Hurricane Project – Array and `ArrayList` Data Structure
Focus: Calculating Basic Statistics, Sorting, and Searching

Name _____ Total _____/125

Date _____ Period _____

Assessor _____

Hurricane Class

_____ Comments include name, date, and summary using Javadoc and uses correct grammar.
_____ Instance variables – declared to be private and variable names are spelled out
_____ Constructor with correct parameters and it calls `determineCategory`
_____ `getName`, `getPressure`, `getSpeed`, `getYear`, `getMonth`, `getCategory`
_____ `determineCategory`
_____ `toString` – uses `String.format` method
_____ `compareYearTo`, `compareNameTo`, `comparePressureTo`, `compareSpeedTo`, `compareCategoryTo` (each are one-line methods)
_____ CheckStyle finds no errors. Javadoc is correctly done for all methods and constructors. These include:

- Summary
- `@param`, if appropriate
- `@return`, if appropriate

HurricaneOrganizer Class with an Array of Hurricane Objects

_____ Author's unique name replaces "`Name`" in class name, e.g., `HurricaneOrganizerArrayKing`
_____ Comments include name, date, and summary using Javadoc and uses correct grammar.
_____ Instance variables – array of hurricane objects
_____ `findMaxPressure` without sorting first
_____ `findMaxWindspeed` without sorting first
_____ `findMinPressure` without sorting first
_____ `findMinWindSpeed` without sorting first
_____ `calculateAveragePressure`
_____ `calculateAverageWindSpeed`
_____ `calculateAverageCategory`
_____ `sortYears` – selection sort that sorts the years ascending and uses `compareYearTo`
_____ `sortNames` – insertion sort that sorts the names ascending and uses `compareNameTo`
_____ `sortCategories` – selection sort that sorts the categories descending and uses `compareCategoryTo`
_____ `sortPressures` – merge sort (not recursive) that sorts the pressures descending and uses `comparePressureTo`. Verify that `sortPressures'` helper method sorts a portion of the array.
_____ `sortWindSpeeds` – recursive merge that sorts ascending (base case is labeled)
_____ `mergeWindSpeeds` – merge to support `sortWindSpeeds` and uses `compareSpeedTo`
_____ `searchYear` – sequential search (array of hurricanes that occurred for a given year) (2 loops)
_____ `searchHurricaneNameHelper` – binary search (array of hurricanes with a given name) (2 base cases are labeled)
Verify that `searchHurricaneNameHelper's` retrieval helper method finds all occurrences of hurricanes with a given name.
_____ CheckStyle finds no errors. Javadoc is correctly done for all methods and constructors. These include:

- Summary
- `@param`, if appropriate (if the parameter is an index, the comment states if index is included or excluded)
- `@return`, if appropriate

Hurricane Project – Array and `ArrayList` Data Structure
Focus: Calculating Basic Statistics, Sorting, and Searching

HurricaneOrganizer Class with an `ArrayList` of Hurricane Objects

_____ Verify there is no **[** except on the line `public static void main(String[]args)`

_____ Comments include name, date, and summary using Javadoc and uses correct grammar.
Comments with the word array have been changed to use `ArrayList`.

_____ Verify in the `main` method that the object `cane` is a "`HurricaneOrganizerArrayList`Name" object.

_____ Instance variables – `ArrayList` of hurricane objects

_____ `determineFileLength` method is not in the class

_____ `findMaxPressure` without sorting first

_____ `findMaxWindspeed` without sorting first

_____ `findMinPressure` without sorting first

_____ `findMinWindSpeed` without sorting first

_____ `calculateAveragePressure`

_____ `calculateAverageWindSpeed`

_____ `calculateAverageCategory`

_____ `sortYears` – selection sort that sorts the years ascending and uses `compareYearTo`

_____ `sortNames` – insertion sort that sorts the names ascending and uses `compareNameTo`

_____ `sortCategories` – selection sort that sorts the categories descending and uses `compareCategoryTo`

_____ `sortPressures` – merge sort (not recursive) that sorts the pressures descending and uses `comparePressureTo.` Verify that `sortPressures'` helper method sorts a portion of the ArrayList.

_____ `sortWindSpeeds` – recursive merge that sorts ascending (base case is labeled)

_____ `mergeWindSpeeds/merge` – merge to support `sortWindSpeeds` and uses `compareSpeedTo`

_____ `searchYear` – sequential search (`ArrayList` of hurricanes that occurred for a given year) (1 loop)

_____ `searchHurricaneNameHelper` – binary search (`ArrayList` of hurricanes with a given name) (2 base cases are labeled)
Verify that `searchHurricaneNameHelper's` retrieval helper method finds all occurrences of hurricanes with a given name.

_____ CheckStyle finds no errors. Javadoc is correctly done for all methods and constructors. These include:

- Summary
- `@param`, if appropriate (if the parameter is an index, the comment states if index is included or excluded)
- `@return`, if appropriate
-

Hurricane Testing

Array	`ArrayList`	Test	Array	`ArrayList`	Test (The number of matches is in the parentheses.)
		Test the sorts in the following order:			Search for names:
		• Sort wind speed (ascending)			• Alex (1)
		• Sort names (ascending)			• Alicia (1)
		• Sort categories (descending)			• AAA (0)
		• Sort years (ascending)			• Wilma (1)
		• Sort pressures (descending)			• Bob (2)
					• Charley (2)
		Test averages:			• Josh (0)
		• Wind `90.86`			• Vince (1)
		• Pressure `961.76`			• Fran (1)
		• Category `2.12`			• Frances (1)
					• ZZZ (0)
		Test minimums and maximums:			Search for years:
		• Speed min `64`			• 2006 (5)
		• Speed max `150`			• 1980 (1)
		• Pressure min `882`			• 3001 (0)
		• Pressure max `1002`			• 1960 (0)