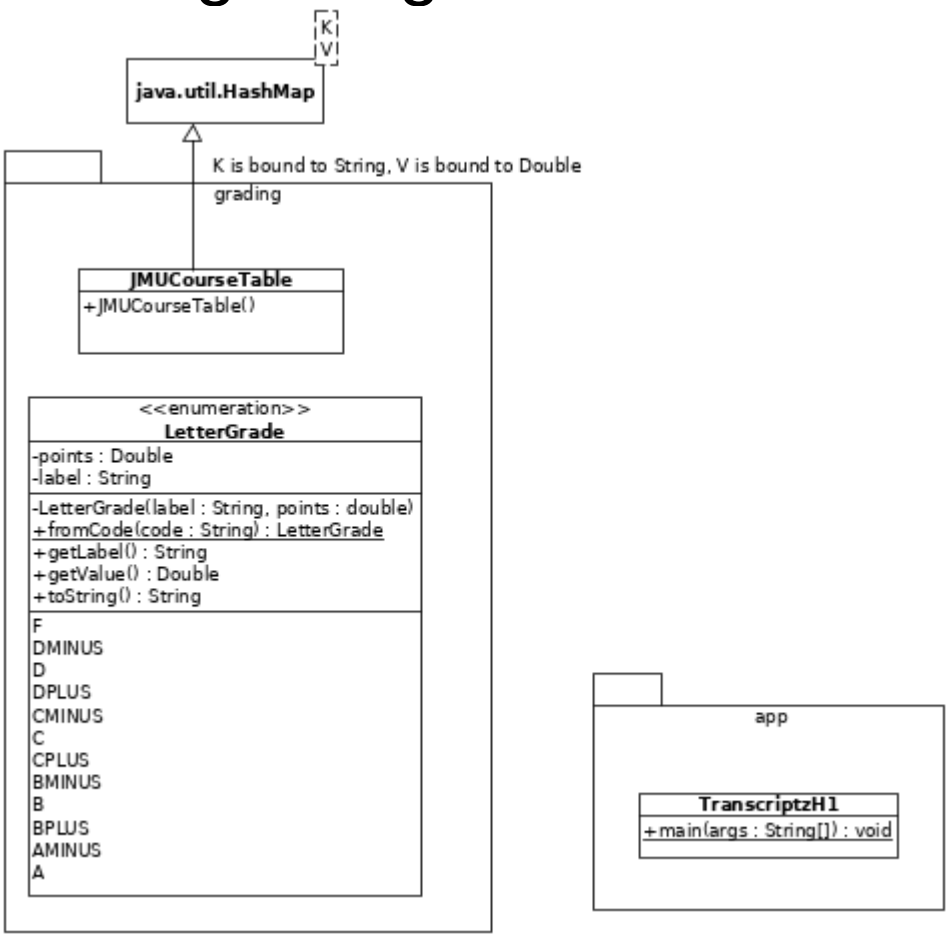
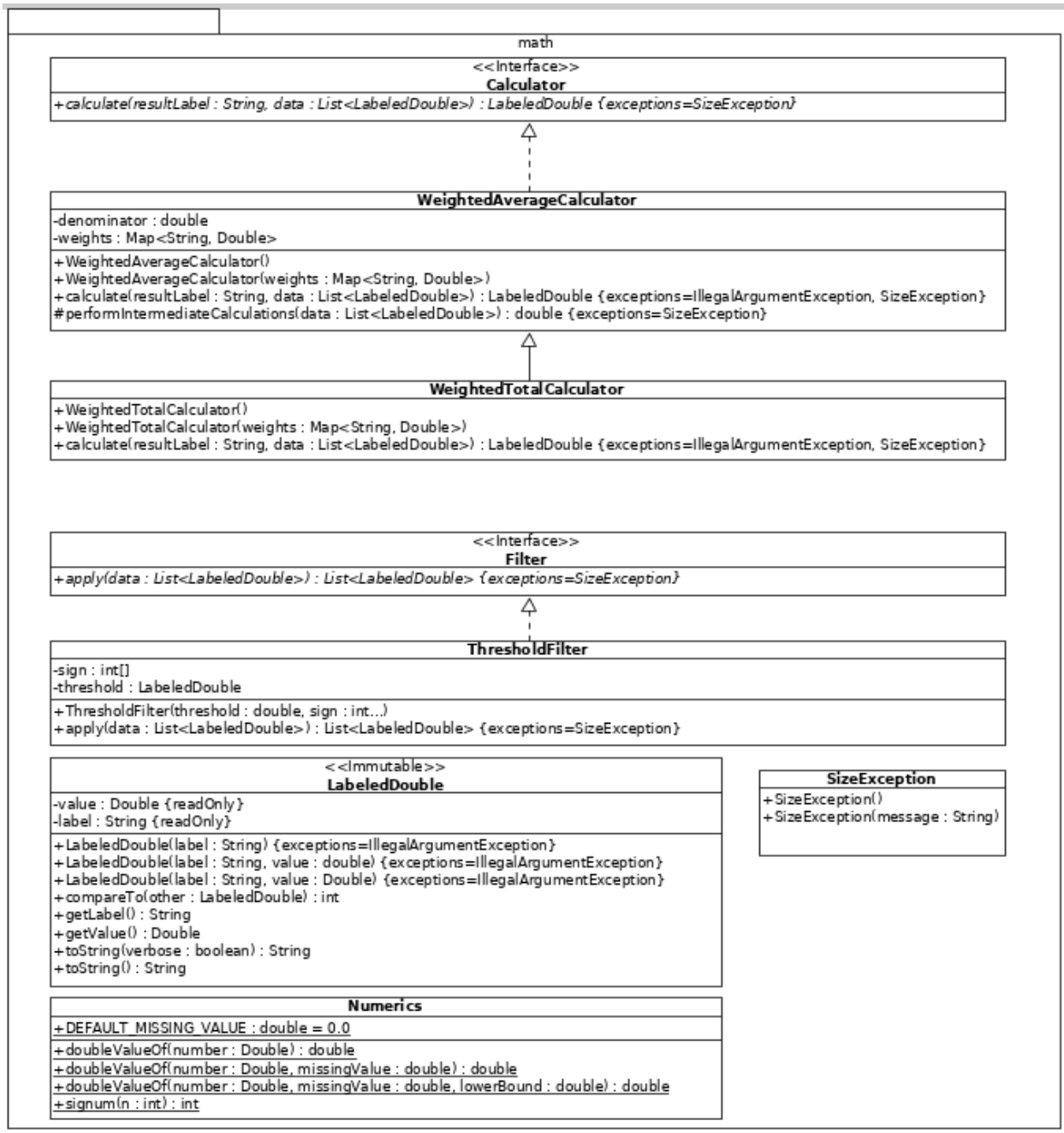
Transcriptz

Engineering Design




math
<<Interface>>
Calculator
+calculate(resultLabel : String, data : List<LabeledDouble>) : LabeledDouble {exceptions=SizeException}
WeightedAverageCalculator
-denominator : double
-weights : Map<String, Double>
+WeightedAverageCalculator()
+WeightedAverageCalculator(weights : Map<String, Double>)
+calculate(resultLabel : String, data : List<LabeledDouble>) : LabeledDouble {exceptions=IllegalArgumentException, SizeException}
#performIntermediateCalculations(data : List<LabeledDouble>) : double {exceptions=SizeException}
WeightedTotalCalculator
+WeightedTotalCalculator()
+WeightedTotalCalculator(weights : Map<String, Double>)
+calculate(resultLabel : String, data : List<LabeledDouble>) : LabeledDouble {exceptions=IllegalArgumentException, SizeException}
<<Interface>>
Filter
+apply(data : List<LabeledDouble>) : List<LabeledDouble> {exceptions=SizeException}
ThresholdFilter
-sign : int[]
-threshold : LabeledDouble
+ThresholdFilter(threshold : double, sign : int...)
+apply(data : List<LabeledDouble>) : List<LabeledDouble> {exceptions=SizeException}
<<Immutable>>
LabeledDouble
-value : Double {readOnly}
-label : String {readOnly}
+LabeledDouble(label : String) {exceptions=IllegalArgumentException}
+LabeledDouble(label : String, value : double) {exceptions=IllegalArgumentException}
+LabeledDouble(label : String, value : Double) {exceptions=IllegalArgumentException}
+compareTo(other : LabeledDouble) : int
+getLabel() : String
+getValue() : Double
+toString(verbose : boolean) : String
+toString() : String
SizeException
+SizeException()
+SizeException(message : String)
Numerics
+DEFAULT_MISSING_VALUE : double = 0.0
+doubleValueOf(number : Double) : double
+doubleValueOf(number : Double, missingValue : double) : double
+doubleValueOf(number : Double, missingValue : double, lowerBound : double) : double
+signum(n : int) : int