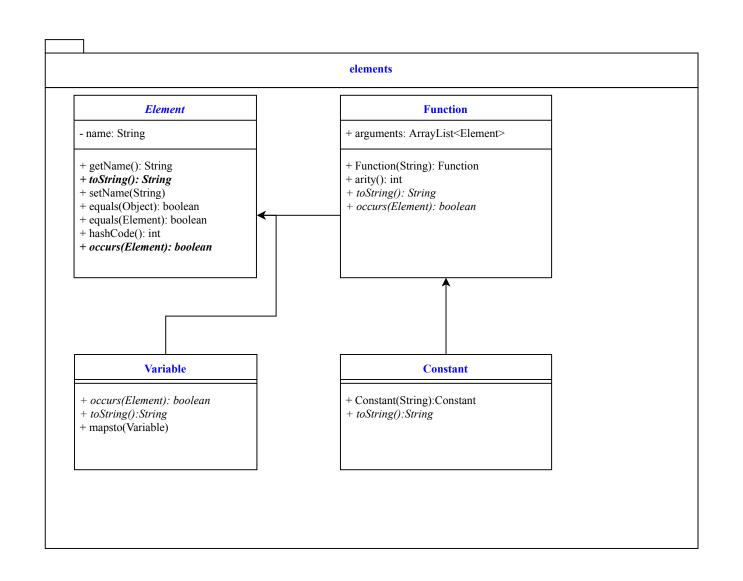## tool

Tuple<E>

- f: E
- s: E

+ Tuple(E,E): Tuple
+ getFirst(): E
+ setFirst(E)
+ getSecond(): E
+ setSecond(E)
+ *toString(): String*

Matrix

- content: float[]
- size: int

+ Matrix(int): Matrix
+ putAt(Tuple<Integer>,float)
+ getAt(Tuple<Integer>): float
+ isPM(Matrix): boolean
+ getSize():int

elements

Element

- name: String

+ getName(): String
+ ***toString(): String***
+ setName(String)
+ equals(Object): boolean
+ equals(Element): boolean
+ hashCode(): int
+ ***occurs(Element): boolean***

Function

+ arguments: ArrayList<Element>

+ Function(String): Function
+ arity(): int
+ *toString(): String*
+ *occurs(Element): boolean*

Variable

+ *occurs(Element): boolean*
+ *toString():String*
+ mapsto(Variable)

Constant

+ Constant(String):Constant
+ *toString():String*

unificationProblem

Unifier

- right: Element
- left: Element
- numberOfFunctions: int
- sortedListOfFunctions: ArrayList<Function>
- proximityRelations: Matrix
- openCases: ArrayList<Tuple<Functions>>

+ addOpenCase(Tuple<Function>): boolean
+ getNextOpenCase(): Tuple<Function>
+ closeCase(Tuple<Function>,float): boolean
+ getRight(): Element
+ setRight(Element)
+ getLeft(): Element
+ setLeft(Element)
+ getNumberOfFunctions(): int
+ setNumberOfFunctions(int)
+ getSortedListOfFunctions(): ArrayList<Function>
+ setSortedListOfFunctions(ArrayList<Function>)
+ getProximityRelations(): Matrix
+ setProximityRelations(Matrix)

inputParser

- FIRST_VARIABLE: char = 'u'
- listOfFunctions: ArrayList<Function>

+ parse(String): ArrayList<Unifier>
- parseSub(String): Element

Problem

+ p: ArrayList<Tuple<Element>>
+ c: ArrayList<Tuple<Element>>
+ sigma: ArrayList<Tuple<Element>>
+ psi: ArrayList<Tuple<Element>>
- lambda: float

---

methods

ALGORITHMS

+ preUnification(Problem): boolean
- preUnification Rules ...