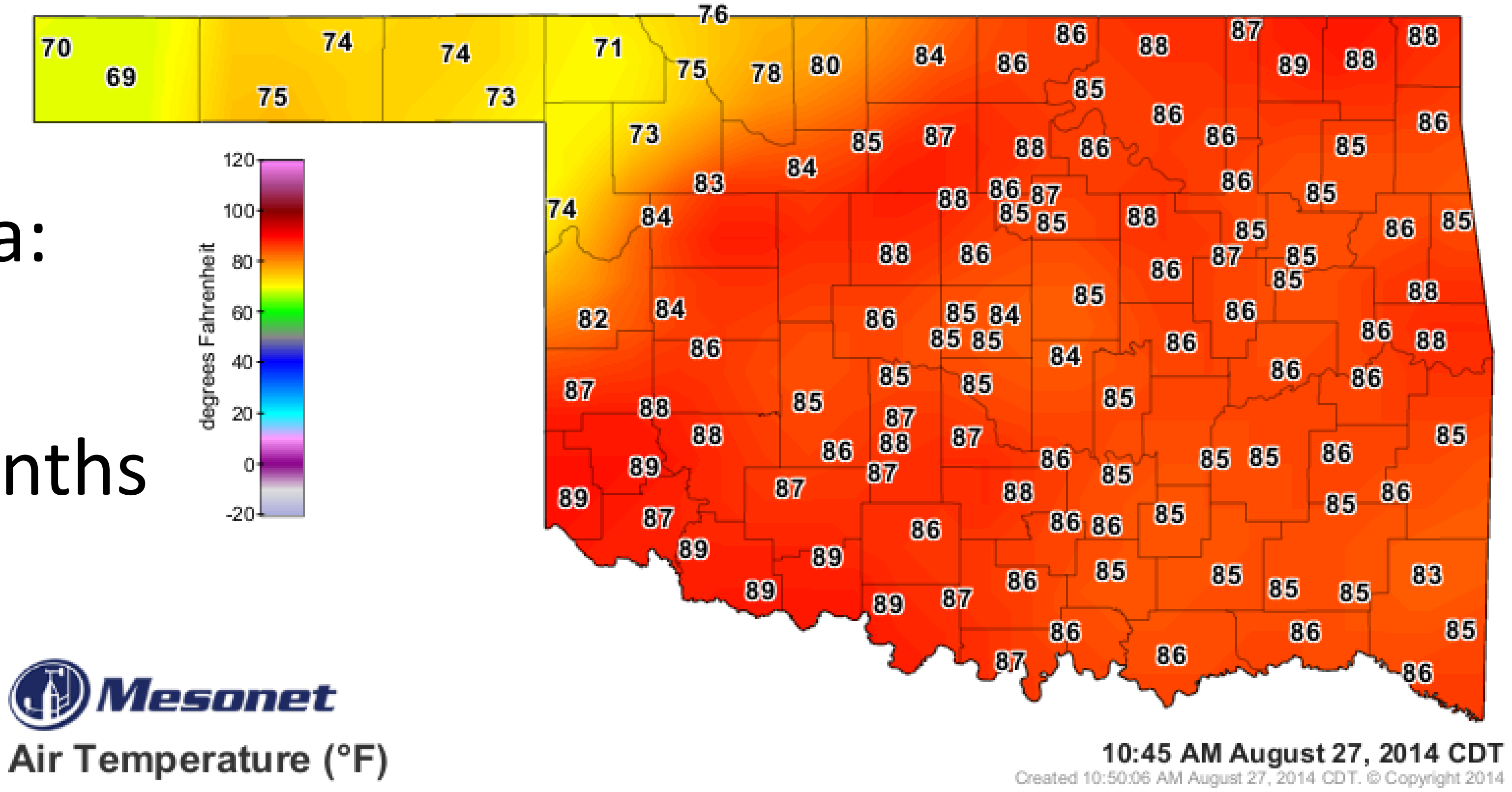
CS 2334: Project 2
Class Abstraction

Project 2

Expanded Mesonet data:

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



Mesonet
Air Temperature (°F)
degrees Fahrenheit
10:45 AM August 27, 2014 CDT
Created 10:50:06 AM August 27, 2014 CDT. © Copyright 2014

Recall Project 1…

- DataDay represented a tuple of Samples for a single day
- DataMonth computed statistics over all days in a month

Project 2

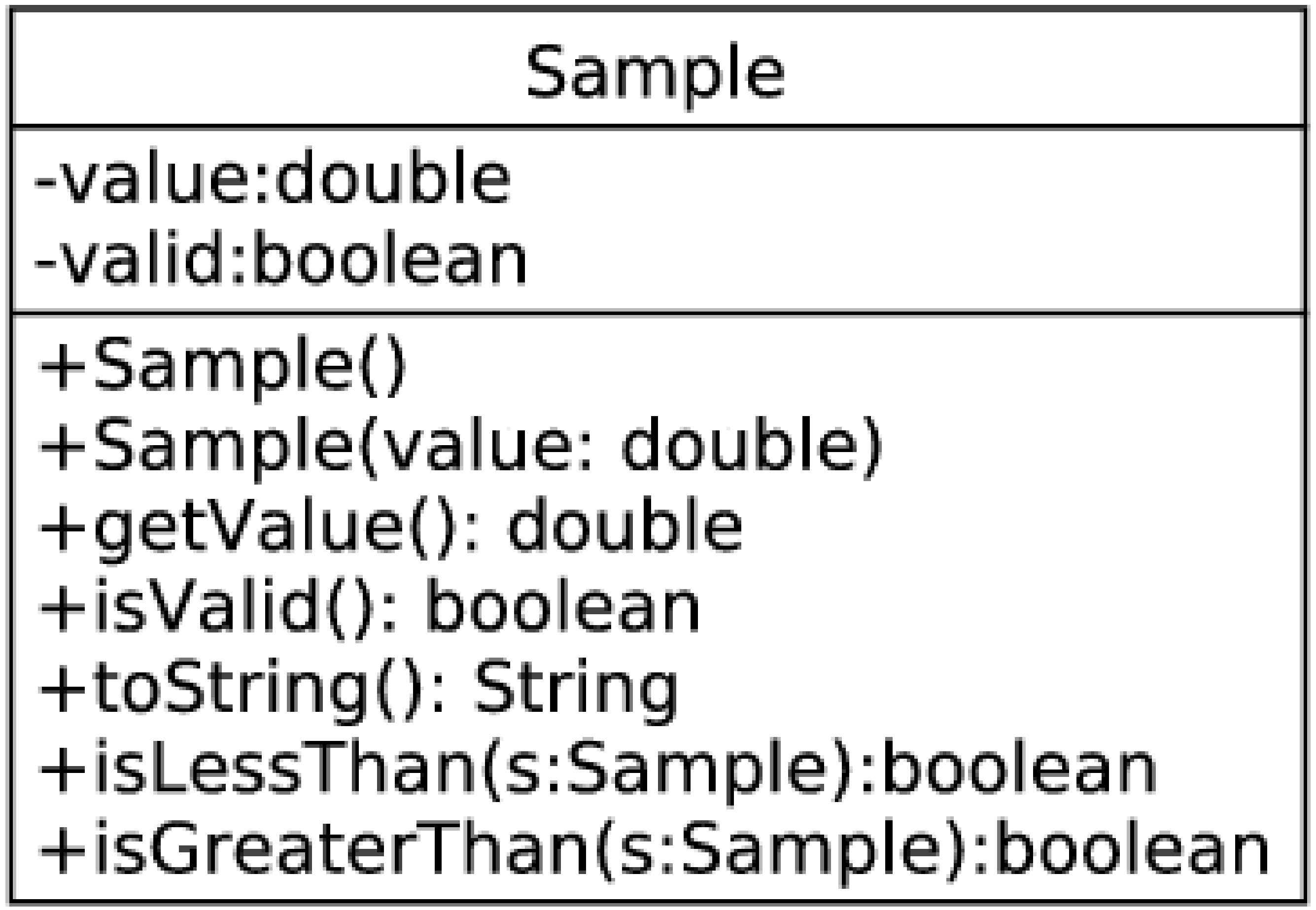
- For this project, we have several “notions” of higher level statistics: months, years and entire data sets
- We want to be able to write our statistics computation code once for all of these and not have to repeat our implementation at these different levels

- Class hierarchies will make this easy

Objectives

- 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

Sample Class



Sample

-value:double
-valid:boolean

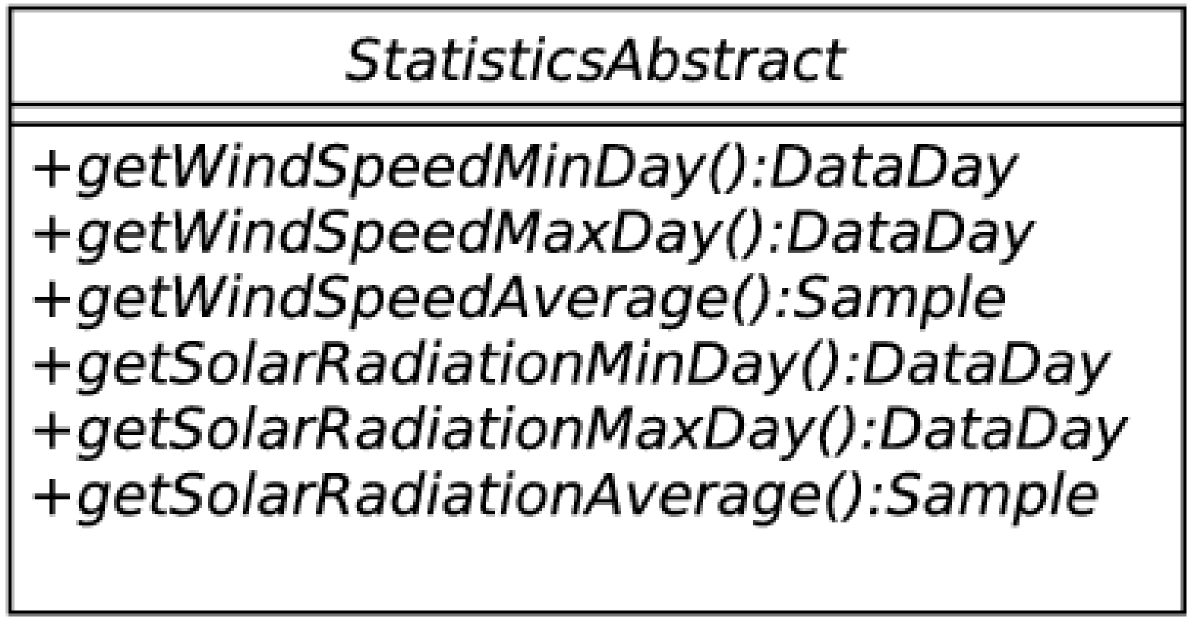
+Sample()
+Sample(value: double)
+getValue(): double
+isValid(): boolean
+toString(): String
+isLessThan(s:Sample):boolean
+isGreaterThan(s:Sample):boolean

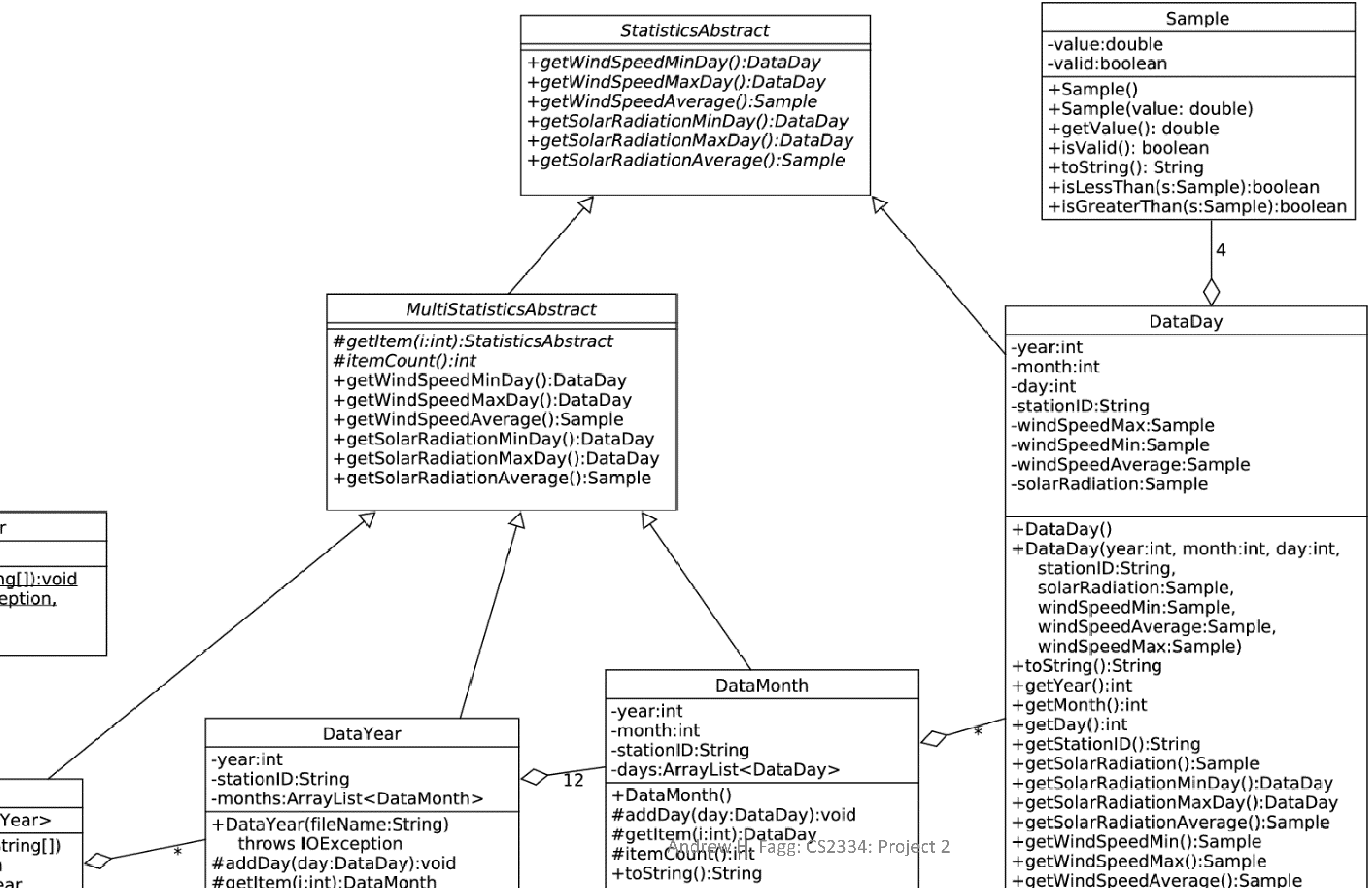
isLessThan() example on board…

StatisticsAbstract

Any class about which statistics can be computed

- Defines a set of abstract methods that are common to all subclasses

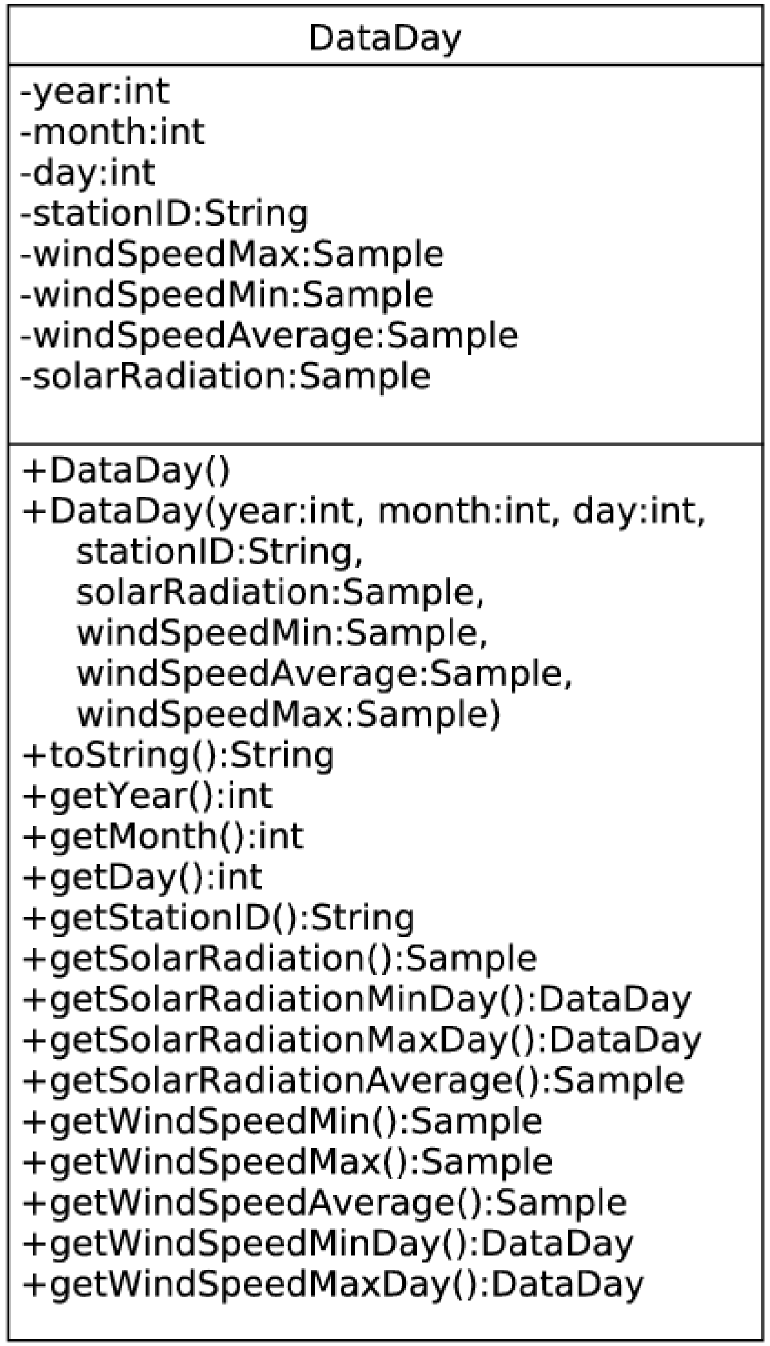

StatisticsAbstract
+*getWindSpeedMinDay():DataDay*
+*getWindSpeedMaxDay():DataDay*
+*getWindSpeedAverage():Sample*
+*getSolarRadiationMinDay():DataDay*
+*getSolarRadiationMaxDay():DataDay*
+*getSolarRadiationAverage():Sample*


StatisticsAbstract
+getWindSpeedMinDay():DataDay
+getWindSpeedMaxDay():DataDay
+getWindSpeedAverage():Sample
+getSolarRadiationMinDay():DataDay
+getSolarRadiationMaxDay():DataDay
+getSolarRadiationAverage():Sample
Sample
-value:double
-valid:boolean
+Sample()
+Sample(value: double)
+getValue(): double
+isValid(): boolean
+toString(): String
+isLessThan(s:Sample):boolean
+isGreaterThan(s:Sample):boolean
4
MultiStatisticsAbstract
#getItem(i:int):StatisticsAbstract
#itemCount():int
+getWindSpeedMinDay():DataDay
+getWindSpeedMaxDay():DataDay
+getWindSpeedAverage():Sample
+getSolarRadiationMinDay():DataDay
+getSolarRadiationMaxDay():DataDay
+getSolarRadiationAverage():Sample
DataDay
-year:int
-month:int
-day:int
-stationID:String
-windSpeedMax:Sample
-windSpeedMin:Sample
-windSpeedAverage:Sample
-solarRadiation:Sample
+DataDay()
+DataDay(year:int, month:int, day:int, stationID:String, solarRadiation:Sample, windSpeedMin:Sample, windSpeedAverage:Sample, windSpeedMax:Sample)
+toString():String
+getYear():int
+getMonth():int
+getDay():int
+getStationID():String
+getSolarRadiation():Sample
+getSolarRadiationMinDay():DataDay
+getSolarRadiationMaxDay():DataDay
+getSolarRadiationAverage():Sample
+getWindSpeedMin():Sample
+getWindSpeedMax():Sample
+getWindSpeedAverage():Sample
r
ng[]):void
eption,
DataMonth
-year:int
-month:int
-stationID:String
-days:ArrayList<DataDay>
+DataMonth()
#addDay(day:DataDay):void
#getItem(i:int):DataDay
#itemCount():int
+toString():String
*
DataYear
-year:int
-stationID:String
-months:ArrayList<DataMonth>
+DataYear(fileName:String) throws IOException
#addDay(day:DataDay):void
#getItem(i:int):DataMonth
12
Year>
tring[])
n
ear
*

DataDay

Extends StatisticsAbstract

- Properties and getters/setters identical to those of project 1
- Note new constructor: creates an invalid DataDay
- Implement abstract methods from StatisticsAbstract

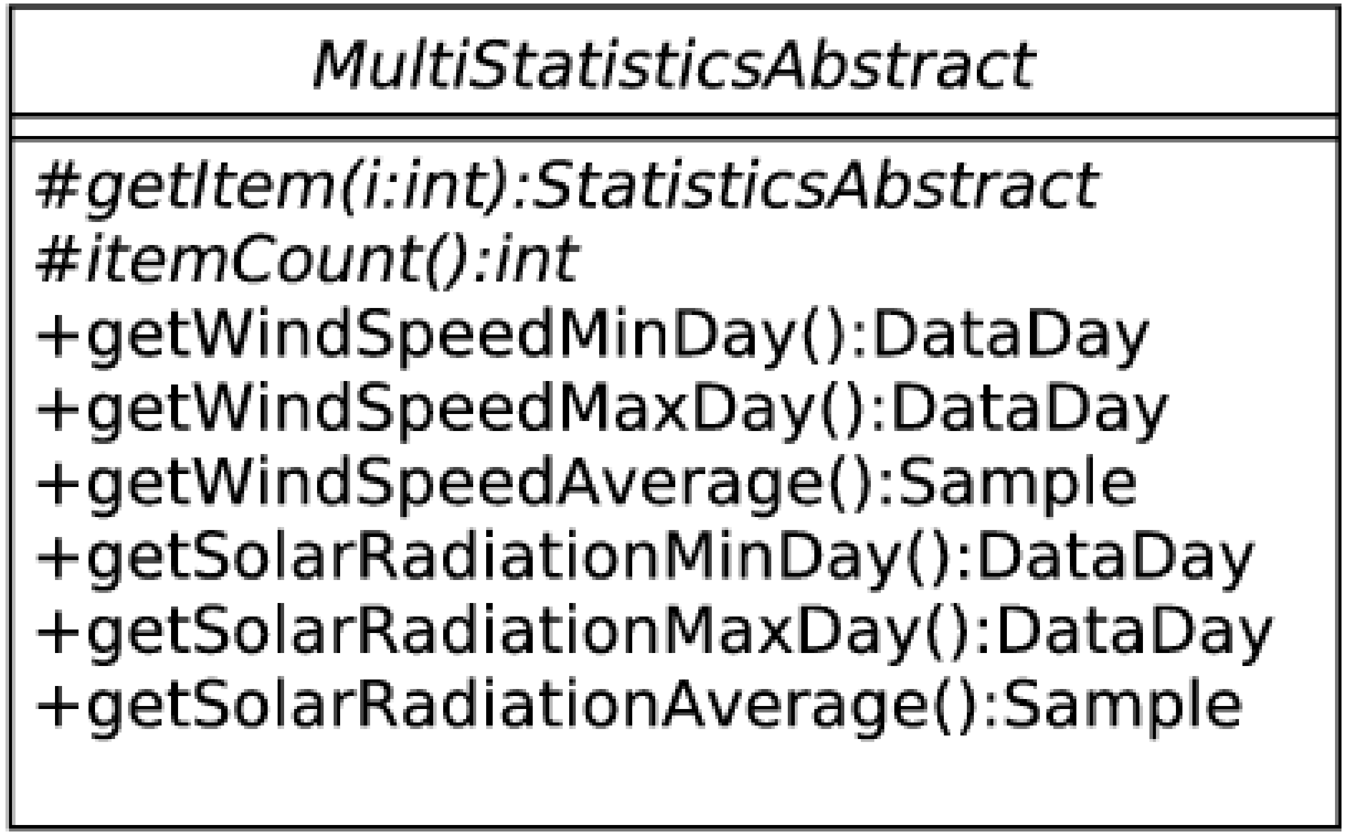


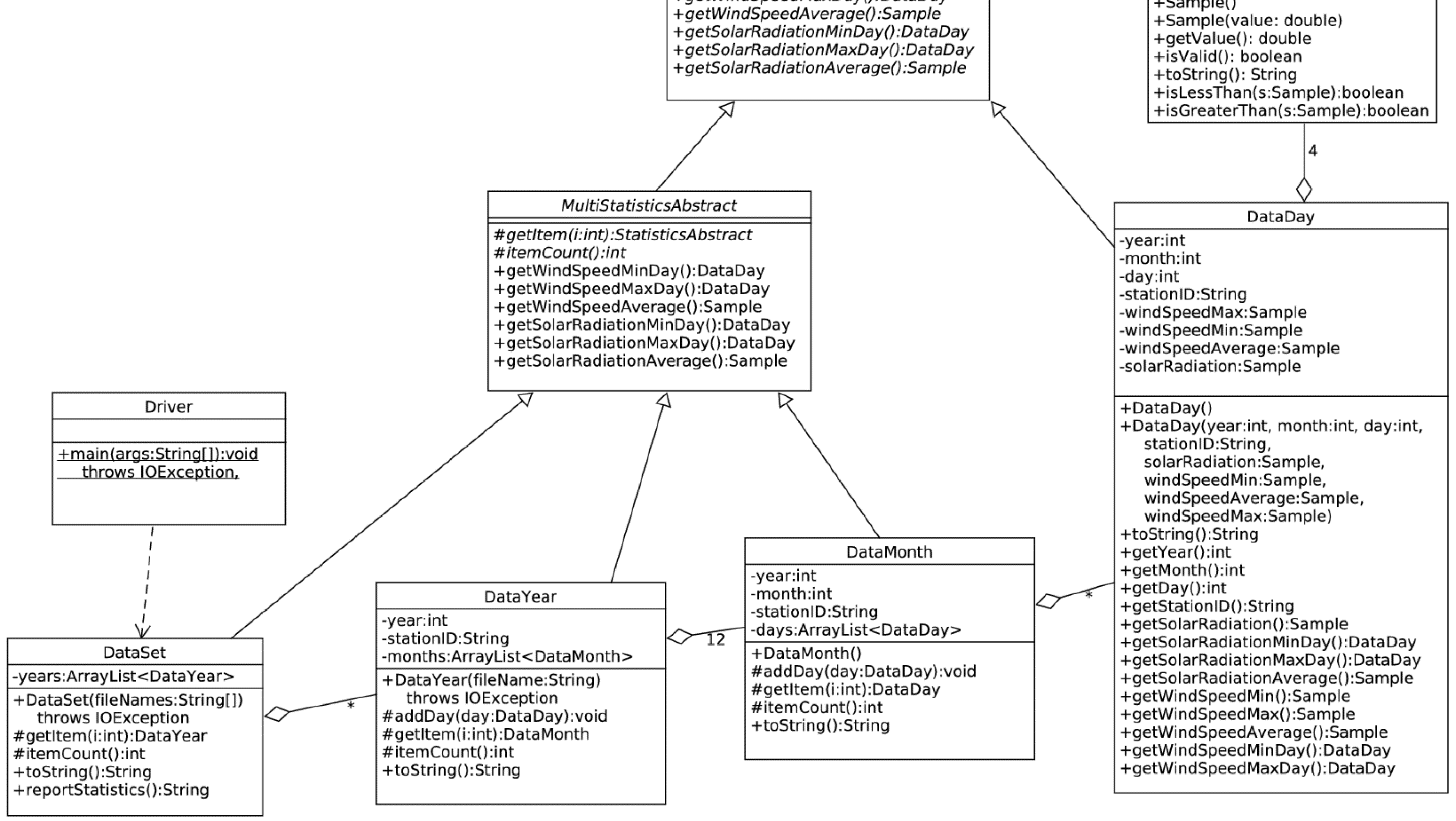
```
DataDay
-year:int
-month:int
-day:int
-stationID:String
-windSpeedMax:Sample
-windSpeedMin:Sample
-windSpeedAverage:Sample
-solarRadiation:Sample

+DataDay()
+DataDay(year:int, month:int, day:int,
    stationID:String,
    solarRadiation:Sample,
    windSpeedMin:Sample,
    windSpeedAverage:Sample,
    windSpeedMax:Sample)
+toString():String
+getYear():int
+getMonth():int
+getDay():int
+getStationID():String
+getSolarRadiation():Sample
+getSolarRadiationMinDay():DataDay
+getSolarRadiationMaxDay():DataDay
+getSolarRadiationAverage():Sample
+getWindSpeedMin():Sample
+getWindSpeedMax():Sample
+getWindSpeedAverage():Sample
+getWindSpeedMinDay():DataDay
+getWindSpeedMaxDay():DataDay
```

MultiStatisticsAbstract

Extends StatisticsAbstract

- Describes any class that computes statistics about multiples of a sub-object
- All statistics computations defined here
- Defines a common set of abstract methods


MultiStatisticsAbstract
#getItem(i:int):StatisticsAbstract
#itemCount():int
+getWindSpeedMinDay():DataDay
+getWindSpeedMaxDay():DataDay
+getWindSpeedAverage():Sample
+getSolarRadiationMinDay():DataDay
+getSolarRadiationMaxDay():DataDay
+getSolarRadiationAverage():Sample


+getWindSpeedAverage():Sample
+getSolarRadiationMinDay():DataDay
+getSolarRadiationMaxDay():DataDay
+getSolarRadiationAverage():Sample
+Sample()
+Sample(value: double)
+getValue(): double
+isValid(): boolean
+toString(): String
+isLessThan(s:Sample):boolean
+isGreaterThan(s:Sample):boolean
4
MultiStatisticsAbstract
#getItem(i:int):StatisticsAbstract
#itemCount():int
+getWindSpeedMinDay():DataDay
+getWindSpeedMaxDay():DataDay
+getWindSpeedAverage():Sample
+getSolarRadiationMinDay():DataDay
+getSolarRadiationMaxDay():DataDay
+getSolarRadiationAverage():Sample
DataDay
-year:int
-month:int
-day:int
-stationID:String
-windSpeedMax:Sample
-windSpeedMin:Sample
-windSpeedAverage:Sample
-solarRadiation:Sample
+DataDay()
+DataDay(year:int, month:int, day:int, stationID:String, solarRadiation:Sample, windSpeedMin:Sample, windSpeedAverage:Sample, windSpeedMax:Sample)
+toString():String
+getYear():int
+getMonth():int
+getDay():int
+getStationID():String
+getSolarRadiation():Sample
+getSolarRadiationMinDay():DataDay
+getSolarRadiationMaxDay():DataDay
+getSolarRadiationAverage():Sample
+getWindSpeedMin():Sample
+getWindSpeedMax():Sample
+getWindSpeedAverage():Sample
+getWindSpeedMinDay():DataDay
+getWindSpeedMaxDay():DataDay
Driver
+main(args:String[]):void throws IOException,
DataMonth
-year:int
-month:int
-stationID:String
-days:ArrayList<DataDay>
+DataMonth()
#addDay(day:DataDay):void
#getItem(i:int):DataDay
#itemCount():int
+toString():String
*
DataYear
-year:int
-stationID:String
-months:ArrayList<DataMonth>
+DataYear(fileName:String) throws IOException
#addDay(day:DataDay):void
#getItem(i:int):DataMonth
#itemCount():int
+toString():String
12
DataSet
-years:ArrayList<DataYear>
+DataSet(fileNames:String[]) throws IOException
#getItem(i:int):DataYear
#itemCount():int
+toString():String
+reportStatistics():String
*

Data Loading

- DataYear now responsible for loading individual files (the new csv files now contain a year’s worth of data)
- DataSet accepts as input an array of files, each containing a year

Month/Year/DataSet toString() output

```
2015-11, TISH: Wind = [0.0000, 8.9807, 28.8100], Solar Radiation = [0.9000, 8.8883, 15.2500]

2015, TISH: Wind = [0.0000, 7.8934, 40.5300], Solar Radiation = [0.4000, 15.7975, 30.3500]

Data Set: Wind = [0.0000, 8.2617, 40.5300], Solar Radiation = [0.4000, 16.3488, 30.6400]
```

DataSet.reportStatistics() Output

```
Max Wind Speed:
2015-12-27, TISH: Wind = [11.7300, 25.5100, 40.5300], Solar Radiation = 0.4000
Average Wind Speed:
8.2617
Min Wind Speed:
2013-06-08, TISH: Wind = [0.0000, 7.5200, 16.1300], Solar Radiation = 27.3400

Max Solar Radiation:
2013-06-02, TISH: Wind = [2.3800, 8.2800, 17.3600], Solar Radiation = 30.6400
Average Solar Radiation:
16.3488
Min Solar Radiation:
2015-12-27, TISH: Wind = [11.7300, 25.5100, 40.5300], Solar Radiation = 0.4000
```

Notes

- For a single day's Samples, some may be valid while others are invalid (this was true in project 1, also)
- For a given Sample type (e.g., windSpeedAverage), it is possible for an entire month to have invalid data
    - getWindSpeedAverage() for the month will return an invalid Sample in this case
- It is also possible for all days within a month to have invalid windSpeedMax Samples
    - getWindSpeedMaxDay() for the month will return an invalid DataDay object

Deadlines

- Project must be submitted by Wednesday, Oct 12th @1:29pm
- Code review must be completed by Wednesday, Oct 19th