Delimiters

⟨*singleQuote*⟩ ::= '

⟨*doubleQuote*⟩ ::= "

⟨*terminator*⟩ ::= ';'

Literals

⟨***int***⟩ ::= [ integer ]

⟨***bool***⟩ ::= 'true' | 'false'

⟨***char***⟩ ::= ⟨*singleQuote*⟩[ character ]⟨*singleQuote*⟩

⟨***string***⟩ ::= ⟨*doubleQuote*⟩[ character* ]⟨*doubleQuote*⟩

⟨***null***⟩ ::= 'null'

Type Definitions

⟨***type***⟩ ::= 'int' | 'bool' | 'char' | 'string' | 'null'
| ⟨*type*⟩ '->' ⟨*type*⟩
| '('[⟨*type*⟩[',' ⟨*type*⟩]*]')' '->' '('[⟨*type*⟩[',' ⟨*type*⟩]*]')'
| 'List' | 'Array' | 'Set' '['⟨*type*⟩']'
| 'Tuple' '['⟨*type*⟩[',' ⟨*type*⟩]*']'
| 'Dict' '['⟨*type*⟩',' ⟨*type*⟩']'
| ⟨*ident*⟩['['⟨*type*⟩']']

Arithmetic and Boolean Operators

⟨***arithOp***⟩ ::= '+' | '-' | '*' | '/' | '%'

⟨***boolOp***⟩ ::= '<' | '>' | '<=' | '>=' | '!' | '!=' | '==' | '&&' | '||'

⟨***op***⟩ ::= ⟨*arithOp*⟩ | ⟨*boolOp*⟩

Pattern Matching

⟨***value***⟩ ::= ⟨*ident*⟩'('[⟨*value*⟩[','⟨*value*⟩]]')'
| ⟨*prim*⟩
| '_'

⟨***caseVal***⟩ ::= ⟨*ident*⟩':'⟨*type*⟩
| ⟨*value*⟩

⟨***match***⟩ ::= 'match''('⟨*smp*⟩')' '{''case'⟨*caseVal*⟩'=>'⟨*smp*⟩['case'⟨*caseVal*⟩'=>'⟨*smp*⟩]*'}'

Types and Typeclasses

⟨***alias***⟩ ::= ‘`alias`’ ⟨*ident*⟩ ‘`=`’ ⟨*type*⟩

⟨*lowerBoundOp*⟩ ::= ‘`:>`’

⟨*upperBoundOp*⟩ ::= ‘`<:`’

⟨*bounding*⟩ ::= [⟨*lowerBoundOp*⟩⟨*ident*⟩][⟨*upperBoundOp*⟩⟨*ident*⟩]

⟨*generics*⟩ ::= ‘`[`’⟨*ident*⟩⟨*bounding*⟩[‘`,`’ ⟨*ident*⟩⟨*bounding*⟩]*‘`]`’

⟨*genericIdent*⟩ ::= ⟨*ident*⟩[⟨*generics*⟩]

⟨*stmntEnd*⟩ ::= ⟨*terminator*⟩⟨*exp*⟩

⟨*derive*⟩ ::= ‘`derives`’ ⟨*ident*⟩

⟨*constructor*⟩ ::= ⟨*ident*⟩[‘`,`’⟨*ident*⟩]*

⟨*cosntructorList*⟩ ::= ⟨*constructor*⟩[‘`|`’⟨*constructor*⟩]

⟨***adt***⟩ ::= ‘`type`’ ⟨*genericIdent*⟩[⟨*derive*⟩]‘{’[⟨*cosntructorLists*⟩]‘}’⟨*stmntEnd*⟩

⟨*members*⟩ ::= ⟨*ident*⟩‘`:`’⟨*type*⟩[‘`,`’⟨*ident*⟩‘`:`’⟨*type*⟩]*

⟨*extend*⟩ ::= ‘`extends`’⟨*genericIdent*⟩

⟨***record***⟩ ::= [‘`sealed`’]‘`record`’⟨*genericIdent*⟩[⟨*extend*⟩][⟨*derive*⟩]‘{’[⟨*members*⟩]‘}’⟨*stmntEnd*⟩

⟨*signatures*⟩ ::= ⟨*ident*⟩‘`=`’⟨*type*⟩[‘`,`’⟨*ident*⟩‘`=`’⟨*type*⟩]*

⟨***typeclass***⟩ ::= [‘`sealed`’]‘`typeclass`’⟨*genericIdent*⟩[‘`extends`’⟨*ident*⟩]‘{’[⟨*signatures*⟩]‘}’⟨*stmntEnd*⟩

⟨***instance***⟩ ::= ‘`instance`’ ⟨*ident*⟩‘`of`’⟨*ident*⟩‘{’[⟨*funDef*⟩]*‘}’⟨*stmntEnd*⟩

Functions

⟨*param*⟩ ::= ⟨*ident*⟩‘`:`’⟨*type*⟩[‘`=`’ ⟨*atom*⟩ | ⟨*collection*⟩]

⟨***funDef***⟩ ::= ‘`fn`’ ⟨*genericIdent*⟩‘`(`’[⟨*param*⟩ [‘`,`’ ⟨*param*⟩]* ]‘`)`’[‘`->`’⟨*type*⟩]‘`=`’⟨*smp*⟩⟨*terminator*⟩

⟨*prog*⟩ ::= [⟨*funDef*⟩]*⟨*exp*⟩

⟨***app***⟩ ::= ⟨*atom*⟩[ [‘`[`’⟨*type*⟩[‘`,`’ ⟨*type*⟩]*‘`]`’ ]* [‘`(`’[⟨*smp*⟩[‘`,`’ ⟨*smp*⟩]*]‘`)`’]* ]

⟨***lambda***⟩ ::= ‘`|`’[⟨*param*⟩ [‘`,`’ ⟨*param*⟩]* ]‘`|`’[‘`->`’⟨*type*⟩]‘`=`’⟨*smp*⟩

Expressions

⟨*atom*⟩ ::= ⟨*int*⟩ | ⟨*bool*⟩ | ⟨*char*⟩ | ⟨*string*⟩ | ⟨*null*⟩
  | '`(`'⟨*smp*⟩'`)`'
  | ⟨*ident*⟩['`.`'⟨*ident*⟩]*

⟨*tight*⟩ ::= ⟨*app*⟩['`|>`'⟨*app*⟩]
  | '{'⟨*exp*⟩'}'

⟨*utight*⟩ ::= [⟨*op*⟩]⟨*tight*⟩

⟨*smp*⟩ ::= ⟨*utight*⟩[⟨*op*⟩⟨*utight*⟩]
  | '`if`' '`(`'⟨*smp*⟩'`)`' ⟨*smp*⟩ ['`else`' ⟨*smp*⟩]
  | '`List`' | '`Tuple`' | '`Array`' | '`Set`' '{'[⟨*smp*⟩['`,`' ⟨*smp*⟩]*]'}'
  | '`Dict`' '{'[⟨*smp*⟩'`:`'⟨*smp*⟩['`,`' ⟨*smp*⟩'`:`'⟨*smp*⟩]*]'}'
  | ⟨*match*⟩
  | ⟨*alias*⟩
  | ⟨*adt*⟩
  | ⟨*record*⟩
  | ⟨*typeclass*⟩
  | ⟨*instance*⟩
  | ⟨*prog*⟩
  | ⟨*lambda*⟩

⟨*exp*⟩ ::= ⟨*smp*⟩[⟨*terminator*⟩⟨*exp*⟩]
  | ['`lazy`'] '`val`' ⟨*ident*⟩['`:`'⟨*type*⟩] '`=`' ⟨*smp*⟩⟨*terminator*⟩⟨*exp*⟩
  | '`include`' ⟨*file*⟩⟨*terminator*⟩⟨*exp*⟩