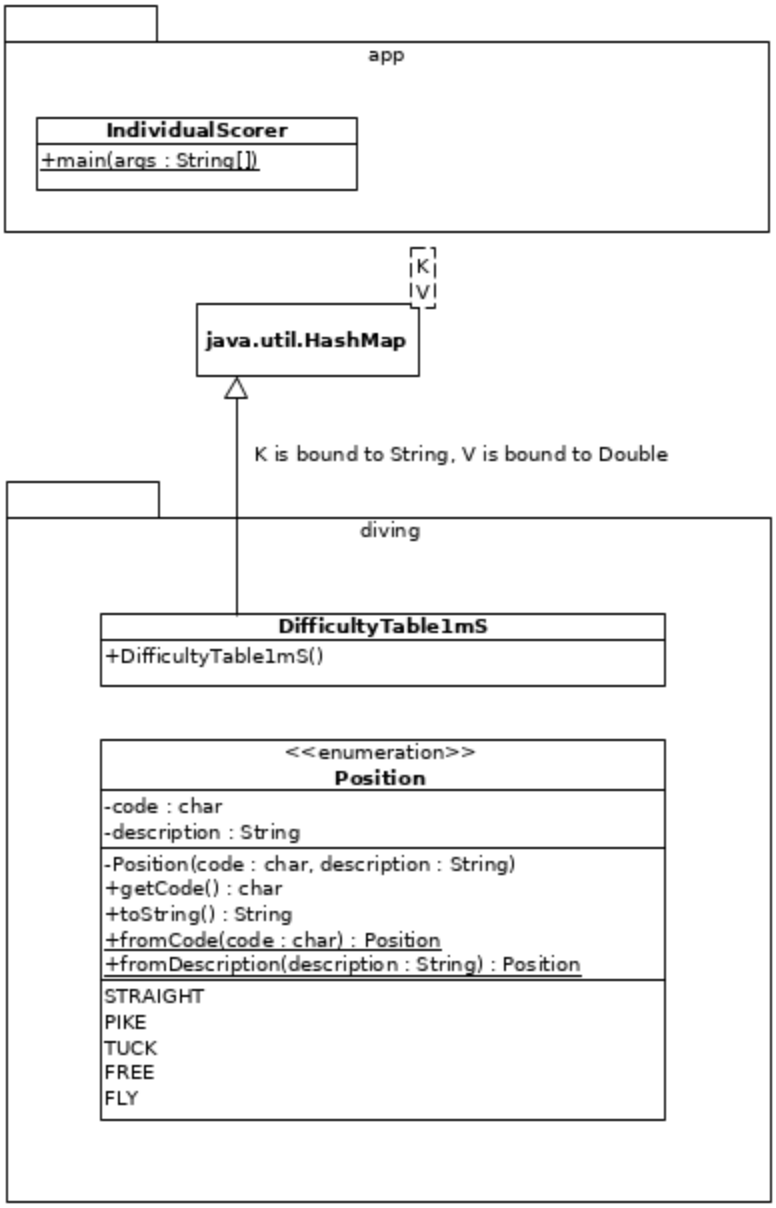
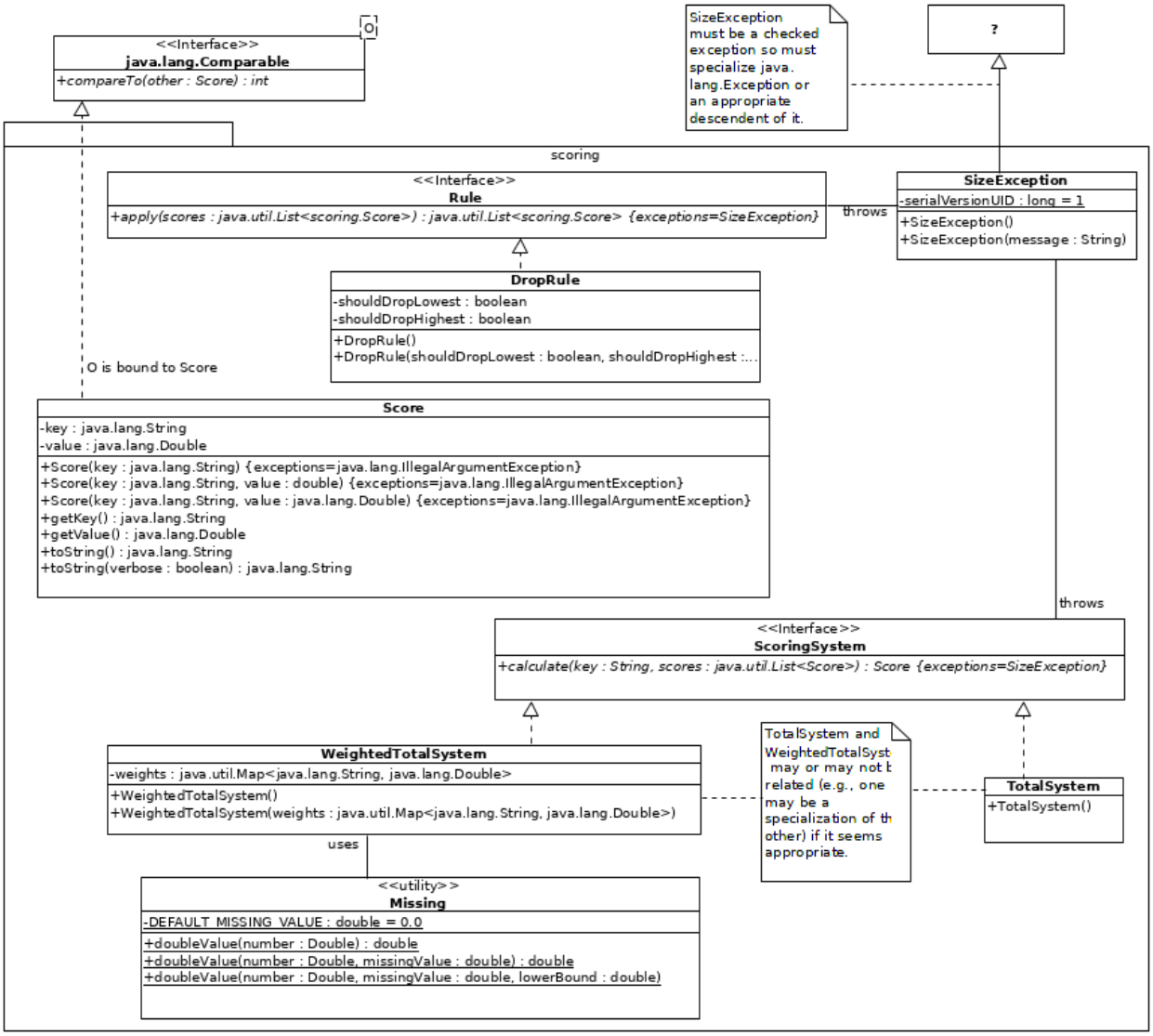
Engineering Design




O
<<Interface>>
java.lang.Comparable
+compareTo(other : Score) : int
SizeException must be a checked exception so must specialize java.lang.Exception or an appropriate descendent of it.
?
scoring
<<Interface>>
Rule
+apply(scores : java.util.List<scoring.Score>) : java.util.List<scoring.Score> {exceptions=SizeException}
throws
SizeException
-serialVersionUID : long = 1
+SizeException()
+SizeException(message : String)
DropRule
-shouldDropLowest : boolean
-shouldDropHighest : boolean
+DropRule()
+DropRule(shouldDropLowest : boolean, shouldDropHighest :...
O is bound to Score
Score
-key : java.lang.String
-value : java.lang.Double
+Score(key : java.lang.String) {exceptions=java.lang.IllegalArgumentException}
+Score(key : java.lang.String, value : double) {exceptions=java.lang.IllegalArgumentException}
+Score(key : java.lang.String, value : java.lang.Double) {exceptions=java.lang.IllegalArgumentException}
+getKey() : java.lang.String
+getValue() : java.lang.Double
+toString() : java.lang.String
+toString(verbose : boolean) : java.lang.String
throws
<<Interface>>
ScoringSystem
+calculate(key : String, scores : java.util.List<Score>) : Score {exceptions=SizeException}
WeightedTotalSystem
-weights : java.util.Map<java.lang.String, java.lang.Double>
+WeightedTotalSystem()
+WeightedTotalSystem(weights : java.util.Map<java.lang.String, java.lang.Double>)
TotalSystem and WeightedTotalSyst may or may not b related (e.g., one may be a specialization of th other) if it seems appropriate.
TotalSystem
+TotalSystem()
uses
<<utility>>
Missing
-DEFAULT_MISSING_VALUE : double = 0.0
+doubleValue(number : Double) : double
+doubleValue(number : Double, missingValue : double) : double
+doubleValue(number : Double, missingValue : double, lowerBound : double)