

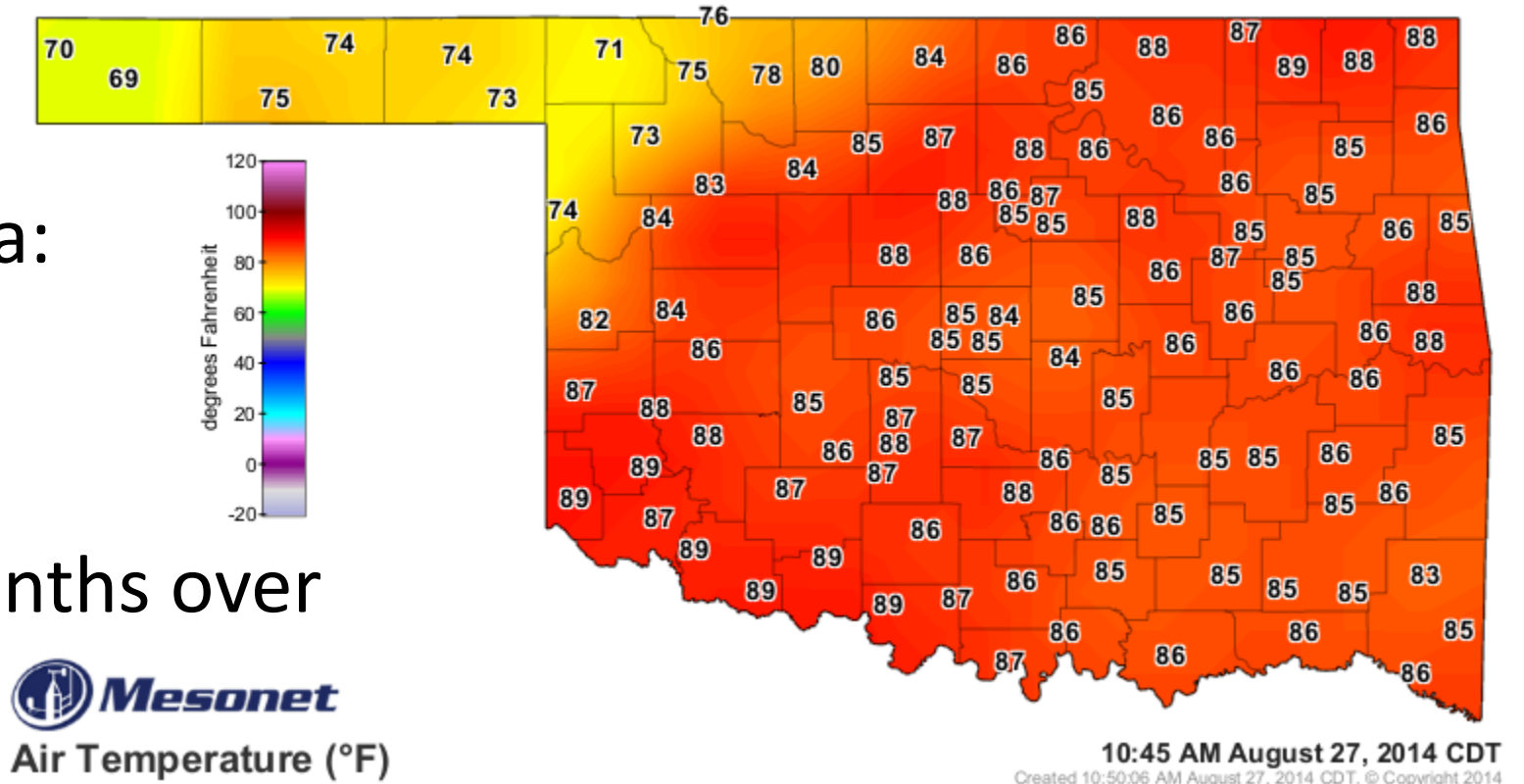
CS 2334: Project 2

Class Abstraction

Project 2

Expanded Mesonet data:

- Wind data
- Additional stations
- Larger data set: all months over twelve years
- More invalid data



High-Level Task: User Selects Data to be Summarized

Through System.in:

- User selects station (one specific one or “all”)
- User selects years to summarize (one specific one or “all”)

- Your program loads in the data set and displays the high-level statistics

- Demonstration

Objectives

- Make an interactive menu for a user and handle errors in
- Automatically load a set of files in a directory (folder)
- Create and use abstract objects and interfaces in appropriate ways
- Make use of polymorphism in code
- Continue to exercise good coding practices for Javadoc and for unit testing

Recall Project 1...

- MonthlyData computed statistics over all days in a month
- DataSet computed statistics over all months in a data set
- Separate, but very similar statistics code...

Project 2

- For this project, we have several “notions” of higher level statistics: months, years and entire data sets
- We want to be able to write our statistics computation code once for all of these
- Class hierarchies will make this “easy”

Observation Class

Observation
-value:double -valid:boolean
+ Observation() + Observation(value:double) + getValue():double + getValid():double + isLessThan(o: Observation):boolean + isGreaterThan(o: Observation):boolean + toString():String

isLessThan() example on board...

StatisticsAbstract

Any class about which statistics can be computed

- Defines some properties and getters that are common to all subclasses
- Defines a set of abstract methods that also must be in common (but can't provide an implementation of)

<i>StatisticsAbstract</i>
<pre>#temperatureMin:Observation #temperatureMax:Observation #temperatureAverage:Observation #windMin:Observation #windMax:Observation #windAverage:Observation #windChillMin:Observation #heatIndexMax:Observation</pre>
<pre>-getTemperatureMinDay():DailyData -getTemperatureMaxDay():DailyData -getRainMin():Observation -getRainMinDay():DailyData -getRainMax():Observation -getRainMaxDay():DailyData -getRainAverage():Observation -getWindMinDay():DailyData -getWindMaxDay():DailyData -getWindChillMinDay():DailyData -getHeatIndexMaxDay():DailyData +OTHER GETTERS</pre>

Daily Data

Extends StatisticsAbstract

- Adds a small number of additional properties
- Implements a large set of abstract methods

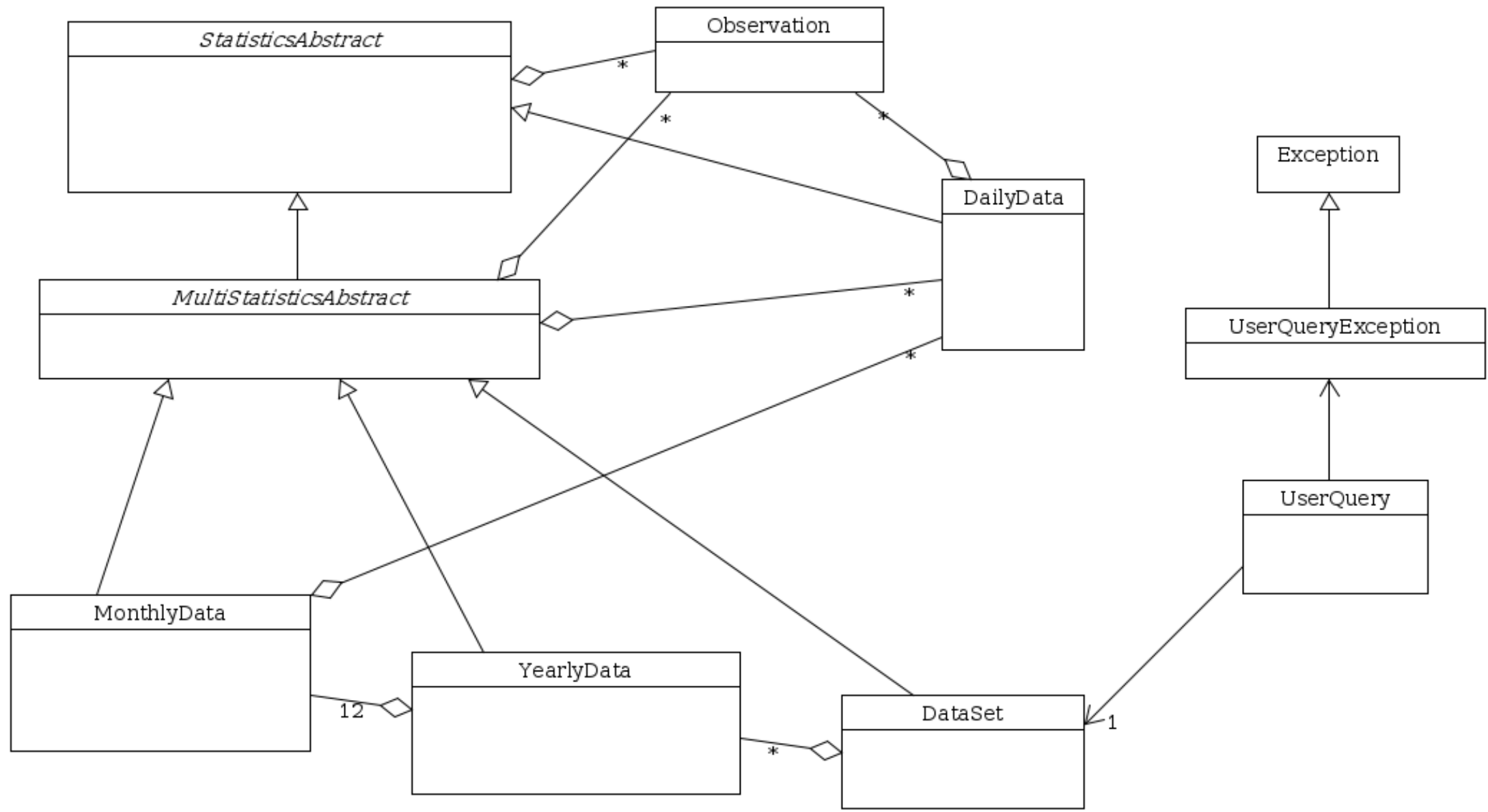
DailyData
-year:int -month:int -day:int -stationId:String -rainFall:Observation
+ DailyData(year:int, month:int, stationId:String, temperatureMax:Observation, temperatureMin:Observation, temperatureAverage:Observation, windMax:Observation, windMin:Observation, windAverage:Observation, rainFall:Observation, heatIndexMax:Observation, windChillMin:Observation) + getDate():String + getTemperatureMinDay():DailyData + getTemperatureMaxDay():DailyData + getRainMin():Observation + getRainMinDay():DailyData + getRainMax():Observation + getRainMaxDay():DailyData + getRainAverage():Observation + getWindMinDay():DailyData + getWindMaxDay():DailyData + getWindChillMinDay():DailyData + getHeatIndexMaxDay():DailyData + OTHER GETTERS

MultiStatisticsAbstract

Extends StatisticsAbstract

- Describes any class that computes statistics about multiples of another class
- Defines a common set of properties and methods
- All statistics computations defined here

<i>MultiStatisticsAbstract</i>
-temperatureMinDay:DailyData -temperatureMaxDay:DailyData -rainMin:Observation -rainMinDay:DailyData -rainMax:Observation -rainMaxDay:DailyData -rainAverage:Observation -windMinDay:DailyData -windMaxDay:DailyData -windChillMinDay:DailyData -heatIndexMaxDay:DailyData
+getTemperatureMinDay():DailyData +getTemperatureMaxDay():DailyData +getRainMin():Observation +getRainMinDay():DailyData +getRainMax():Observation +getRainMaxDay():DailyData +getRainAverage():Observation +getWindMinDay():DailyData +getWindMaxDay():DailyData +getWindChillMinDay():DailyData +getHeatIndexMaxDay():DailyData +computeStats(list:ArrayList<? extends StatisticsAbstract>):void +toString():String -computeRainStats(list:ArrayList<? extends StatisticsAbstract>):void -computeTemperatureStats(list:ArrayList<? extends StatisticsAbstract>):void -computeWindStats(list:ArrayList<? extends StatisticsAbstract>):void OTHER GETTERS



Notes

- For a single day's Observations, some may be valid while others are invalid (this was true in project 1, also)
- For a given Observation type (e.g., windMax), it is possible for an entire month to have invalid data. So, a month's windMax must also be an Observation (not a double)
 - This is addressed in our definition of StatisticsAbstract (but you need to provide implementation)

Deadlines

- Project must be submitted by Wednesday, Oct 14th @1:29pm
- Code review must be completed by Friday, Oct 23rd