CS 2334: Project 3
Java Collections Framework

Project 2 Lessons

Project 2 Lessons

- Class hierarchies
- Code reuse through abstract classes
- Computing statistics in the presence of invalid data

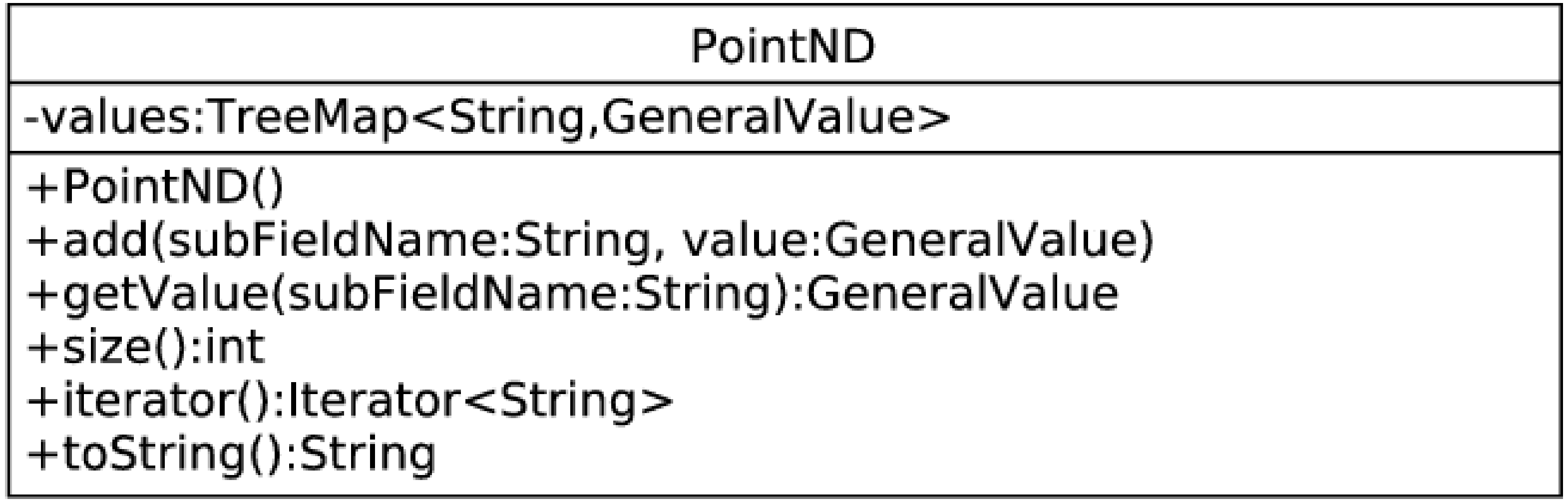
- GeneralValue:
  - Should only ask what the value is if we know that it is valid
  - Should not use exceptions to check validity

New Data Format

Many different fields

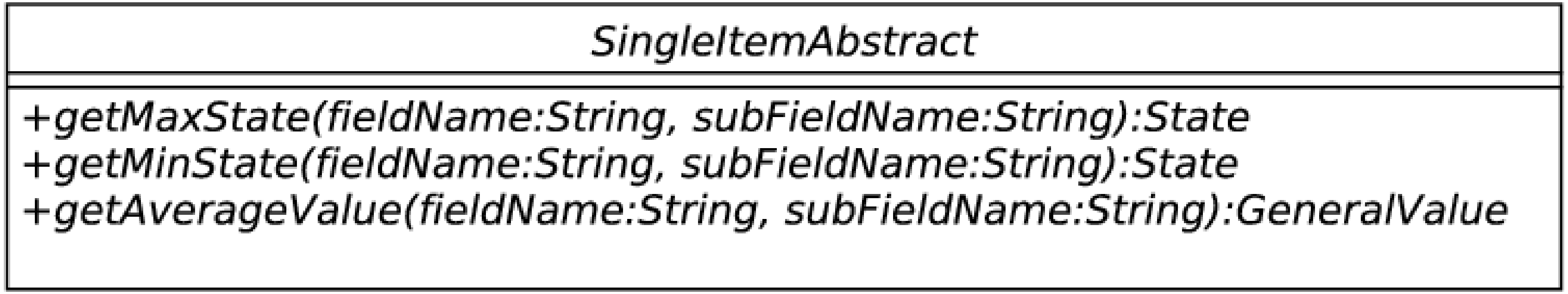
- left_wrist, right_elbow, upper_back, …
- Each field can contain subfields: x, y, z
- GeneralValue associate with each field/subfield pair
- Columns can come in any order
- Different columns for different files

Representing the Values in a Field



```
PointND
-values:TreeMap<String,GeneralValue>
+PointND()
+add(subFieldName:String, value:GeneralValue)
+getValue(subFieldName:String):GeneralValue
+size():int
+iterator():Iterator<String>
+toString():String
```

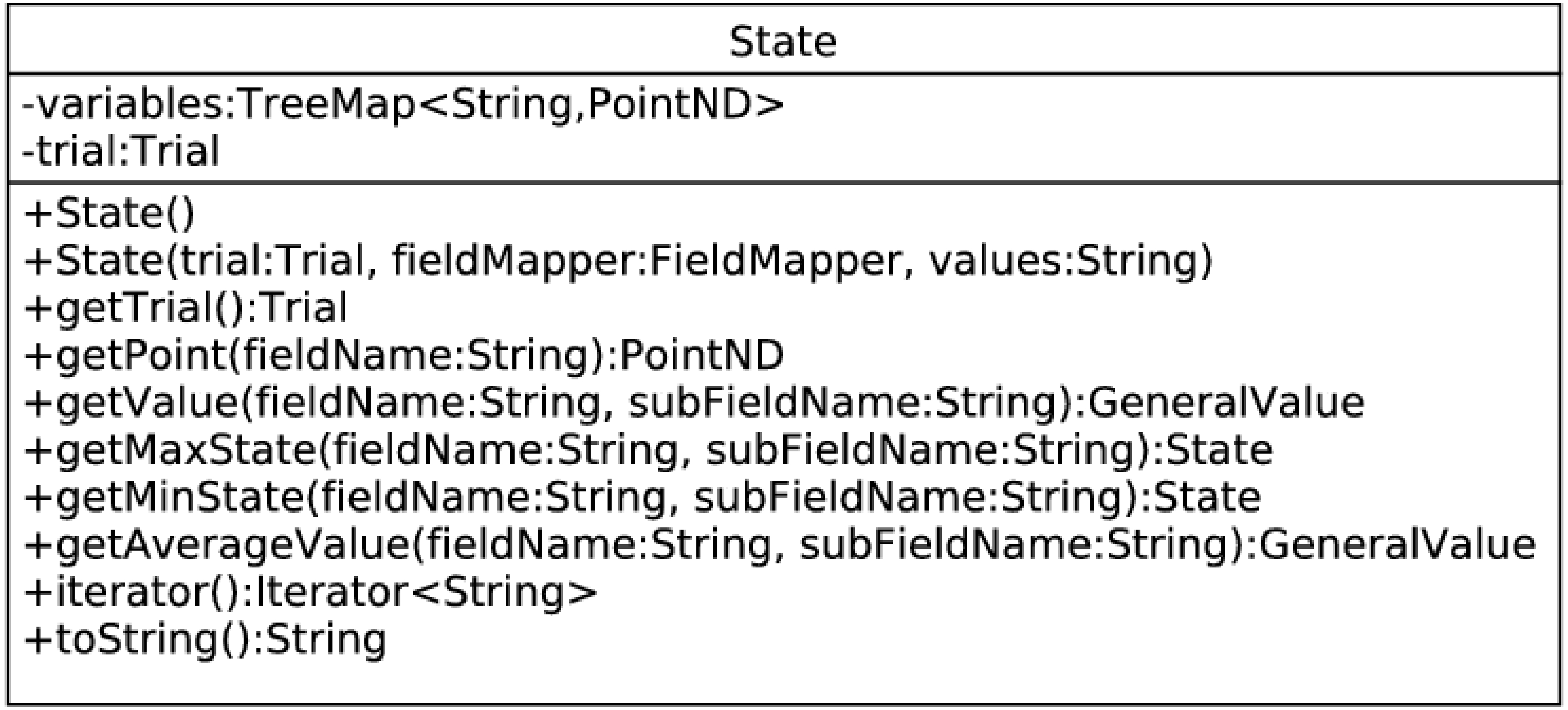
Computing Statistics



SingleItemAbstract

+getMaxState(fieldName:String, subFieldName:String):State
+getMinState(fieldName:String, subFieldName:String):State
+getAverageValue(fieldName:String, subFieldName:String):GeneralValue

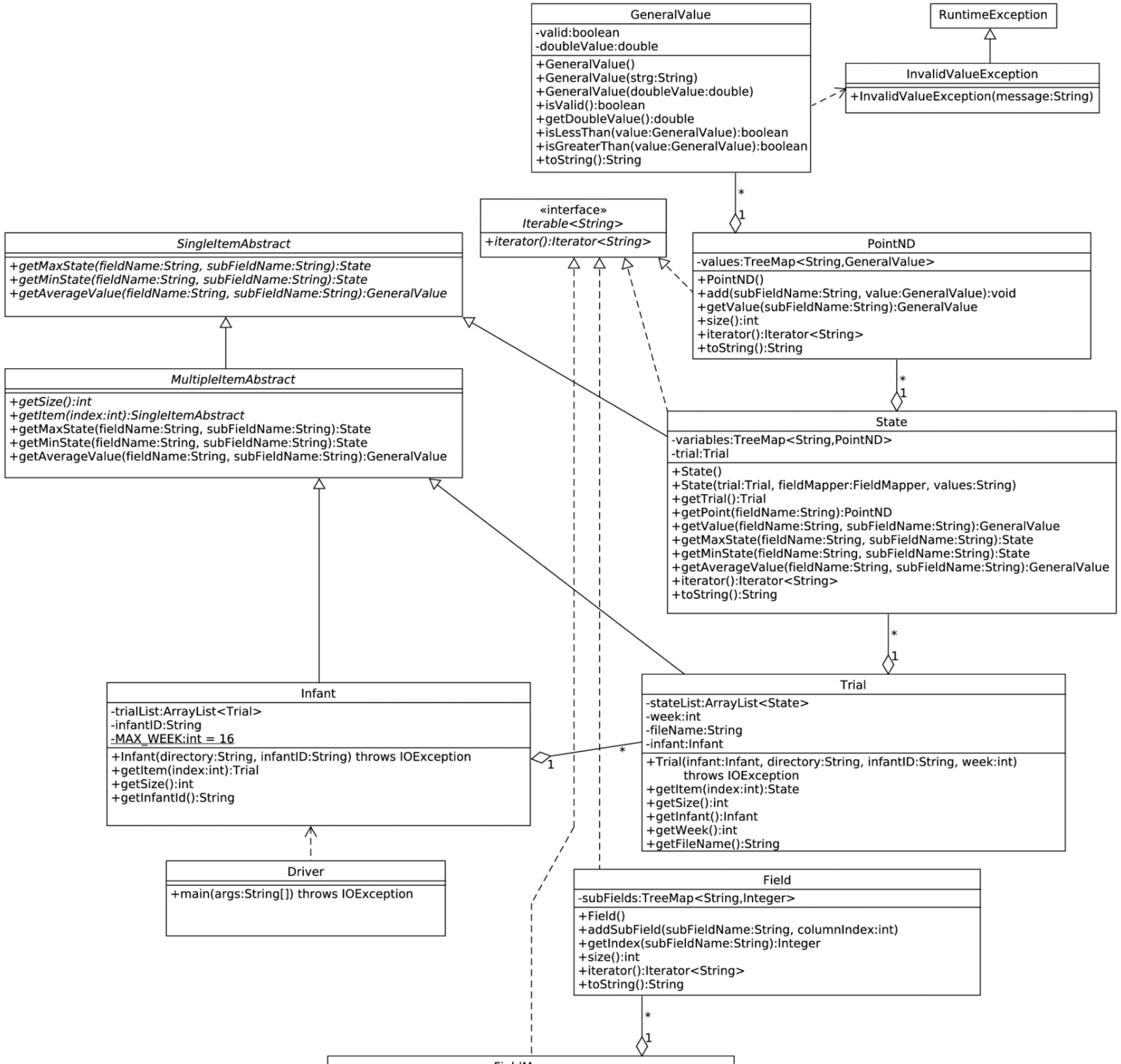
States



State

-variables:TreeMap<String,PointND>
-trial:Trial

+State()
+State(trial:Trial, fieldMapper:FieldMapper, values:String)
+getTrial():Trial
+getPoint(fieldName:String):PointND
+getValue(fieldName:String, subFieldName:String):GeneralValue
+getMaxState(fieldName:String, subFieldName:String):State
+getMinState(fieldName:String, subFieldName:String):State
+getAverageValue(fieldName:String, subFieldName:String):GeneralValue
+iterator():Iterator<String>
+toString():String



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)

«interface»
Iterable<String>

+iterator():Iterator<String>

SingleItemAbstract

+getMaxState(fieldName:String, subFieldName:String):State
+getMinState(fieldName:String, subFieldName:String):State
+getAverageValue(fieldName:String, subFieldName:String):GeneralValue

PointND

-values:TreeMap<String,GeneralValue>

+PointND()
+add(subFieldName:String, value:GeneralValue):void
+getValue(subFieldName:String):GeneralValue
+size():int
+iterator():Iterator<String>
+toString():String

MultipleItemAbstract

+getSize():int
+getItem(index:int):SingleItemAbstract
+getMaxState(fieldName:String, subFieldName:String):State
+getMinState(fieldName:String, subFieldName:String):State
+getAverageValue(fieldName:String, subFieldName:String):GeneralValue

State

-variables:TreeMap<String,PointND>
-trial:Trial

+State()
+State(trial:Trial, fieldMapper:FieldMapper, values:String)
+getTrial():Trial
+getPoint(fieldName:String):PointND
+getValue(fieldName:String, subFieldName:String):GeneralValue
+getMaxState(fieldName:String, subFieldName:String):State
+getMinState(fieldName:String, subFieldName:String):State
+getAverageValue(fieldName:String, subFieldName:String):GeneralValue
+iterator():Iterator<String>
+toString():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

Trial

-stateList:ArrayList<State>
-week:int
-fileName:String
-infant:Infant

+Trial(infant:Infant, directory:String, infantID:String, week:int)
throws IOException
+getItem(index:int):State
+getSize():int
+getInfant():Infant
+getWeek():int
+getFileName():String

Driver

+main(args:String[]) throws IOException

Field

-subFields:TreeMap<String,Integer>

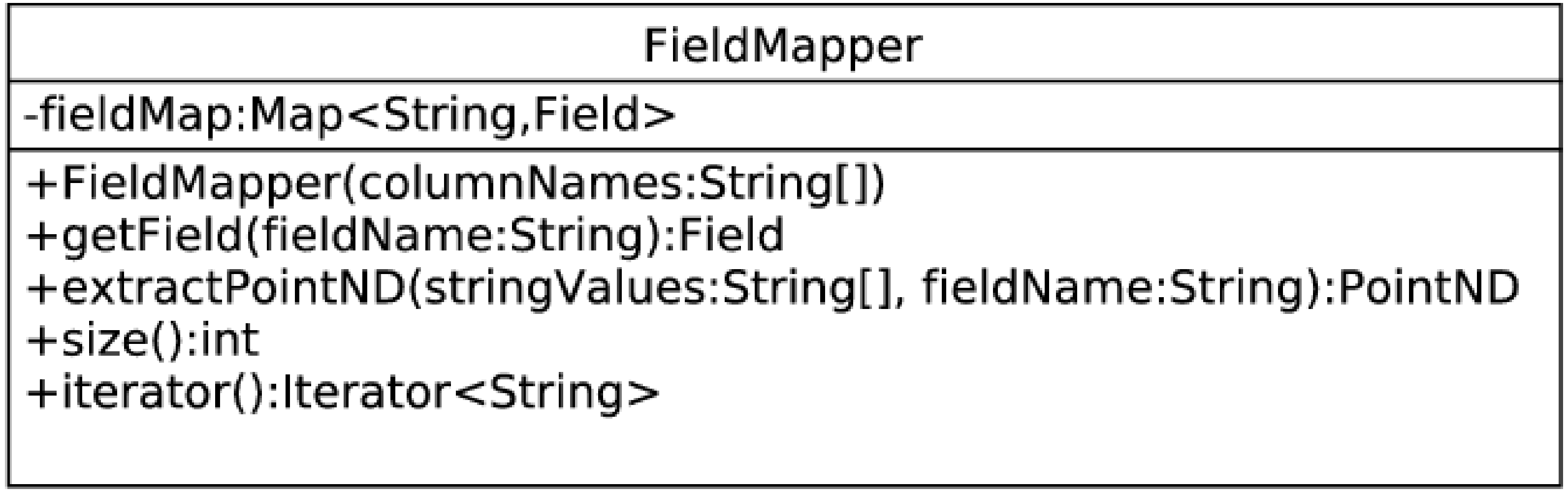
+Field()
+addSubField(subFieldName:String, columnIndex:int)
+getIndex(subFieldName:String):Integer
+size():int
+iterator():Iterator<String>
+toString():String

FieldM

Data Files to Data Representation

- Need a way to translate from columns in a file to Fields/Subfields
- FieldMapper: represent all fields in a file
- Field: represent a single field (with its subfields)
    - Translate subfield into a column index

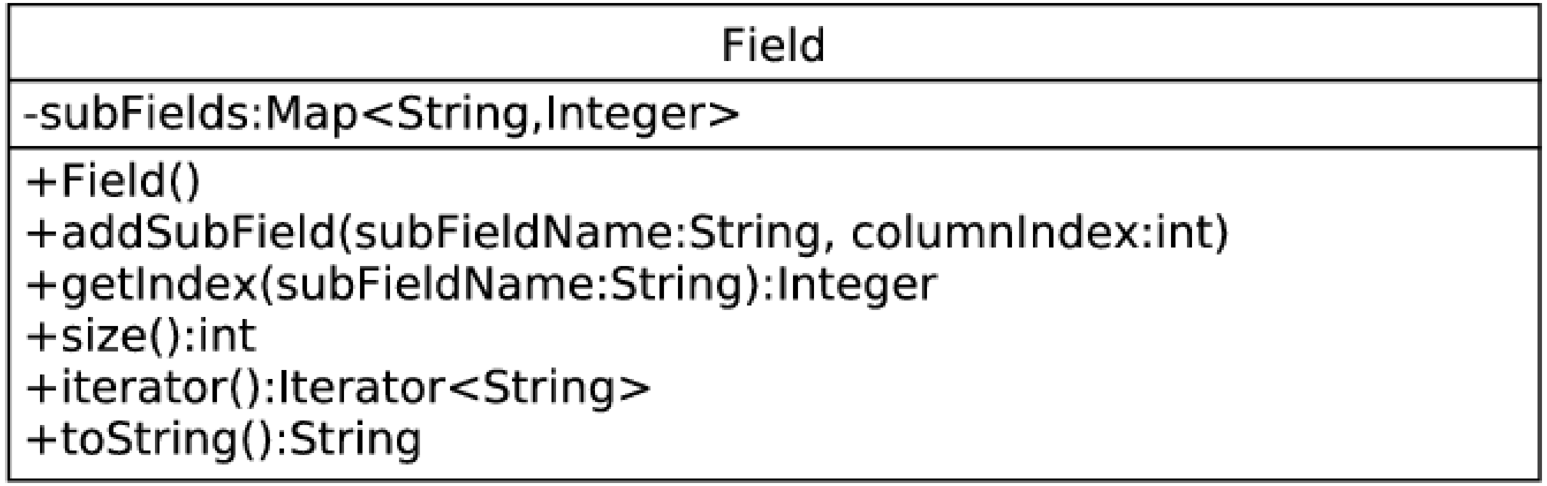
FieldMapper



FieldMapper

-fieldMap:Map<String,Field>

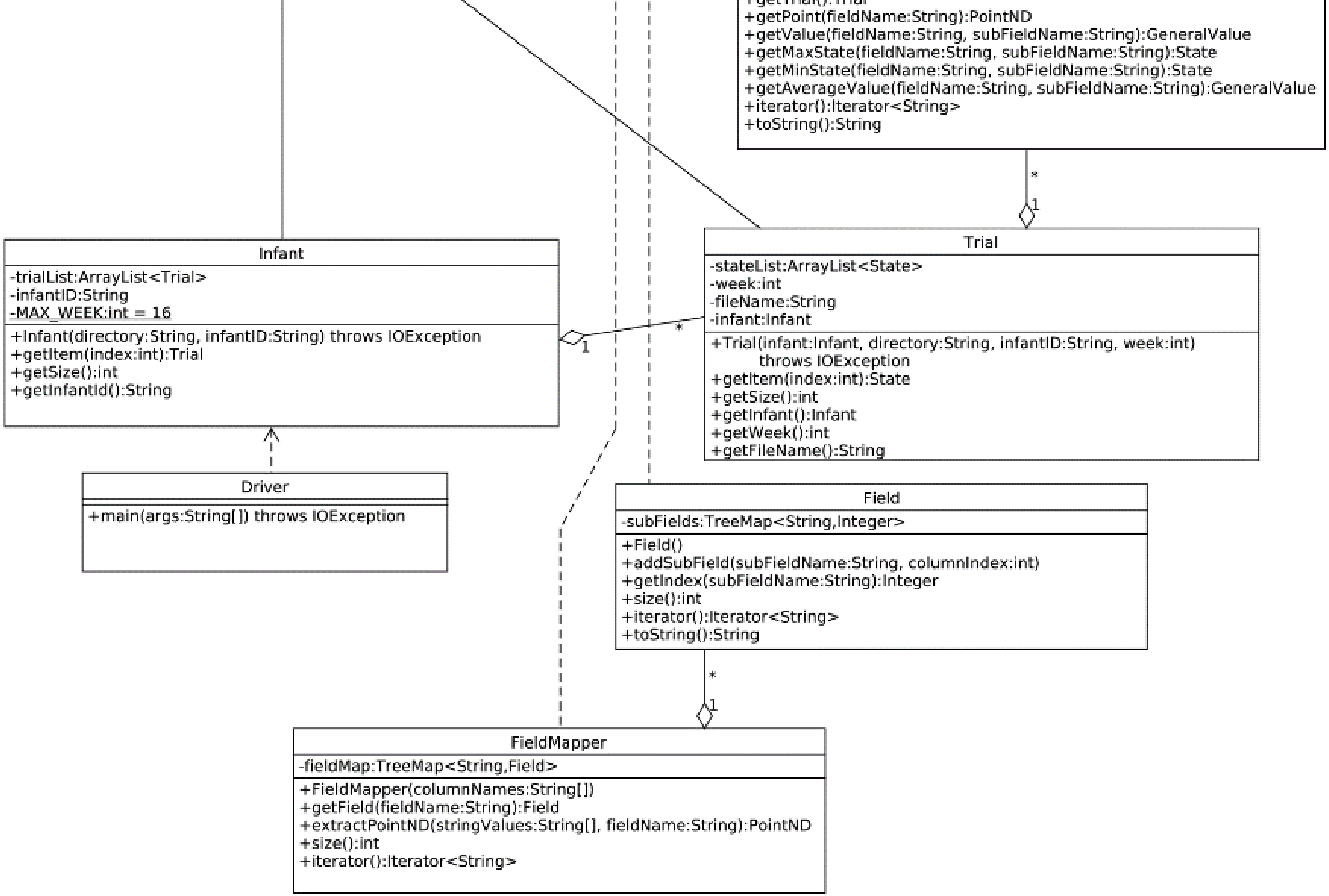
+FieldMapper(columnNames:String[])
+getField(fieldName:String):Field
+extractPointND(stringValues:String[], fieldName:String):PointND
+size():int
+iterator():Iterator<String>



Field

-subFields:Map<String,Integer>

+Field()
+addSubField(subFieldName:String, columnIndex:int)
+getIndex(subFieldName:String):Integer
+size():int
+iterator():Iterator<String>
+toString():String


+getTrial():Trial
+getPoint(fieldName:String):PointND
+getValue(fieldName:String, subFieldName:String):GeneralValue
+getMaxState(fieldName:String, subFieldName:String):State
+getMinState(fieldName:String, subFieldName:String):State
+getAverageValue(fieldName:String, subFieldName:String):GeneralValue
+iterator():Iterator<String>
+toString():String
*
1
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
*
Trial
-stateList:ArrayList<State>
-week:int
-fileName:String
-infant:Infant
+Trial(infant:Infant, directory:String, infantID:String, week:int) throws IOException
+getItem(index:int):State
+getSize():int
+getInfant():Infant
+getWeek():int
+getFileName():String
Driver
+main(args:String[]) throws IOException
Field
-subFields:TreeMap<String,Integer>
+Field()
+addSubField(subFieldName:String, columnIndex:int)
+getIndex(subFieldName:String):Integer
+size():int
+iterator():Iterator<String>
+toString():String
*
1
FieldMapper
-fieldMap:TreeMap<String,Field>
+FieldMapper(columnNames:String[])
+getField(fieldName:String):Field
+extractPointND(stringValues:String[], fieldName:String):PointND
+size():int
+iterator():Iterator<String>

Deadlines

- Project must be submitted by Friday, Oct 27th @1:29pm
- Code review must be completed by Friday, Nov 3rd