PA6 – Programming Workflow

Name: Natalie Young
UCSD email: nyoung@ucsd.edu

First 20 Minutes

Screenshot or copy/paste of program:

```
/**
 * Programming Assignment 6
 * https://ucsd-cse11-f21.github.io/assignments/pa6.html
 *
 * @author Natalie Young
 * @since 2021-11-09
 */

class AveragePositives
{
	/**
	 * Prints average of command-line args strictly greater than 0
	 * when parsed as doubles; produces 0 if no greater numbers
	 */
	public static void main(String[] args)
	{
		double sum = 0.0;
		int totalArgs = 0;
		double strValue = 0.0;
		double mean = 0.0;

		for (String arg : args)
		{
			strValue = Double.valueOf(arg);

			if (strValue > 0.0)
			{
				sum += strValue;

				totalArgs += 1;
```

```
                                mean = sum / totalArgs;
                            }
                        }

                        System.out.println(mean);
                }
}

/**
 * Programming Assignment 6
 * https://ucsd-cse11-f21.github.io/assignments/pa6.html
 *
 * @author Natalie Young
 * @since 2021-11-09
 */

class Pair
{}
```

Screenshot or copy/paste of ./run or java/javac output (if any):

```
nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ javac AveragePositives.java

nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ java AveragePositives 1 2
1.5

nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ java AveragePositives -1 2
2.0

nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ java AveragePositives 1 2 0
1.5

nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ java AveragePositives -1 -2
0.0

nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
```

```
$ java AveragePositives 1.33333 2.66666
1.9999949999999997

nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ java AveragePositives 0 0 0
0.0
```

Thoughts on your progress:

I worked on this task with a top-down approach: I created the files, added the file info in comments at the beginning, added the classes, and added method stubs. After that, I filled AveragePositives in little by little until I thought it was ready for compiling. I spent a little more time than I could have on the AveragePositives task because I didn't carefully look over my code and had to tweak parts of it whenever errors or unexpected results came up.

Distractions:

Yes, 2 minutes

Second 20 minutes:

Screenshot or copy/paste of program:

```
/**
 * Programming Assignment 6
 * https://ucsd-cse11-f21.github.io/assignments/pa6.html
 *
 * @author Natalie Young
 * @since 2021-11-09
 */

import tester.*;

class Pair
{
	int a;
	int b;

	Pair(int a, int b)
	{
```

```
                this.a = a;
                this.b = b;
        }
}

class PairSelect
{
        static int[] getAs(Pair[] pairArray)
        {
                int[] As = new int[pairArray.length];

                for(int i = 0; i < pairArray.length; i++)
                {
                        As[i] = pairArray[i].a;
                }

                return As;
        }
}
```

Screenshot or copy/paste of ./run or java/javac output (if any):

n/a

Thoughts on your progress:

As with the other task, I went with a top-down approach to completing the program. I forgot to leave an informative comment above the getAs method since I got distracted. I had to pay close attention to the data types of the methods and variables; I almost set a variable for an array of pairs equal to a pair. I ran into some issues with the classpath, which took some adjustments to resolve. I didn't get very much done due to environmental distractions.

Distractions:

Yes, 18 minutes.

Final 20 minutes:

Screenshot or copy/paste of program:

```
/**
 * Programming Assignment 6
 * https://ucsd-cse11-f21.github.io/assignments/pa6.html
 *
 * @author Natalie Young
 * @since 2021-11-09
 */

import tester.*;

class Pair
{
	int a;
	int b;

	Pair(int a, int b)
	{
		this.a = a;
		this.b = b;
	}
}

class PairSelect
{
	static int[] getAs(Pair[] pairArray)
	{
		int[] As = new int[pairArray.length];

		for(int i = 0; i < pairArray.length; i++)
		{
			As[i] = pairArray[i].a;
		}

		return As;
	}
```

```
}

class ExamplePairs
{
	Pair pair1 = new Pair(0,3);
	Pair pair2 = new Pair(1,2);
	Pair pair3 = new Pair(2,1);
	Pair pair4 = new Pair(3,0);

	Pair[] pairArray1 = {pair1, pair2, pair3, pair4};
	Pair[] pairArray2 = {pair2, pair2};
	Pair[] pairArray3 = {pair3};
	Pair[] pairArray4 = {};

	int[] expected1 = {0, 1, 2, 3};
	int[] expected2 = {1, 1};
	int[] expected3 = {2};
	int[] expected4 = {};

	void testGetAs(Tester t)
	{
		t.checkExpect(PairSelect.getAs(pairArray1), expected1);
		t.checkExpect(PairSelect.getAs(pairArray2), expected2);
		t.checkExpect(PairSelect.getAs(pairArray3), expected3);
		t.checkExpect(PairSelect.getAs(pairArray4), expected4);
	}
}
```

Screenshot or copy/paste of ./run or java/javac output (if any):

```
nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ export CLASSPATH=./tester.jar

nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ echo $CLASSPATH
./tester.jar

nyoung@LAPTOP-SV5IGG6N ~/CSE11/cse11-pa6-starter
$ ./run.bat ExamplePairs
error: file not found: ExamplePairs.java
```

```
Usage: javac <options> <source files>
use --help for a list of possible options
Tester Library v.3.0
-----------------------------------
Tests defined in the class: ExamplePairs:
--------------------------
ExamplePairs:
--------------
new ExamplePairs:1(
 this.pair1 = new Pair:2(
  this.a = 0
  this.b = 3)
 this.pair2 = new Pair:3(
  this.a = 1
  this.b = 2)
 this.pair3 = new Pair:4(
  this.a = 2
  this.b = 1)
 this.pair4 = new Pair:5(
  this.a = 3
  this.b = 0)
 this.pairArray1 = new Pair[4]:6{
  [0] Pair:2,
  [1] Pair:3,
  [2] Pair:4,
  [3] Pair:5
 }
 this.pairArray2 = new Pair[2]:7{
  [0] Pair:3,
  [1] Pair:3
 }
 this.pairArray3 = new Pair[1]:8{
  [0] Pair:4
 }
 this.pairArray4 = new Pair[0]:9{
 }
 this.expected1 = new int[4]:10{
  [0] 0,
  [1] 1,
  [2] 2,
  [3] 3
 }
 this.expected2 = new int[2]:11{
```

```
 [0] 1,
 [1] 1
}
this.expected3 = new int[1]:12{
 [0] 2
}
this.expected4 = new int[0]:13{
})
---------------

Ran 4 tests.
All tests passed.

--- END OF TEST RESULTS ---
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by tester.Main
(file:/C:/Users/nyoung/UCSD/CSE11/cse11-pa6-starter/tester.jar)
WARNING: Please consider reporting this to the maintainers of tester.Main
WARNING: System::setSecurityManager will be removed in a future release
```

Thoughts on your progress:

I managed to get a lot done in a relatively short period of time once I made sure there weren't any distractions around me. I ran into some issues while trying to compile (like static/non-static referencing), but once I managed to compile, the tests passed on the first run.

Distractions:

2 minutes

Overall Reflection

I found that it was difficult to keep track of data types and to troubleshoot issues when there were distractions around me, but this is generally what I spend the most time on anyways when programming, regardless of distractions. I think my top-down approach to handling programming tasks is relatively efficient since I do most of the planning before I start to actually write code; this rarely results in issues with data types. I also tend to print values of variables before and after they are accessed to make sure they contain expected values. One thing I would like to work on is keeping track of time, since these tasks took me longer than they should have taken to complete.

This process reflects how I normally complete programming assignments with the exception that I had the choice of which tasks to complete--obviously, I picked the simpler ones for this assignment. From this experience, I learned that my programming workflow is generally robust against program errors but is susceptible to distraction in the planning stages.