

Association
multiplicity : GZ
visibility : init
+Association (head: Class*, tail: Class*, multiplicity: GZ, roleName: Text) <constructor>
+getMultiplicity : GZ
+getRoleName : Text

Inheritance
+Inheritance (head: Class*, tail: Class*) <constructor>

testChart
classes : std::vector<std::unique_ptr<Class>>
edges : std::vector<std::unique_ptr<Edge>>
+addClass (class_: Class)
+addEdge (edge: Edge*)
+getClasses () const: std::vector<std::unique_ptr<Class>>
+getEdges () const: std::vector<std::unique_ptr<Edge>>

Edge
+Edge (head: Class*, tail: Class*) <constructor>
+Edge () <destructor>
+getHead : Class*
+getTail : Class*

Argument
name : Text
type : std::unique_ptr<Type>
+Argument (name: Text, type: Type*) <constructor>
+getName : Text
+getType : const std::unique_ptr<Type>&

Class
name : Text
operations : std::vector<Operation>
attributes : std::vector<Attribute>
edges : std::vector<Edge*>
+Class (name: Text) <constructor>
+getName : Text
+getOperations () const: std::vector<Operation>&
+getAttributes () const: std::vector<Attribute>&
+getEdges () const: std::vector<Edge*>&
+getParentClasses () const: std::vector<Class*>&
+getParameters (operation: Operation)
+addOperation (operation: Operation)
+setAttribute (attribute: Attribute)
+addEdge (edge: Edge*)
+isAbstract : Boolean

Attribute
name : Text
type : std::unique_ptr<Type>
visibility : Visibility
+Attribute (name: Text, type: Type*, visibility: Visibility) <constructor>
+getName : Text
+getType : const std::unique_ptr<Type>&
+getVisibility : Visibility

Operation
name : Text
returnType : std::unique_ptr<Type>
arguments : std::vector<Argument>
visibility : Visibility
abstract : Boolean
stereotype : Stereotype
+Operation (name: Text, returnType: Type*, arguments: std::vector<Argument>&, visibility: Visibility, abstract: Boolean, stereotype: Stereotype) <constructor>
+getName : Text
+getReturnType : const std::unique_ptr<Type>&
+getArguments () const: std::vector<Argument>&
+getVisibility : Visibility
+isAbstract : Boolean
+getStereotype : Stereotype

Argument
name
getName
return name
getType
return type
Operation
name
return name
getReturnType
return returnType
getArguments
return arguments
getVisibility
return visibility
isAbstract
return abstract
getStereotype
return stereotype
Attribute
name
getName
return name
getType
return type
getVisibility
return visibility
Edge
name
getHead
return head
getTail
return tail
Class
name
getName
return name
getOperations
return operations
getAttributes
return attributes
getEdges
return edges
getParentClasses
vector < Class > parentClasses
each edge: Class in edges
edge: get(Tail) == this
visibility
parentClass.push_back(edge.getHead())
return parentClasses
getAncestorClasses
each parent: Class in getParentClasses()
visibility
classes.insert(parent)
parent: getAncestorClasses().classes
addOperation
operations.push_back(move(operation))
addAttribute
attributes.push_back(move(attribute))
addEdge
edges.push_back(edge)
isAbstract
self < Class > classes
getAncestorClasses().classes
classes.insert(this)
map < string, bool > operationImplemented
each parent: Class in classes
operationImplemented(operation.getName()) == not operation.isAbstract()
each implemented: Boolean in operationImplemented
visibility
not implemented
return true
return false
Inheritance
name
Association
name
getMultiplicity
return multiplicity
getRoleName
return roleName
addClass
classes.emplace_back(class_)
addEdge
edges.emplace_back(edge)
edge: getHead().addEdge(edge)
edge: getTail().addEdge(edge)
getClasses
return classes
getEdges
return edges