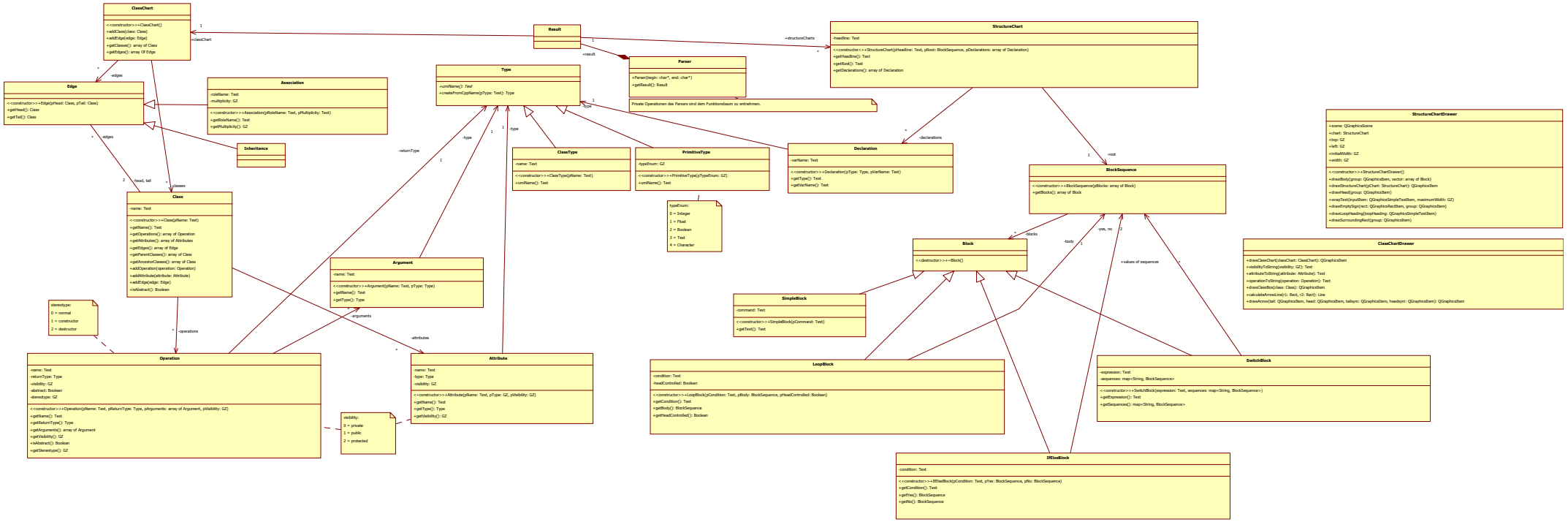
ClassChart
<<constructor>>+ClassChart()
+addClass(class: Class)
+addEdge(edge: Edge)
+getClasses(): array of Class
+getEdges(): array Of Edge
1
+classChart
Result
1
+result
+structureCharts
*
StructureChart
-headline: Text
<<constructor>>+StructureChart(pHeadline: Text, pRoot: BlockSequence, pDeclarations: array of Declaration)
+getHeadline(): Text
+getRoot(): Text
-getDeclarations(): array of Declaration
Parser
+Parser(begin: char*, end: char*)
+getResult(): Result
Private Operationen des Parsers sind dem Funktionsbaum zu entnehmen.
Edge
<<constructor>>+Edge(pHead: Class, pTail: Class)
+getHead(): Class
+getTail(): Class
*
-edges
Association
-roleName: Text
-multiplicity: GZ
<<constructor>>+Association(pRoleName: Text, pMultiplicity: Text)
+getRoleName(): Text
+getMultiplicity(): GZ
Inheritance
Type
+umlName(): Text
+createFromCppName(pType: Text): Type
1
-type
-type
1
1
-type
-returnType
1
ClassType
-name: Text
<<constructor>>+ClassType(pName: Text)
+umlName(): Text
PrimitiveType
-typeEnum: GZ
<<constructor>>+PrimitiveType(pTypeEnum: GZ)
+umlName(): Text
typeEnum:
0 = Integer
1 = Float
2 = Boolean
3 = Text
4 = Character
Declaration
-varName: Text
<<constructor>>+Declaration(pType: Type, pVarName: Text)
+getType(): Text
+getVarName(): Text
*
-declarations
1
-root
BlockSequence
<<constructor>>+BlockSequence(pBlocks: array of Block)
+getBlock(): array of Block
StructureChartDrawer
+scene: QGraphicsScene
+chart: StructureChart
+top: GZ
+left: GZ
+initialWidth: GZ
+width: GZ
<<constructor>>+StructureChartDrawer()
+drawBody(group: QGraphicsItem, vector: array of Block)
+drawStructureChart(pChart: StructureChart): QGraphicsItem
+drawHead(group: QGraphicsItem)
+wrapText(inputItem : QGraphicsSimpleTextItem, maximumWidth: GZ)
+drawEmptySign(rect: QGraphicsRectItem, group: QGraphicsItem)
+drawLoopHeading(loopHeading: QGraphicsSimpleTextItem)
+drawSurroundingRect(group: QGraphicsItem)
Class
-name: Text
<<constructor>>+Class(pName: Text)
+getName(): Text
+getOperations(): array of Operation
+getAttributes(): array of Attributes
+getEdges(): array of Edge
+getParentClasses(): array of Class
+getAncestorClasses(): array of Class
+addOperation(operation: Operation)
+addAttribute(attribute: Attribute)
+addEdge(edge: Edge)
+isAbstract(): Boolean
2
-head, tail
*
-classes
*
-operations
Argument
-name: Text
<<constructor>>+Argument(pName: Text, pType: Type)
+getName(): Text
+getType(): Type
-arguments
-attributes
*
Block
<<destructor>>+~Block()
*
-blocks
-body
1
-yes, no
2
+values of sequences
*
ClassChartDrawer
+drawClassChart(classChart: ClassChart): QGraphicsItem
+visibilityToString(visibility: GZ): Text
+attributeToString(attribute: Attribute): Text
+operationToString(operation: Operation): Text
+drawClassBox(class: Class): QGraphicsItem
+calculateArrowLine(r1: Rect, r2: Rect): Line
+drawArrow(tail: QGraphicsItem, head: QGraphicsItem, tailsym: QGraphicsItem, headsym: QGraphicsItem): QGraphicsItem
SimpleBlock
-command: Text
<<constructor>>+SimpleBlock(pCommand: Text)
+getText(): Text
stereotype:
0 = normal
1 = constructor
2 = destructor
Operation
-name: Text
-returnType: Type
-visibility: GZ
-abstract: Boolean
-stereotype: GZ
<<constructor>>+Operation(pName: Text, pReturnType: Type, pArguments: array of Argument, pVisibility: GZ)
+getName(): Text
+getReturnType(): Type
+getArguments(): array of Argument
+getVisibility(): GZ
+isAbstract(): Boolean
+getStereotype(): GZ
visibility:
0 = private
1 = public
2 = protected
Attribute
-name: Text
-type: Type
-visibility: GZ
<<constructor>>+Attribute(pName: Text, pType: GZ, pVisibility: GZ)
+getName(): Text
+getType(): Type
+getVisibility(): GZ
LoopBlock
-condition: Text
-headControlled: Boolean
<<constructor>>+LoopBlock(pCondition: Text, pBody: BlockSequence, pHeadControlled: Boolean)
+getCondition(): Text
+getBody(): BlockSequence
+getHeadControlled(): Boolean
SwitchBlock
-expression: Text
-sequences: map<String, BlockSequence>
<<constructor>>+SwitchBlock(expression: Text, sequences: map<String, BlockSequence>)
+getExpression(): Text
+getSequences(): map<String, BlockSequence>
IfElseBlock
-condition: Text
<<constructor>>+IfElseBlock(pCondition: Text, pThen: BlockSequence, pElse: BlockSequence)
+getCondition(): Text
+getThen(): BlockSequence
+getElse(): BlockSequence