.

Grading

- (60%) Public Tests
- (12%) Release Tests
- (10%) Secret Tests
- (10%) Student Tests
- (8%) Style (Good indentation, Good Variable Names, Avoiding Code Duplication)

Code Distribution

The project's code distribution is available at PhotoManager.zip (https://www.cs.umd.edu/class/fall2021/cmsc131-01XX-03XX/prot/projects/zipFiles/PhotoManager.zip) The code distribution provides you with the following:

- A file named **Photo.java** - This is a class you need to implement.
- A file named **PhotoManager.java** - This is a class you need to implement.
- A file named **PublicTests.java** - This represents the public tests for the project.
- A file named **StudentTests.java** - Where you will write student tests.
- A file named **Driver.java** - Illustrates how to use some of the methods of the classes you need to implement. We have left two files: DriverWebPage1Expected.html and DriverWebPage2Expected.html that provide the expected content of files created by the driver (the actual files created by the driver are DriverWebPage1.html and DriverWebPage2.html).
- **expectedResults** - A folder that has the expected results for each of the public tests.
- **results** - A folder that the results generated by your code for each of the public tests.

Specifications

A description of the methods you need to implement for each class can be found at javadoc (doc/index.html).

Other

- For your project, the initialization of instance variables should take place in the constructors and not at declaration time.
- Your tests will fail if your toString() method is not generating the correct string.
- Although you could define them, there is no need to define additional methods.
- Remember that hard-coding your code so you pass tests is considered an academic integrity violation.
- When you run public tests, files with the "Results.txt" extension (e.g., pub1ConsResult.txt) will be created if no exceptions where thrown by your code (an exception is an error). To see the file(s) created after running public tests, you need to Refresh the Eclipse project (right-click on the project and select Refresh)
- You have 5 release tokens that regenerated every 24 hours.

Student Tests

Test your code as you develop it, by writing tests in the **StudentTests.java** file. See the information we provided in that file. Using a JUnit test method is equivalent to writing a main method that calls the methods you are implementing. The JUnit test method will help you as you don't have to write the main method (just the code that appears inside). For this project you don't have to worry about using assertions with your student tests. **You may not be able to get help during office hours unless you have student tests. Some students implement their project by running public tests over and over; this is fine at the beginning, but now you have to start writing your own tests, as public tests may not be present in future projects.**

To grade your student tests, we will check that you have at least one test for each of methods of the classes you need to implement.

You should write your tests as you write your methods (that is the correct approach). Remember that in the real world you will not have public tests; you have to learn how to write your own tests. Also, we have secret tests, so if you want to pass them, you need to thoroughly test your code. **By the way, sharing of student tests is not allowed.**

Sample Driver

The following driver relies on the class you need to implement. You will find this driver in the code distribution. The html files this driver generates can be found in the code distribution.

Driver

```
public class Driver {

	public static void main(String[] args) {
		PhotoManager photoManager = new PhotoManager();
		String target = "umcp/college1.jpg", answer = "";

		/* Adding photos */
		photoManager.addPhoto(target, 300, 400, "09/17/2020-17:10");
		photoManager.addPhoto("umcp/college3.jpg", 200, 200, "11/18/2019-09:00");
		photoManager.addPhoto("umcp/college8.jpg", 200, 200, "10/18/2020-18:30");

		answer += "PhotoManager\n";
		answer += photoManager + "\n";
		photoManager.addComment(target, "School Visit");
		photoManager.addComment(target, "Cousins with me");
		answer += "Retrieving comments for " + target + "\n";
		answer += photoManager.getComments(target);
		photoManager.sortPhotosByDate();
		answer += "\nAfter sorting photos by date\n" + photoManager + "\n";
		System.out.println(answer);
		photoManager.createHTMLPage("DriverWebPage1.html");
		System.out.println("Loading Photos");
		PhotoManager photoManager2 = new PhotoManager();
		photoManager2.loadPhotos("photoInfoToLoad.txt");
		photoManager2.createHTMLPage("DriverWebPage2.html");
	}
}
```

Output

```
PhotoManager
umcp/college1.jpg,300,400,09/17/2020-17:10
umcp/college3.jpg,200,200,11/18/2019-09:00
umcp/college8.jpg,200,200,10/18/2020-18:30

Retrieving comments for umcp/college1.jpg
School Visit,Cousins with me
After sorting photos by date
umcp/college3.jpg,200,200,11/18/2019-09:00
umcp/college1.jpg,300,400,09/17/2020-17:10
umcp/college8.jpg,200,200,10/18/2020-18:30


DriverWebPage1.html has been created (Refresh Eclipse project to see the file).
Loading Photos
DriverWebPage2.html has been created (Refresh Eclipse project to see the file).
```

Submission

Submit your project as usual. Make sure you check the results in the submit server. Remember that you need to do release testing to see release tests results.

Academic Integrity

Please make sure you read the academic integrity section of the syllabus so you understand what is permissible in our programming projects. We want to remind you that we check your project against other students' projects and any case of academic dishonesty will be referred to the University's Office of Student Conduct (https://www.studentconduct.umd.edu/).

Web Accessibility (https://www.umd.edu/web-accessibility/)