HOME : IMPORTPACKAGES PACKAGENAME HOME SATISFIES REQUIRES NAMESPACE

IMPORTPACKAGES: IMPORTKWD IDENTIFIER | IMPORTKWD IDENTIFIER IMPORTPACKAGES | IMPORT PACKAGES ASKWD IDENTIFIER | IMPORT PACKAGES ASKWD IDENTIFIER IMPORTPACKAGES

#namespace

PACKAGENAME: PACKAGEKWD IDENTIFIER

NAMESPACE: NAMESPACEKWD IDENTIFIER LEFT BRACE CLASS RIGHT BRACE

FACADE:FACADEKWD LEFTBRACKET FACADEOPERATORS RIGHTBRACKET

#facade interface, interface, interface, interface expression

FACADEOPERATORS: IDENTIFIER LEFTTBRACE EXPRESSION RIGHTBRACE| FACEDEOPERATORS IDENTIFIER LEFTBRACE EXPRESSION RIGHTBRACE | FACADDEOPERATORS

SATISFIES: SATISFIESKWD | EMPTY

REQUIRES: REQUIRESKWD | EMPTY

#class definition

CLASS: CLASS\_HEADER\_TYPE: CLASS\_DEFINITION

CLASS\_HEADER: TYPE\_CONSTRUCTOR NEWLINE PROTOTYPE NEWLINE SHARPSYMBOL CLASSNAME

CLASSNAME:SEALED ABSTRACT MIXIN CLASSKEYWD IDENTIFIER DIAMOND META DECORATOR THROWS|

SEALED ABSTRACT MIXIN CLASSKEYWD LAMBDAKWD DIAMOND META DECIRATOR THROWS CLASS |: DECORATOR CLASS\_HEADER PRIMARY\_CONSTRUCTOR MIXIN\_CLASSES INHERITANCE CLASS\_DEFINITION

#Primary constructor lets you define the constructor at the top of th eclass instead of as a function

PRIMARY\_CONSTRUCTOR: LPAREN PRIMARY RPAREN

PRIMARY: CLASSNAME IDENTIFIER | COMMA CLASSNAME IDENTIFIER PRIMARY

#class header

SEALED: SEALEDKWD

#prototype makes the method cloneable

PROTOTYPE: PROTOTYPEKWD | EMPTY

#abstract - declares as an abstract class

ABSTRACT: ABSTRACTKWD | EMPTY

#mixin. A short or abstract class added to a class The mixin keyword decalres it as a mxiin class

MIXIN: MIXINKWD | EMPTY

/// (x:MixinClass)

MIXIN\_CLASSES: LPAREN IDENTIFIER COLON IDENTIFIER RPAREN |   
 LPAREN IDENTIFIER COLON IDENTIFIER RPAREN MIXIN\_CLASSES

#throws

THROWS: THROWSKEWD IDENTIFIER

#meta - declares the class as a meta class

META: METAKWD | EMPTY

#inheritance

INHERITANCE: EXTENDS IDENTIFIER

| EXTENDS IDENTIFIER COMMA INHERITANCE

| IMPLEMENTS IDENTIFIER

| IMPLEMENTS IDENTIFIER COMMA INHERITANCE

#Diamond operator - Use for template classes. (Generics). Through the rest of the class then the class type is <>.

DIAMOND: DIAMONDKWD | EMPTY

#type constructor

TYPE\_CONSTRUCTOR : INHTYPE

| INHTYPE ARROW TYPE\_CONSTRUCTOR

;

#body of the class

CLASS\_DEFINITION: LEFT\_BRACKET CLASS\_CONTENTS RIGHT\_BRACKET ENDCLASSKEYWORD

CLASS\_CONTENTS: MEMBER\_VARIABLE\_DEFINITION | CLASS\_CONTENTS  
 MEMEBER\_METHOD\_DEFINITION | CLASS\_CONTENTS

INITENTITYPOOL: | CLASS CONTENTS | CLASS

#class is there for innner classses

INITENTITYPOOL: LEFTBRACE NEWKEYWRD ENTITYPOOLKEYWRD RIGHTBRACE

#member variable defiinition

MEMBER\_VARIABLE\_DEFINITION: TRANSIENT STREAM\_INIT | MEMBER\_VARIABLE\_DEFINITION

| MEMBER\_VARIABLE\_INIT MEMBER\_VARIABLE\_DEFINITION

STREAM\_INIT\_VARIABLE\_DEFINITION: LEFTBRACE STREAMKWD IDENTIFIER IDENTIFIER RIGHTBRACE

#initialize member variable. Can use let  
MEMBER\_VARIABLE\_INIT:PREFIXVAR TYPENAME OBJECTNAME STREAMOPERATORLEFT IDENTIFIER STREAMOPERATORRIGHT IDENTIFIER | LET STREAMOPERATORLEFT IDENTIFIER STREAMOPERATORIGHT IDENTIFIER |

STREAMOPERATORLEFT IDENTIFIER STREAMOPERATORRIGHT NEUTRAL

PREFIXVAR: TRANSIENT PREFIXVAR | AUTO PREFIXVAR | STATICF PREFIXVAR | FINAL | EMPTY

TRANSIENT: TRANSIENTKWD

OBJECTNAME: IDENTIFIER

| VOID   
 | SHORT

| CHAR  
 | LONG  
 | BOOLEAN  
 | INT

| DOUBLE

| FLOAT

| FIXED

|UINT  
| USHORT

| ULONG  
| ENUM

| COLLECTION  
| SET

|ARRAY

|ARRAYLIST  
|LINKEDLIST

|DOUBLYLINKEDLIST

|DOUBLYLINKEDARRAYLIST  
|VECTOR  
|HASHTABLE  
|MAP  
|MULTIMAP  
|STRING #immutable

|OBJECT  
|LIST #mutable

|TREE  
|STACK  
|QUEUE

|VECTOR

|MATRIX

|SEQUENCE

|TUPLE #immutable

|

#member method definition

MEMBER\_METHOD\_DEFINITION: ANNOTATE ACCESSPRIV STATIC FUNCTKIND CONSTRUCT EXPLICIT IMPLICIT FRIEND OPTIONAL REQUIRED VIRTUAL OVERRIDE DELEGATE LISTENER RETURNCLASS METHOD\_NAME BRIDGE | VISITOR | VISITEE METHOD\_PARAMETERS CALLER LBRACKET METHOD\_CONTENTS RBRACKET WHERECL

METHOD\_NAME: IDENTIFIER | OPERATOR | CUSTOMOPERATOR

CUSTOMOPERATOR: LPAREN IDENTIFIER RPAREN LBRACKET EXPRESSION RBRACKET

#visitor - the visitor and visitee patterns wire together the two sides of the pattern,

VISITOR: VISITORKWD

VISITEE: VISITEEKWD

#BRidge design pattern. Using this method and only this method will be able to

#to connect to another class. Both classes must have the same function signature

#colon bridge identifier (bridge to identifier)

BRIDGE: COLON BRIDGEKWD IDENTIFIER

#operator can be any operator

RETURNCLASS: CLASSNAME | CLASSNAME COMMA RETURNCLASS

WHERECL: WHERE\_CLAUSE | EMPTY

ACCESSPRIV: PUBLICKWD | PROTECTEDKWD | INTERNAKWDL | PRIVATEKWD

FUNCTKIND: FUNCTIONTYPEKWD | PROCEDURETYPEKWD

#explicit (has to do with type conversion)

EXPLICIT: EXPLICITKWD | EMPTY

#implicit has to do with htype conversion

IMPLICIT: IMPLICITKWD | EMPTY

#decalres the method as a constructor

CONSTRUCT: CONSTRUCTKWD | EMPTY

#delegate -label function as a delegate

DELEGATE = DELEGATEKWD

STATIC: STATICKWD | EMPTY

METHOD\_PARAMETERS: LEFTPAREN M\_PARAM RIGHTPAREN:: |

LEFTPAREN ARGC RIGHTPAREN

M\_PARAM: CLASS\_IDENTIFIER: CONST CLASS\_IDENTIFIER IDENTIFIER COMMA

| CONST CLASS\_IDENTIFIER IDENTIFIER |

CONST CLASS\_IDENTIFIER IDENTIFIER EQUALS IDENTIFIER | CONST CLASS\_IDENTIFIER

| CONST DEEP IDENTIFIER

| CONST NEED IDENTFIER

| CONST PASSBYREFERENCE IDENTIFIER

| CONST PASSBYREFBYVAL

#pass by reference by value

#like Java.

PASSBYREFBYVAL: IDENTIFIER

#pass by reference for when it doesn’t make sense to copy the whole object

PASSBYREFERENCE: ADDRESSOFOPERATOR IDENTIFIER

#pass by need does a promise like haskell

NEED: NEEDKWD

#shallow copy by default. Deep does deep copy

DEEP: DEEPKWD

//pass by constant reference

CONST: CONSTKWD | EMPTY

ARGC: STAROPERATOR ARGCKKWD IDENTIFIER

#method body

METHOD\_CONTENTS: EXPRESSION METHOD\_CONTENTS | RETURN | EMPTY

#if method contents are empty then it is an abstract method

EXPRESSION: LBRACE STATEMENT RBRACE | LBRACE STATEMENT RBRACE EXPRESSOIN | EMPTY

#identifier can be a tuple to return multiple things

RETURN: RETURNKWD IDENTIFIER SEMICOLON

##

STATEMENT: STREAMOPERATIONS | CLASSINSTANTIATION | MISCKEYWORDS| LISTCOMPRHENSION | THREADING | MESSAGEPASSING | DEBUG | ASSERT | TRY| CATCH | FINALLY | LASTKWD | TAG | MUULTICATCH | THROW | CLASSCONVERSION |

STRUCT | UNION | GLOBAL | COMMANDEXECUTE | DEL | SIZEOF | EXTERN | WHILELOOP | TEXTLITERAL | STRINGCONCAT | UNINITIALIZED | STRINGWILDCARD | MATH | EVALUATION | COMMAOPERATOR | MEMBERSHIPTEST | CONVO | LISP | IFELSE | DOWHILE | DOUNTIL | WHILE | WAIT| FORLOOP | FOR | FORSIMPLE | GOTO | SWITCH |

SWITCHENHANCED | LABEL | RANGE | LEN | WITH | IN | INNNERCLASS | ANONINNERCLASS | OBJECTOPERATORS | GETINSTANCE | GETCLASS | GET | SET | EXPRESSBODYMETHOD | SUPER | RENAME | REFLECTION | SHALL | MEMBER\_METHOD\_DEFINITION | PRECONDITION | POSTCONDITION | FUNCTIONCOMPOS | FUNCTIONCONCAT, REWRITE, ENDREWRITE, WRITE, ENDWRITE,

LAMBDAOP | TOSTRING | FROMSTRING | CONVERT | ADDRESSOF | DEREFERENCE | POINTER\_ARITH | PRINT | SLICING | COPY | COPYEQUIVALENT | COPYCONTENT |  
COPYBASECLASS | CONS | BASE | TAIL | ABSTRACTMEMBER\_METHOD\_DEFINITION | ABSTRACTFACTORY | ACCESSFACTORY | ACCESSABSTRACTFACTORY | FACTORY | CLONE | WRAPPED\_EXPRESSION | STREAM\_INIT\_WRAPPED\_DEFINITION | WRAPPERCLASS\_HEADER | ADAPTER | DECORATE | COMPOSITE | FLYWEIGHT PROXY | CHAIN | COMMAND | COMMANDINVOKER | MEDIATOR | STOREMOMENTO | RETREIVEMOMENTO | STRATEGY | VISITOR | SINGLETON| STATEFUL| EQUIV

###

MISCKEYWORDS: VOLTATILEKWD

| LAZYKWD  
 | DYNAMICKWD

| MEMORYMANAGEMENT

| FROZENKWD  
 | THAWKWD

| THISKWD

| RESIDEKWD

| BASEKWD

MEMORYMANAGMENT: RELEASEKWD

| RETAINKWD  
 | AUTORELEASEKWD

LISTCOMPREHENSION:: LISTCOMP

| LISTCOMP LISTCOMPREHENSION

LISTCOMP: LEFTBRACKET VARTUPLE FORSTMTNT IDENTIFIER INTEGER RIHTBRACKET

GENERATOR: GENRTR | GENRTR GENERATOR

GENRTR: LEFTPAREN VARTUPLE FORSTMNT IDENTIFIER INTEGER RPAREN

THREADING: LOCK LPAREN IDENTIFIER RPAREN |

SYNC

#message passing

MESSAGEPASSING: LEFTBRACKET FUNCTIONKWD IDENTIFIER COLON DESTINATIONKWD: IDENTIFIER |  
 LEFTBRACKET VARIABLEKWD IDENTIFER COLON DESTINATIONKWD IDENTIFIER |  
 LEFTBRACKET FUNCTIONKWD IDENTIFIER COLON RECEPTIONKWD IDENTIFIER |  
 LEFTBRACKET VARIABLEKWD IDENTIFIER COLON RECEPTIONKWD IDENTIFIER

#debugging

DEBUG: PRINTDEBUGSTMT LPAREN QUOTES IDENTIFIER QUOTES RPAREN SEMICOLON

#assert

ASSERT: ASSERTKWD LPAREN IDENTIFIER BOOLEANEQUALS IDENTIFIER RPAREN

#try catch

TRY: TRYKWD LEFTBRACE TRYEXPRESSIONRIGHTBRACE

CATCH: CATCHKWD LPAREN BOOLEANSTATEMENT RPAREN LEFTBRACE CATCHEXPRESSIONRIGHTBRACE

TRYEXPRESSION: EXPRESSION |

AWAITKWD |

AUTOFALLKWD

FINALLY: FINALLYKWD LEFTBRACE FINALLYEXPRESSION RIGHTBRACE

#Tag

TAG: TAGKWD IEDENTIFIER

#Multicatch

MULTICATCH: TAGKWD LPAREN EXCEPTIONS RPAREN

EXCEPTIONS: IDENTIFIER |

GUARD IDENTIFIER

THROW: THROWKEYWD LPAREN IDENTIFIER RPAREN

CLASSCONVERSION:: IDENTIFIER OBJEQUALS LPAREN IDENTIFIER RPAREN IDENTIFIER

#strucfts

STRUCT: ACCESSPRIV STRUCTKWD LBRACKET EXPRESSION RBRACKET

#unions

UNION: ACCESSPPRIV UNIONKWD LBRACKAET EXPRESSION RBRACKET

#global

GLOBAL: GLOBALKWD IDENTIFIER SEMICOLON

#pass

PASS: does not hing

COMMANDEXECUTE: COMMANDEXECUTEKWD LPAREN QUOTE STRING QUOTE RPAREN SEMICOLON

#del

DEL: DELKWD IDENTIFIER SEMICOLON

#boolean expression

BOOLEANSTMT: LEFTOPERAND BOOLOPERATOR RIGHTOPERAND |  
 BOOOLCONJUNCTION LEFTOPERAND BOOLOPERATOR RIGHTOPERAND BOOLEANSTMT

BOOLCONJUNCTION: ANDKWD | ORKWD | XORWLD | NOT

BOOLOPERATOR: EQUALSKWD | DOESNTEQLKWD | LESSEQUALKWD | GREATEREQUALKWD | LESSKWD | GREATERKWD

#scope

SIZEOF: SIZEOFKWD IDENTIFIER SEMICOLON

#exterm

EXTERN: EXTERNKWD IDENTIFIER SEMICOLON

#while loop

WHILE\_LOOP: WHILEKWD LPAREN BOOLEANSTMT RPARENT LEFTBRACKET WHILEEXPRESSION RIGHTBRACKET LOOPELSE

WHILEEXPRESSION: EXPRESSION | CONTINUE EXPRESSION | RESIDUE | RESOLVE LEAP | INCLUSIONKWD IDENTIFIER EQUALS IDENTIFIER

LEAP: LEAPKWD STATEMENT | EMPTY

CONTINUE: CONTINUESTMT SEMICOLON

#do while loop

DOWHILE: DOKWD LEFTBRACKET WHILEEXPRESSIONRIGHTBRACKET WHILEKWD LPAREN BOOLEANSTMT RPAREN LOOPELSE

#unti lloop

DOUNTIL: DOKWD LEFTBRACKET WHILEEXPRESSION RIGHTBRACKET UNTILKWD LPAREN BOOLEANSTMT RPAREN LOOPELSE

#loop else

LOOPELSE: ELSEKWD LBRACKET EXPRESSION RBRACKET

#wait function

WAIT: WAITKWD LPAREN BOOLEANSTMT RPAREN

#if else

IFELSE: IFCLAUSE | IFELSECLAUSE | ELSEIFCLAUSE

IFCLAUSE: IFKWD LPAREN BOOLEANSTMT RPAREN LBRACKET EXPRESSION RIGHTBRACKET

IFELSECLAUSE: IFKWD LPAREN BOOLEANSTMT RPAREN LBRACKET EXPRESSION RIGHTBRACKET ELSEKWD LBRACKET EXPRESSION RIGHTBRACKET

ELSEIFCLAUSE: IFCLAUSE ELSEIFKWD LPAREN BOOLEANSTMT RPAREN LBRACKET EXPRESSION RIGHTBRACKET | ELSEIFCLAUSE

ELSEIFKWD LPAREN BOOLEANSTMT RPAREN LBRACKET EXPRESSION RIGHTBRACKET

#where clause

WHERECLAUSE: WHEREKWD LPAREN BOOLEANSTMT RPAREN

#for loop enhanced

FORLOOP: FOREACHKWD LPAREN IDENTIFIER IDENTIFIER COLON IDENTIFIER RPAREN LBRACKETWHILEEXPRESSION RBRACKET

FOR: FOR KWD LPAREN BOOLEANSTMT SEMICOLON BOOLREANSTMT SEMICOLON BOOLEANSTMT RPAREN LBRACKET WHILEEXPRESSION RBRACKET

#simpler for loop

FORSIMPLE:: FORKYWRD LPAREN BOOLEANSTMT) RPAREN LBRACKET WHILEEXPRESSION RBRACKET

GOTO: GOTOKWD IDENTIFIER

#switch

SWITCH: SWITCHKWD LPAREN IDENTIFIER RPAREN LBRACKET SWITCHBLOCK RBRACKET

SWITCHBLOCK: CASE BLOCK | SWITCHBLOCK CASEBLOCK

CASEBLOCK: CASEKWD LPAREN BOOLEANSTMT RPAREN COLON CASEEXPRESSION |

CASEKWD LPAREN PARTIALBOOLEAN RPAREN COLON CASEEXPRESSION

CASEEXPRESSION: EXPRESSION |: EXITKWD | PASSKWD | BREAKKWD

#enhanced case expression

SWITCHENHANCED: CASEENHANCED | CASEENHANCED SWITCHENHANCED

CASEENHANCED: LEFTBRACE PARTIALBOOLEAN RIGHTBRACE LEFTBRACE RETURNSTRINGKWD EQUALS IDENTIFIER FALL RIGHTBRACE

FALL: FALLKWD | CASEEXPRESSION | EMPTY

PARTIALBOOLEAN: BOOLEANOPERATOR IDENTIFIER

#text literal

TEXTLITERAL: ATSIGNKWD COLON QUOTE STRING QUOTE SEMICOLON

#strings

STRINGCONCAT: IDENTIFIER LEFTPAREN CONCATKWD RIGHTPAREN IDENTIFIER |  
 IDENTIFIER LEFTPAREN CONCATKWD RIGHTPAREN IDENTIFIER STRINGCONCAT

#unintialized conditional

UNINITIALIZED: IFKWD LPAREN IDENTIFIER DOUBLEQUESTIONKWD RPAREN SEMICOLON

#string wildcard

STRINGWILDCARD:: STAR\_OPERATOR IDENTIFIER

#math operators

MATH: BINMATH TRIMATH UNMATH

| BINMATH UNMATH TRIMATH  
 | TRIMATH UNIMATH BINMATH  
 | TRIMATH BINMATH UNIMATH

| UNIMATH BINMATH TRIMATH

| UNIMATH TRIMATH BINMATH

| EMPTY

BINMATH: BINARYMATH | BINARYMATH BINMATH | EMPTY

UNMATH: UNARYMATH | UNARYMATH UNMATH | EMPTY

TRIMATH: TRINARYMATH | TRINARYMATH TRIMATH | EMPTY

UNARYMATH: IDENTIFIER POWERKWD IDENTIFIER

| NEGATIONKWD IDENTIFIER

| POSIGATIONKWD IDENTIFIER

| PERCENT IDETIFIER

| FACTORIAL IDENTIFIER

| IDENTIFIER PLUSPLUSPREFIX

| PLUSPLUSPOSTFIX IDENTIFIER

| MINUSMINUSPREFIX IDENTIFIER

| IDENTIFIER MINUSMINUSPOSTFIX

| COMPOUNDADDITION IDENTIFIER

| COMPOUNDSUBTRACTION IDENTIFIER  
 |COMPOUNDMULTIPLICATION IDENTIFIER  
 |COMPOUJNDDIVISION IDENTIFIER

|EXPONENTIAL IDENTIFIER

|ABSOLUTE IDENTIFIER  
|

BINARYMATH: IDENTIFIER ADDITIONKWD IDENTIFIER

| IDENTIFIER SUBTRACTION IDENTIFIER  
 | IDENTIFIER INTDIV IDENTIFIER  
 | IDENTIFIER FLOATDIV IDENTIFIER  
 | IDENTIFIER MULT IDENTIFIER  
 | IDENTIFER MOD IDENTIFIER  
 | IDENTIFIER REM IDENTIFIER  
 | IDENTIFIER SHIFTLEFT IDENTIFIER  
 | IDENTIFIER SHIFTRIGHT IDENTIFIER  
 | IDENTIFIER EQUALS IDENTIFIER  
 | IDENTIFIER ASSIGNMENT IDENTIFIER  
 | IDENTIFIER LESSTHAN IDENTIFIER  
 | IDENFITER GREATERTHAN IDENTIFIER  
 | IDENTIFIER LESSTHANEQUALS IDENTIFER  
 | IDENTFIER GREATERTHANEQUALS IDENTIFIER  
 | IDENTIFIER NOTEQUALTO IDENTIFIER  
 |

#evaluation

EVALUATION: BACKTICKOPERATOR IDENTIFIER

| FRONTTICKOPERATOR IDENTIFIER

#back tick automatically evaluates entity

#front tick turns an expression into a list

#comma operator

#comma can be used to join two statements in braces

COMMAOPERATOR: STATEMENT COMMA STATEMENT

MEMBERSHIPTEST: INMETHOD IDENTIFIER IDENTIFIER |  
 INCLASS IDENTIFIER IDENTIFIER |  
 EMPTY

#turns a string into a list

CONVO: CONVOKWD IDENTIFIER IDENTIFIER

LISP: CARKWD IDENTIFIER | CDRKWD | EMPTY

CONS: CONSKWD IDENTIFIER IDENTIFIER

LABEL: LABELKWD IDENTIFIER #a spot for goto to jump to

#range and len

RANGE: RANGEKWD LPAREN RANGECALC RPAREN

RANGECALC: IDENTIFIER COMMA IDENTIFIER

| COMMA RANGECALC IDENTIFIER

LEN: LENKWD LPAREN IDENTIFIER RPAREN

#with comand’

WITH: WITHKWD LPAREN IDENTIFIER RPAREN LBRACKET EXPRESSIONS RBRACKET

#in command

IN: IDENTIFIER INKWD IDENTIFIER

#innerclass

INNER\_CLASS: CLASS

#anonymous inner class

ANONINNERCLASS : ANONCLASS\_HEADER CLASS\_DEFINITION

#class header

ANONCLASS\_HEADER: TYPE\_CONSTRUCTOR NEWLINE SHARPSYMBOL ANONCLASSNAME

ANONCLASSNAME:SEALED CLASSKEYWD INHERITANCE THROWS |

SEALED CLASSKEYWD LAMBDAKWD INHERITANCE THROWS

#object operators

OBJECTOPERATORS: IDENTIFIER ISKYWRD IDENTIFIER |  
 IDENTIFIER ISNOTKWD IDENTIFIER |  
 IDENTIFIER IDENTIFIER ISNOTHINGKWD |  
 TYPEOF LPAREN IDENTIFIER RPAREN |  
 INSTANCEOF LPAREN IDENTIFIER RPAREN |

GETTYPEKWD IDENTIFIER |  
 EMPTY

#get instance of a class from entity pool

GETINSTANCE: ENTITYPOOLKWD DOTGETKEYWORD INSTANCEKEWD LPAREN IDENTIFIER RPAREN

#get class from object

GETCLASS: IDENTIFIER DOTGETCLASSKEYWORD

#get keyword

GET: GETKWRD LBRACE EXPRESSION RBRACE

SET: SETKWQRD LBRACE EXPRESSION RBRACE

#Expression body methods

EXPRESS\_BODY\_METHOD: IDENTIFIER BIGARROW LBRACE EXPRESSION RBRACE

##interfaces

INTERFACE: INTERFACETYPECONSTRUCTOR INTHEADER LBRACKET NTERFACEBODY RBRACKET

INTERFACETYPECONSTRUCTOR: IDENTIFIER | LITTLEARROW INTERFACETYPECONSTRUCTOR IDENTIFIER

INTHEADER: SHARPSYMBOL INTERFACEKWD IDENTIFIER ACCESS IMPLEMENTS

ACCESS: PUBLIC | PRIVATE | INTERNAL | PROTECTED

IMPLEMENTS: IMPLEMENTSKWD IDENTIFIER

INTERFACEBODY: METHODSTUBS | INTERFACEBODY METHODSTUBS | VARIABLESTUBS | VARIABLESTUBS INTERFACEBODY

METHODSTUBS:

MEMBER\_METHOD\_STUB: ACCESSPRIV FUNCTKIND STATIC RETURNCLASS

METHOD\_PARAM\_STUB: LBRACKET METHOD\_CONTENTS RBRACKET

METHOD\_PARAMETERS: LEFTPAREN M\_PARAM RIGHTPAREN:: |

LEFTPAREN ARGC RIGHTPAREN

VARIABLESTUBS

MEMBER\_VARIABLE\_STUB IDENTIFIER STREAMOPERATOR IDENTIFIER

#flyweight

abstractfctory myFactory = new <>();

#fllyweight interface implementation factory

#flyweight class implements interface from factory code for flyweight

FLYWEIGHT: FLYWEIGHTKWD IDENTIFIER IMPLEMENTS INTERFACE FROM IDENTIFIER EXPRESSION

#super

SUPER: SUPERKWRD

#rename

RENAME: RENAMEKWD IDENTIFIER BIGARROW IDENTIFIER

#reflection

REFLECTION: CLASSKWD LEFTANGLEBRACKET QUESTIONOPERATOR RIGHTANGLEBRACKET EQUALS QUOTE IDENTIFIER QUOTE

#generics (instantiation)

#CLASSTYPE is the template argument

#shall keyword

SHALL: CLASSKWD IDENTIFIER SHALLKWD IDENTIFIER

#neutral keyword

NEUTRAL: NEUTRALKWD

#friend keyword

FRIEND: FRIENDKWD IDENTIFIER

#optional annotation

OPTIONAL: OPTIONALANNOTATION | EMPTY

#requried annotation

REQUIRED: REQUIREDANNOTATION | EMPTY

#FOr purposes of clarity you can mark in the function defintion line what may call the function

CALLER: COLON CALLERKWD LPAREN CALLERS RPAREN

CALLERS: IDENTIFIER | COMMA IDENTIFIER CALLERS

#precondition and postcondition

PRECONDITION: DOLLARSIGN PRECONDITIONKWD LPAREN BOOLEANSTMT RPAREN  
POSTCONDITION: DOLLARSIGN POSTCONDITIONKWD LPAREN BOOLEANSTMT RPAREN

#virtual and overrride

VIRTUAL: VIRTUALKWD #can be overridden

OVERRIDE: OVERRIDEKWD #overides a virtual function

#function composition

FUNCTIONCOMPOS: IDENTIFIER COLON IDENTIFIER

#function concatenation

FUNCTIONCONCAT: IDENFTIFIER ATSIGN IDENTIFIER

#self modifying code

REWRITE: LBRACKET LBRACKET REWRITEKWD EQUALS IDENTIFIER RBRACKET RBRACKET

ENDREWRITE: LBRACKET LBRACKET ENDREWRITEKWD RBRACKET RBRACKET

WRITE: LBRACKET LBRACKET WRITEKWD EQUALS IDENTIFIER RBRACKET RBRACKET

ENDWRITE: LBRACKET LBRACKET ENDWRITEKWD RBRACKET RBRACKET

#lambdaop

LAMBDAOP: LAMBDAARGUMENTS STAR\_OP BIG\_ARROW EXPRESSION

LAMBDAARGUMENTS: IDENTIFIER | COMMA LAMBDARGUMENTS

#TOSTRING

TOSTRING: IDENTIFIER TOSTRINGKWD

#FROMSTRING

FROMSTRING: IDENTIFIER FROMSTRINGKWD

#CONVERT

CONVERT: IDENTIFIER CONVERT LPAREN IDENTIFIER RPAREN

#Trigger - event handling

TRIGGER: EVENTKWD FIREKWD TRIGGERKWD LPAREN IDENTIFIER RPAREN

#listener event handling

LISTENER: LISTENERKWD LPAREN LPAREN LBRACKET IDENTIFIER RPAREN RPAREN

#address of (for pointers)

ADDRESSOF: ADDRESSOFOPERATOR IDENTIFIER

#dereference pointer

DEREFERENCE: STAR\_OPERATOR IDENTIFIER

#pointer arithmetic

POINTERARITH: IDENTIFIER PLUSOPERATOR OBJECTNAME

#print statement

PRINT: PRINTKWD LPAREN QUOTE PRINTSTRING QUOTE COMMA SUBSTITUTE RPAREN

PRINTSTRING: TEXT | TEXT FORMATTING PRINTSTRING | TEXT FORMATTING ESCAPE PRINTSTRING

TEXTFORMATTING:: INTEGERFORMAT DECIMALFORMAT OCTALFORMAT HEXFORMAT FLOATINGFORMAT CHARFORMAT STRINGFORMAT POINTERFORMAT

#format specifiers

INTEGERFORMAT: IFORMAT | EMPTY  
DECIMALFORMAT: DFORMAT | EMPTY  
OCTALFORMAT: OFORMAT | EMPTY

HEXFORMAT: HFORMAT | EMPTY  
FLOATINGFORMAT: FFORMAT | EMPTY  
CHARFORMAT: CFORMAT | EMPTY  
STRINGFORMAT: SFORMAT | EMPTY  
POINTERFORMAT: PFORMAT | EMPTY

#escape strings

ESCAPE: TAB BACKSPACE NEWLINE CARRIAGERETURN FORMFEED SINGLEQUOTE DOUBLEQUOTE BACKSLASH ALARM HORIZONTALTAB VERTICALTAB QUOTATIONMARK

TAB: TABX | EMPTY

BACKSPACE: BACKX | EMPTY  
NEWLINE: NEWX | EMPTY  
CARRIAGERETURN: RETURNX | EMPTY

FORMFEED: FORMFEEDX | EMPT Y

SINGLEQUOTE: SINGLEQUX | EMPTY  
DOUBLEQUOTE: DOUBLE QUX | EMPTY  
BACKSLASH: BACKSLASHX | EMPTY

ALARM: ALARMX | EMPTY

HORIZONTAL TAB: HORIZONTALTABX | EMPTY  
VERTICALTAB: VERTICALTABS | EMPTY  
QUOTATIONMARK: QUOTATIONMARKX | EMPTY

#slicing on arrays for subarrays

SLICING: IDENTIFIER LBRACKET COLON IDENTIFIER RBRACKET |

LBRACKET IDENTIFIER COLON RBRACKET

#shallow an deep copy

COPY: IDENTIFIER EQUALS IDENTIFIER | IDENTIFIER SHALLOW IDENTIFIER

#string equivalent

COPYEQUIVALENT: IDENTIFIER ADDRESSOF EQUALS IDENTIFIER

COPYCONTENT: IDENTIFIER EQUALS IDENTIFIER  
COPYBASECLASS IDENTIFIER EQUALS DOLLARSIGN IDENTIFIER

#recursion

BASE: BASEKWD LEFTBRACE EXPRESSION RIGHTBRACE

TAIL: TAILKWD LEFTBRACE EXPRESSION RIGHTBRACE

#annotations

ANNOTATE: ATSIGN LBRACKET EXPRESSION RBRACKET NEWLINE | EMPTY

##DESIGN PATTENS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#member method definition

ABSTRACTMEMBER\_METHOD\_DEFINITION: ANNOTATE AACCESSPRIV ASTATIC AFUNCTKIND AEXPLICIT AIMPLICIT FRIEND OPTIONAL REQUIRED VIRTUAL OVERRIDE LISTENER RETURNCLASS ABSTRACKWDMETHOD\_NAME A METHOD\_PARAMETERS CALLER LBRACKET METHOD\_CONTENTS RBRACKET WHERECL

AWHERECL: WHERE\_CLAUSE | EMPTY

AACCESSPRIV: PUBLIC | PROTECTED | INTERNAL | PRIVATE

AFUNCTKIND: FUNCTIONTYPE | PROCEDURETYPE

#explicit (has to do with type conversion)

AEXPLICIT: EXPLICITKWD | EMPTY

#implicit has to do with htype conversion

AIMPLICIT: IMPLICITKWD | EMPTY

ASTATIC: STATICKWD | EMPTY

AMETHOD\_PARAMETERS: LEFTPARENA M\_PARAM RIGHTPAREN:: |

LEFTPAREN AARGC RIGHTPAREN

AM\_PARAM: CLASS\_IDENTIFIER: CONST CLASS\_IDENTIFIER IDENTIFIER COMMA

| CONST CLASS\_IDENTIFIER IDENTIFIER |

CONST CLASS\_IDENTIFIER IDENTIFIER EQUALS IDENTIFIER | CONST CLASS\_IDENTIFIER

| CONST DEEP IDENTIFIER

| CONST NEED IDENTFIER

| CONST PASSBYREFERENCE IDENTIFIER

| CONST PASSBYREFBYVAL

AMETHOD\_CONTENTS: EMPTY

# abstractfctory myFactory = new <>();

ABSTRACTFACTORY: ABSTRACTFACTORYKWD IDENTIFIER IDENTIFIER EQUALS NEWKWD DIAMONDOPERATOR LPAREN RPAREN SEMICOLON

#access

ACCESSABSTRACTFACTORY: IDENTIFIER DOTGET LPAREN IDENTIFIER RPAREN

#\*\*\*\*\*\*\*\*\*\*FACTORY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#member method definition

FACTORYMEMBER\_METHOD\_DEFINITION: ANNOTATE ACCESSPRIV STATIC FUNCTKIND EXPLICIT IMPLICIT FRIEND OPTIONAL REQUIRED VIRTUAL OVERRIDE LISTENER RETURNCLASS METHOD\_NAME METHOD\_PARAMETERS CALLER LBRACKET METHOD\_CONTENTS RBRACKET WHERECL

# factory myFactory = (s) new <>();

#initialize

FACTORY: FACTORYKWD IDENTIFIER IDENTIFIER EQUALS STREAM NEWKWD DIAMONDOPERATOR LPAREN RPAREN SEMICOLON

#access

ACCESSFACTORY: IDENTIFEIR DOTGET LPREN IDENTIFIER RPAREN

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PROTOTYPE

# clone newclassname = prototypeclass

CLONE: CLONEKWD = STREAM IDENTIFIER

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WRAPPER CLASS

WRAPPED: WRAPPERDWD

WRAPPER: WRAPPERKWD

WRAPPERCLASS\_HEADER: TYPE\_CONSTRUCTOR NEWLINE PROTOTYPE NEWLINE WRAPPER NEWLINE SHARPSYMBOL WRAPPERCLASSNAME

WRAPPERCLASSNAME:SEALED ABSTRACT MIXIN CLASSKEYWD IDENTIFIER DIAMOND META THROWS|

SEALED ABSTRACT MIXIN CLASSKEYWD LAMBDAKWD DIAMOND META THROWS CLASS : CLASS\_HEADER PRIMARY\_CONSTRUCTOR MIXIN\_CLASSES INHERITANCE WRAPPERCLASS\_DEFINITION

WRAPPERCLASS\_DEFINITION: LEFT\_BRACKET WRAPPERMETHOD\_CONTENTES RIGHT\_BRACKET ENDCLASSKEYWORD

WRAPPERMETHOD\_CONTENTS: WRAPPEREXPRESSION | METHOD\_CONTENTS | RETURN | EMPTY

#wrapped classname identifier

STREAM\_INIT\_WRAPPED\_DEFINITION: LEFTBRACE STREAMKWD IDENTIFIER IDENTIFIER RIGHTBRACE

#initialize wrapped expression

WRAPPEDEXPRESSION: LEFTBRACE WRAPPERKWD STREAMKWD IDENTIFIER IDENTIFIER RIGHTBRACE

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ADAPTER\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Adapter identifier expresion sourceclassname destinationclassname

ADAPTER: ADAPTERKWD IDENTIFIER EXPRESSION IDENTIFIER IDENTIFIER

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COMPOSITE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#Accesspriv composite name = new diamondoperator lparen rparen

COMPOSITE: ACCESSPRIV COMPOSITEKWD EQUALS NEW DIAMONDOPERATOR LPAREN RPAREN

ACCESSPRIV: PUBLIC | PRIVATE | INTERNAL | PROTECTED

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DECORATOR\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

DECORATE: DECORATEKWD CLASSTYPENAME

#We should use a proxy to stand in for the heavyweight object to take the place of the #object until it is needed. The proxy acts just like the class and handles instantiating it once #its needed.

#this keyword generates a proxy for a class that can be accessed like the class

#proxy classname proxyname

PROXY: PROXYKWD IDENTIFIER IDENTIFIER

#chain of responsbility

#avoie coupling the sender and receiver of a message. Allow for multiple objects to receive

#chain sourceclasss link link link,,,,ddestinationclass

CHAIN: CHAINKWD LINKS IDENTIFIER  
LINKS: LINKKWD IDENTIFIER | COMMA IDENTIFIER

#/\*\*\*\*\* COMMAND DESIGN PATTERN \*\*\*\*/

#command classname { expression as an object }

COMMAND: COMMANDKWD IDENTIFIER LBRACE EXPRESSION RBRACE

COMMANDINVOKER: EXECUTEKWRD IDENTIFIER

#Mediate mediator{ myFirstClass, myFirstInterface, AbstracftClass, #implementAbstractClass }

MEDIATOR: MEDIATORKWD LBRACE MEDLIST RBRACE

MEDLISET: IDENTIFIER | COMMA DENTIFIER MEDLISET

STOREMOMENTO: IDENTIFIER

RETRIEVEMOMENTO: IDENTIFIER

#strategy

#efine a family of algorithms, encapsulate each one, , and make them interchangeable. #Strategy alllows the algorithm to vary indendently from clients that use it.

#strategy myfamily { hello.run(), bye.run() }

STRATEGY: STRATEGYKWD IDENTIFIER LBRACE FAMILYALGORITHMS RBRACE  
FAMILYALGORITHMS: IDENTIFIER | COMMA FAMILYALGORITHMS IDENTIFIER

#singleton

SINGLETON: SINGLETONKWD IDENTIFIER

#tie one variable to anothoer

STATEFUL: STATEFULKWD IDENTIFIER EQUALS TIEKWD

#Equiv one statement assignment

EQUIV: EQUIVKWD IDENTIFIER IDENTIFIER

#MVC

#@model:(ties, ties, ties)

#@view(ties, ties, ties)

#@controller(ties, ties, ties)

MODEL: ATSIGN MODELKWD COLON LBRACE MODELLIST RBRACE

MODELLIST: IDENTIFIER | COMMA MODELLIST IDENTIFIER

VIEW: ATSIGN VIEWKWD COLON LBRACE VIEWLIST RBRACE

VIEWLIST: IDENTIFIER | COMMA VIEWLISET IDENTIFIER  
CONTROLLER: ATSIGN CONTROLLERKWD COLON LBRACE CONTROLLERLIST RBRAWCE  
CONTROLLERLIST: IDENTIFIER | COMMA CONTROLLERLIST IDENTIFIER