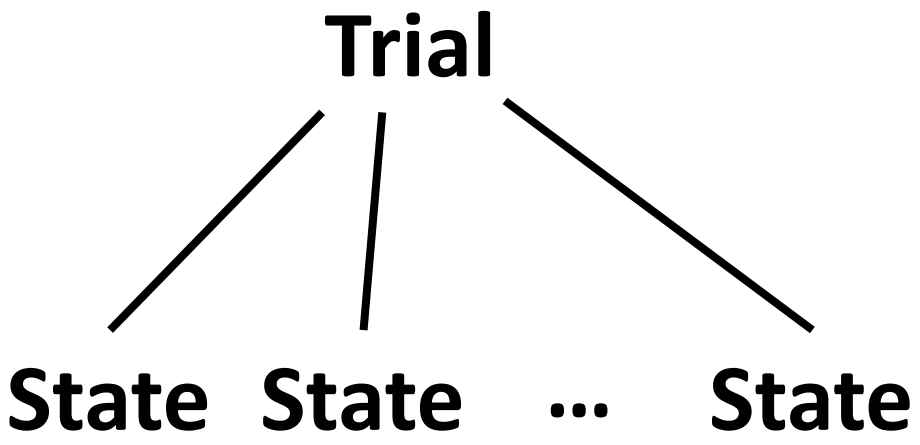
CS 2334: Project 2
Class Abstraction

Project 1 Design

A Trial has many States

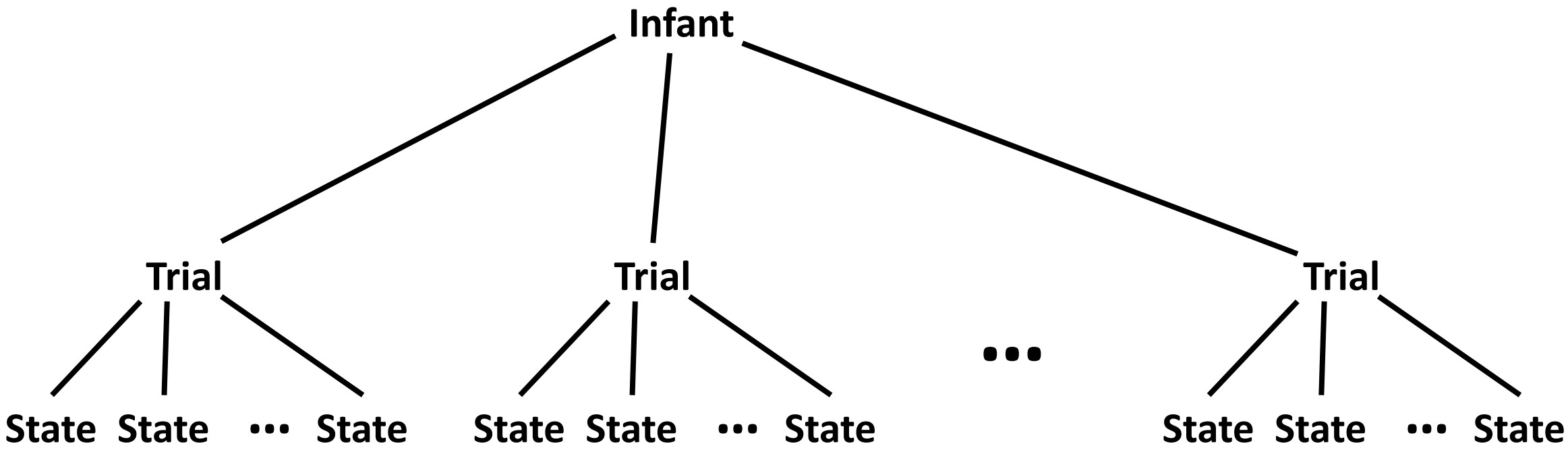
- A Trial knows how to compute statistics over its list of States



Project 2

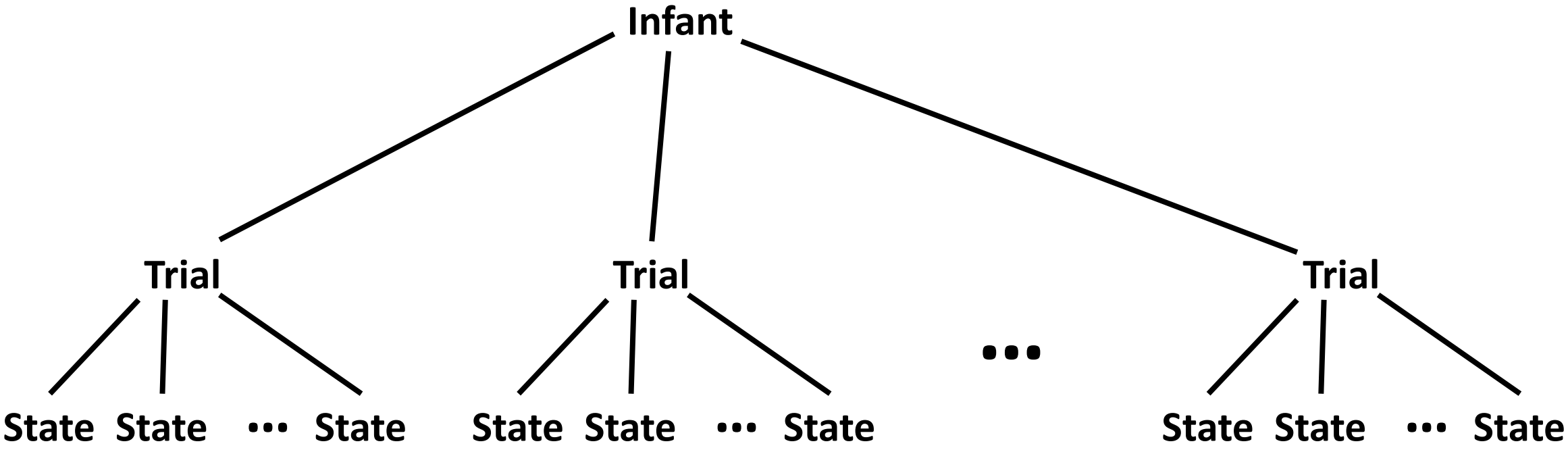
Now, we want to think about the list of Trials that belong to a single Infant …

Project 2


Infant
Trial
Trial
Trial
State
State
State
State
State
State
State
State
State

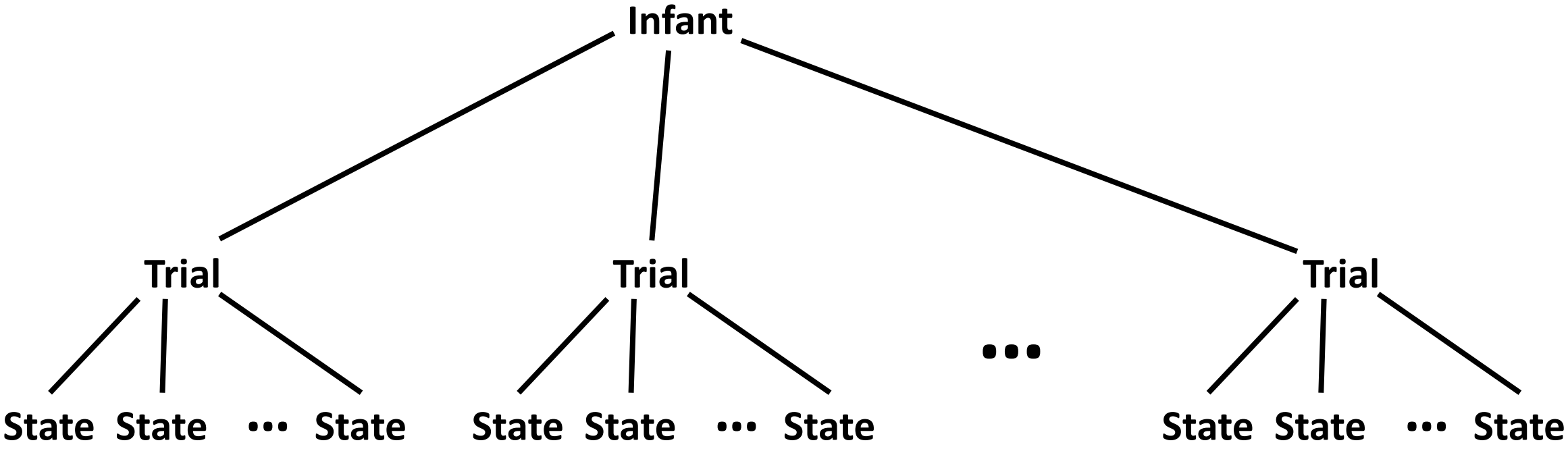
Project 2

In what way are Infants and Trials the same?



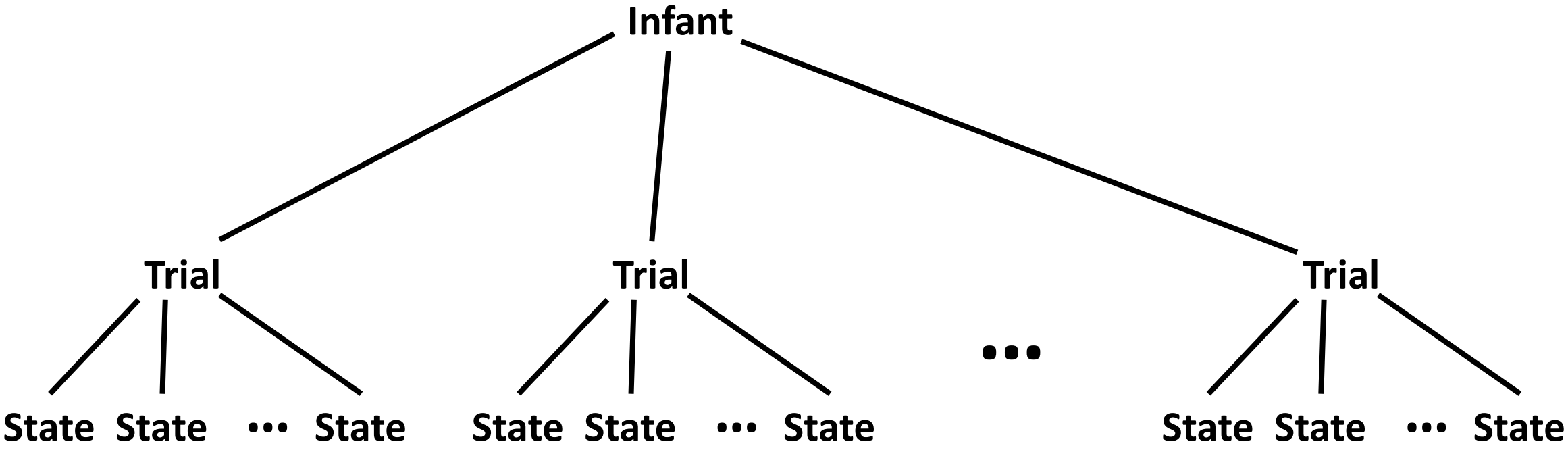
Project 2

Infants and Trials contain lists of items about which they can compute statistics!



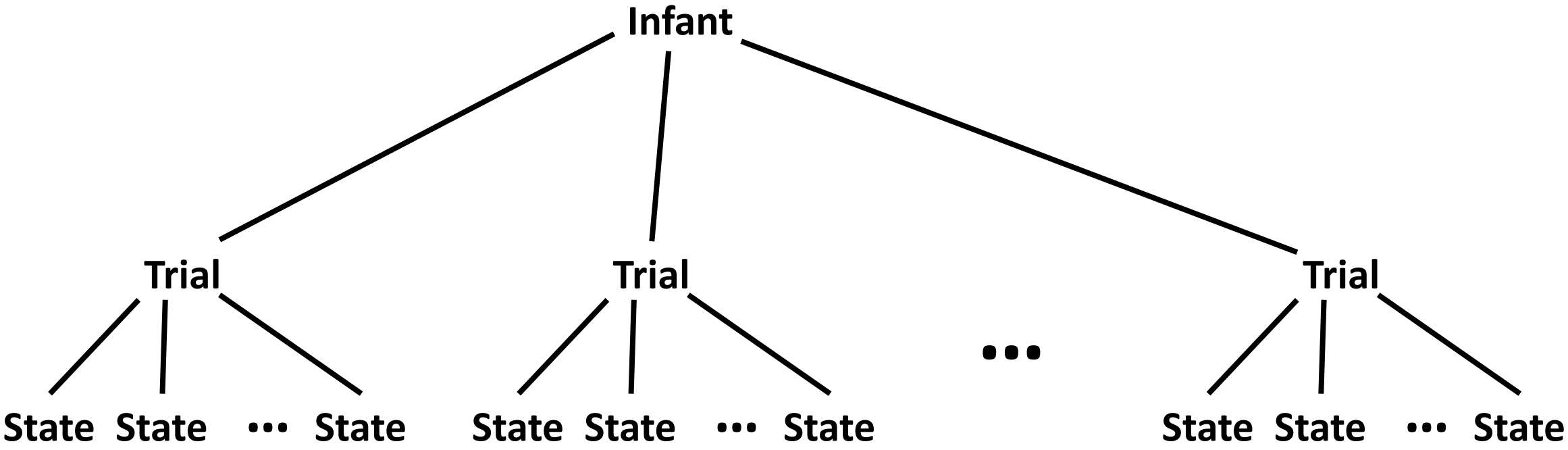
Project 2

In what way are Infants, Trials and States the same?



Project 2

Infants, Trials and States can have statistics computed ***about*** them!



Project Design

- SingleItemAbstract:
  - Classes about which statistics can be computed
  - Contain one or more States

- MultipleItemAbstract:
  - Classes that compute statistics
  - Contain a list of items (which implies more than one State)

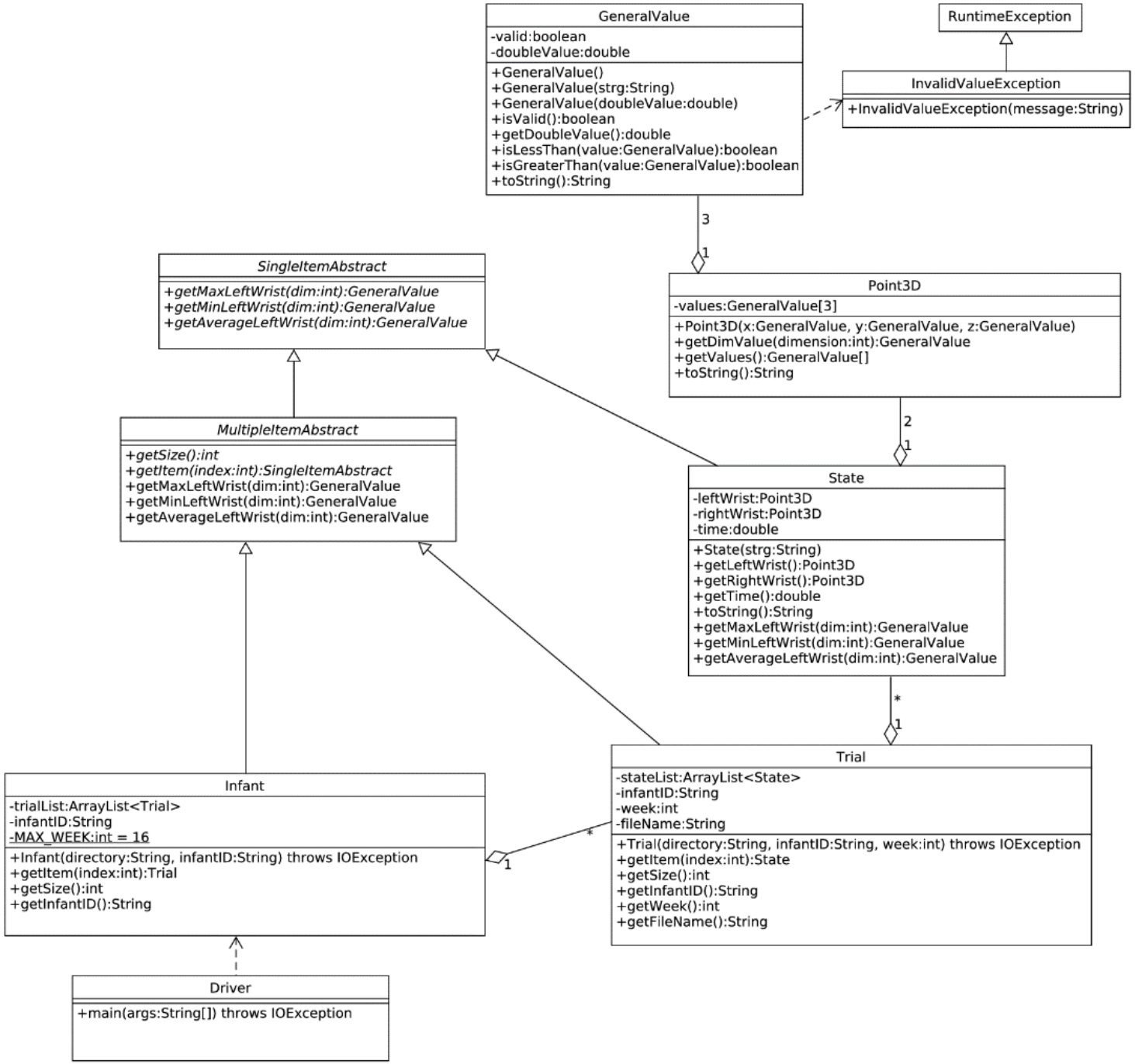
Project Design

- SingleItemAbstract:
  - Classes about which statistics can be computed
  - Contain one or more States

Infant
Trial
State

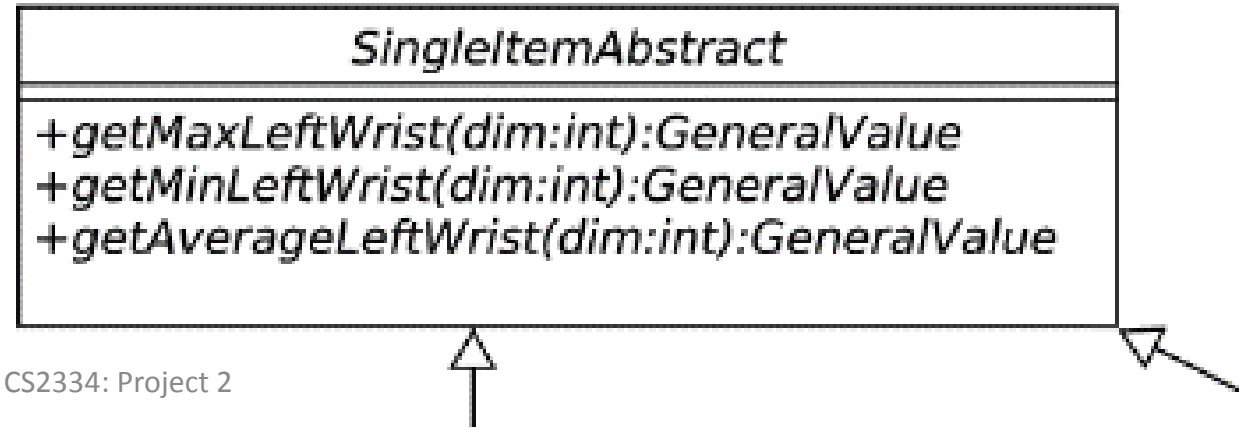
- MultipleItemAbstract:
  - Classes that compute statistics
  - Contain a list of items (which implies more than one State)

Infant
Trial


GeneralValue
-valid:boolean
-doubleValue:double
+GeneralValue()
+GeneralValue(strg:String)
+GeneralValue(doubleValue:double)
+isValid():boolean
+getDoubleValue():double
+isLessThan(value:GeneralValue):boolean
+isGreaterThan(value:GeneralValue):boolean
+toString():String
RuntimeException
InvalidValueException
+InvalidValueException(message:String)
3
1
SingleItemAbstract
+getMaxLeftWrist(dim:int):GeneralValue
+getMinLeftWrist(dim:int):GeneralValue
+getAverageLeftWrist(dim:int):GeneralValue
Point3D
-values:GeneralValue[3]
+Point3D(x:GeneralValue, y:GeneralValue, z:GeneralValue)
+getDimValue(dimension:int):GeneralValue
+getValues():GeneralValue[]
+toString():String
2
1
MultipleItemAbstract
+getSize():int
+getItem(index:int):SingleItemAbstract
+getMaxLeftWrist(dim:int):GeneralValue
+getMinLeftWrist(dim:int):GeneralValue
+getAverageLeftWrist(dim:int):GeneralValue
State
-leftWrist:Point3D
-rightWrist:Point3D
-time:double
+State(strg:String)
+getLeftWrist():Point3D
+getRightWrist():Point3D
+getTime():double
+toString():String
+getMaxLeftWrist(dim:int):GeneralValue
+getMinLeftWrist(dim:int):GeneralValue
+getAverageLeftWrist(dim:int):GeneralValue
*
1
Trial
-stateList:ArrayList<State>
-infantID:String
-week:int
-fileName:String
+Trial(directory:String, infantID:String, week:int) throws IOException
+getItem(index:int):State
+getSize():int
+getInfantID():String
+getWeek():int
+getFileName():String
Infant
-trialList:ArrayList<Trial>
-infantID:String
-MAX_WEEK:int = 16
+Infant(directory:String, infantID:String) throws IOException
+getItem(index:int):Trial
+getSize():int
+getInfantID():String
*
1
Driver
+main(args:String[]) throws IOException

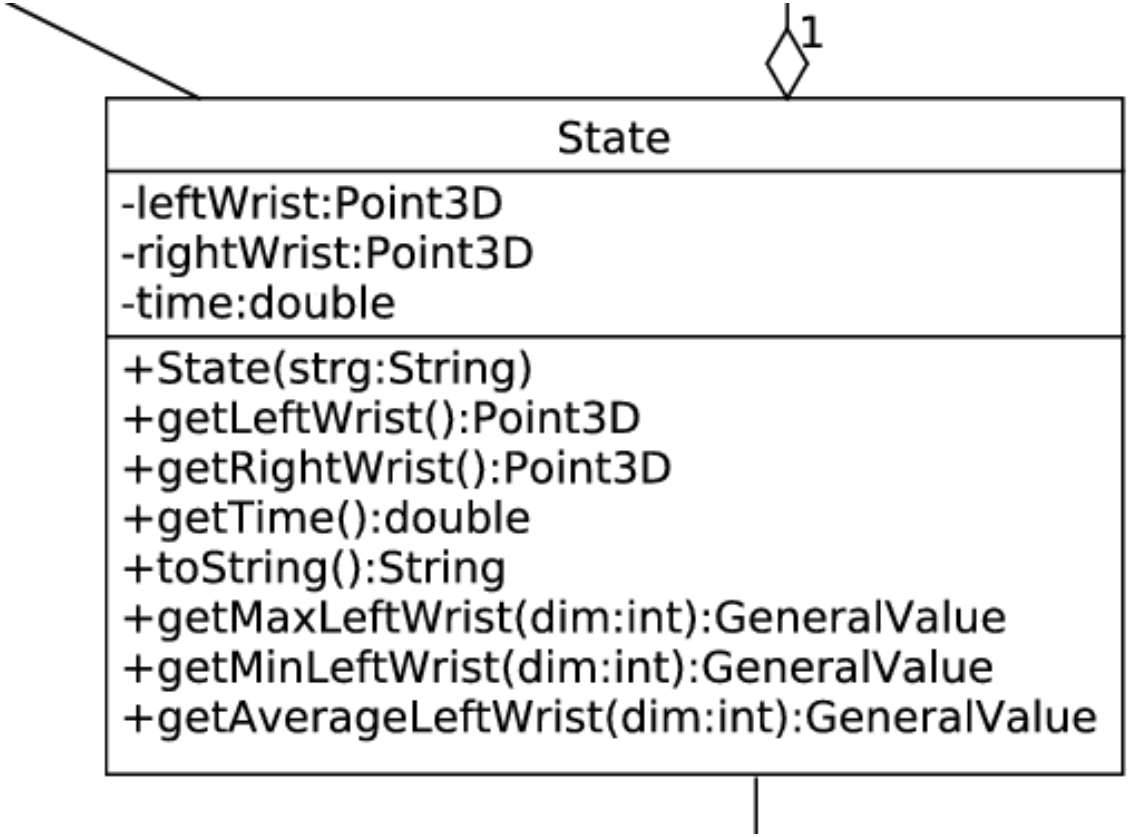
SingleItemAbstract

- Requires implementing classes to provide a way to compute statistics over States
  - For an individual State, this is trivial
  - For a Trial: you have already implemented this code
- Note that a GeneralValue is returned



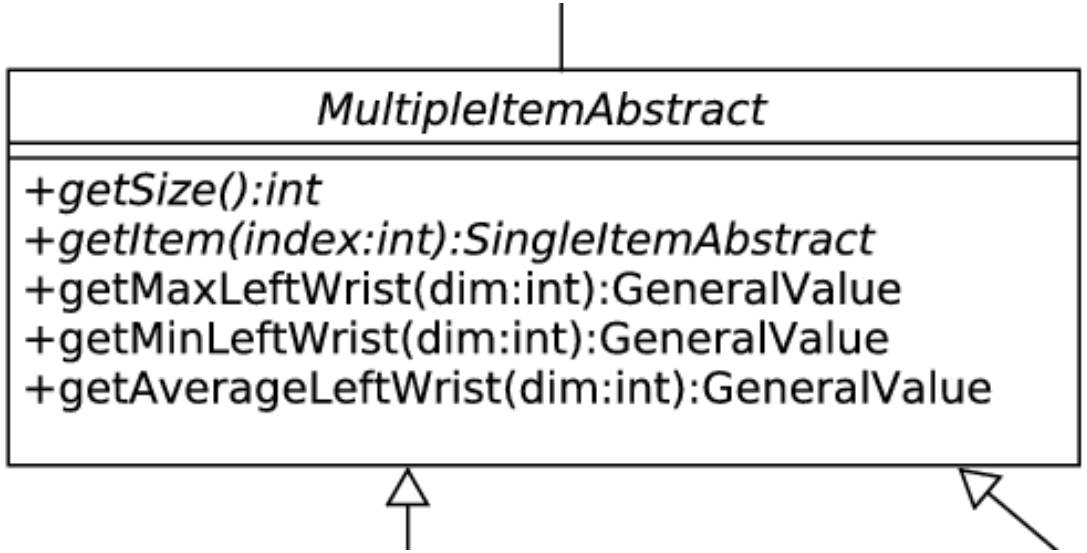
State

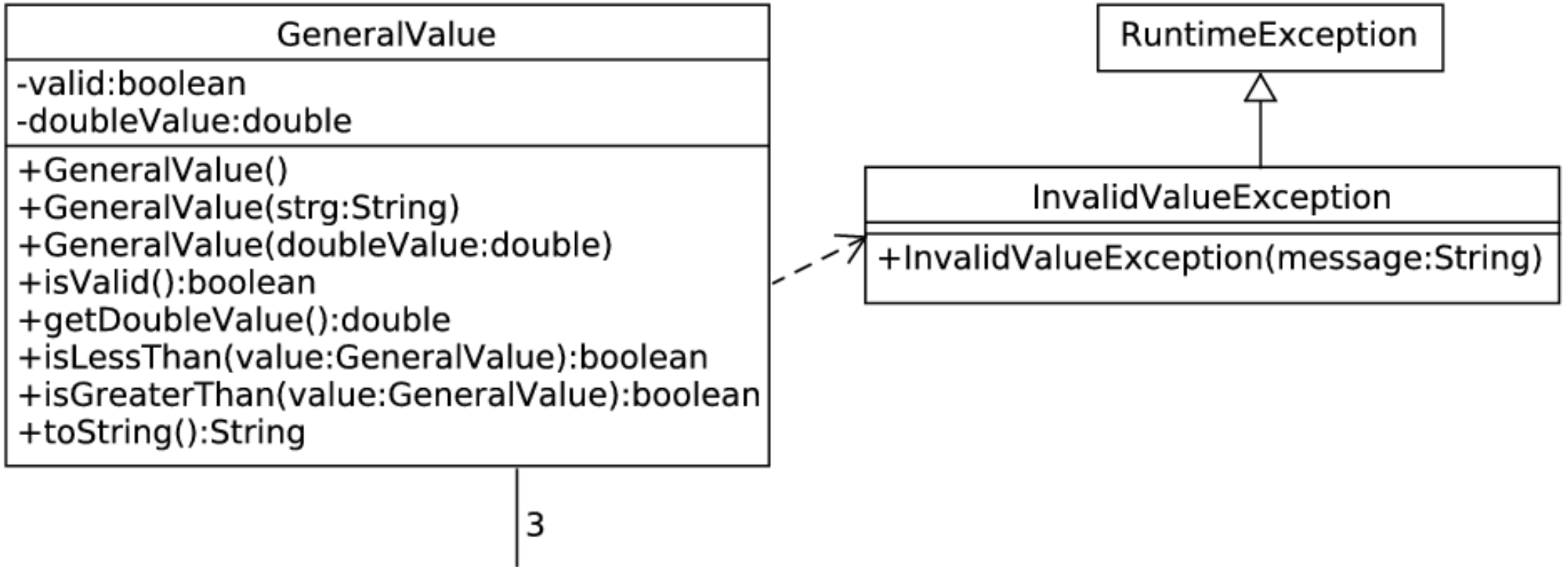
- Similar to project 1
- New: “compute” statistics



MultipleItemAbstract

- Requires from implementing classes:
  - The number of component items
  - Access to the individual items
- Provides:
  - Statistics computation over all containing items




GeneralValue
-valid:boolean
-doubleValue:double
+GeneralValue()
+GeneralValue(strg:String)
+GeneralValue(doubleValue:double)
+isValid():boolean
+getDoubleValue():double
+isLessThan(value:GeneralValue):boolean
+isGreaterThan(value:GeneralValue):boolean
+toString():String
3
RuntimeException
InvalidValueException
+InvalidValueException(message:String)

double getDoubleValue()

- InvalidValueException class: extends RunTimeException
- This method now throws InvalidValueException if this method is called on an invalid GeneralValue
  - Because this is a RunTimeException, we don’t need to explicitly declare this in the method prototype

boolean isLessThan(GeneralValue v)

Should *this* replace v as the smaller value?

this	**v**	**return**
invalid	invalid	false
invalid	5.7	false
2.38	invalid	true
2.38	5.7	true
5.7	2.38	false
5.7	5.7	false

Data Loading

- Trial constructor:
  - Takes as input a directory, an infant ID and a week
  - Loads the corresponding file
  - Throws FileNotFoundException if the file does not exist

- Infant constructor:
  - Takes as input a directory and an infant ID
  - Iterates through the possible weeks: 1 … MAX_WEEKS
  - Attempts to load the Trial
  - When the Trial successfully loads, add it to the list of Trials

Notes

- It is possible that all of the States in a Trial have invalid data
  - Statistics computation must acknowledge this by returning a GeneralValue and not a double
- Likewise for Trials in an Infant

Deadlines

- Project must be submitted by Wednesday, Oct 11th @1:29pm
- Code review must be completed by Wednesday, Oct 18th