

Appendix A Syntax

The Carmel Grammar is based on the grammar to the Java Language, which is defined in the Java Language Specification, Second Edition, §18.1.

The grammar uses the following BNF-style conventions:

- $[x]$ denotes zero or one occurrences of x .
- $\{x\}$ denotes zero or more occurrences of x .
- $x \mid y$ denotes one of either x or y .

For details of the lexical structure of the Carmel language, such as the definition of identifiers and literals, see the Java Language Specification, Second Edition, §3.

A.1 Identifiers, Literals and Types

Identifier:

IDENTIFIER

QualifiedIdentifier:

Identifier [, Identifier]

Literal:

NumericLiteral

BooleanLiteral

NullLiteral

NumericLiteral:

[(NumericType)] [-] IntegerLiteral

VariableInitializer:

Literal

ArrayInitializer

ArrayInitializer:

{ [Literal { , Literal } [,]] }

Type:

Identifier { . Identifier } BracketsOpt

BasicType

BasicType:

NumericType

boolean

NumericType:

byte

short

int

BracketsOpt:

{ [] }

A.2 Packages, Classes and Interfaces

CompilationUnit:

PackageDeclaration

```

PackageDeclaration:
    package QualifiedIdentifier AID ; PackageBody

SpecialCompilationUnit:
    { SpecialPackageDeclaration }

SpecialPackageDeclaration:
    package [ QualifiedIdentifier ] [ AID ] ; PackageBody

AID:
    { NumericLiteral { , NumericLiteral } }

PackageBody:
    { ImportDeclaration } { TypeDeclaration }

ImportDeclaration:
    import Identifier { . Identifier } [ . * ] ;

TypeDeclaration:
    ClassOrInterfaceDeclaration
    ;

ClassOrInterfaceDeclaration:
    ModifiersOpt ( ClassDeclaration | InterfaceDeclaration )

ModifiersOpt:
    { Modifier }

Modifier:
    public
    protected
    private
    static
    abstract
    final
    native

ClassDeclaration:
    class Identifier [ extends Type ] [ implements TypeList ] ClassBody

InterfaceDeclaration:
    interface Identifier [ extends TypeList ] InterfaceBody

TypeList:
    Type { , Type }

ClassBody:
    { { ClassBodyDeclaration } }

InterfaceBody:
    { { InterfaceBodyDeclaration } }

ClassBodyDeclaration:
    ModifiersOpt MemberDecl
    ;

MemberDecl:
    MethodOrFieldDecl
    void Identifier MethodDeclaratorRest
    Identifier ConstructorDeclaratorRest
    void <init> ConstructorDeclaratorRest

MethodOrFieldDecl:
    Type Identifier MethodOrFieldRest

MethodOrFieldRest:
    VariableDeclaratorRest
    MethodDeclaratorRest

VariableDeclaratorRest:
    BracketsOpt [ = VariableInitializer ]

```

```

InterfaceBodyDeclaration:
    ModifiersOpt InterfaceMemberDecl
    ;

InterfaceMemberDecl:
    InterfaceMethodOrFieldDecl
    void Identifier VoidInterfaceMethodDeclaratorRest

InterfaceMethodOrFieldDecl:
    ConstantDeclaratorsRest ;
    InterfaceMethodDeclaratorRest

ConstantDeclaratorsRest:
    ConstantDeclaratorsRest { , ConstantDeclarator }

ConstantDeclarator:
    Identifier ConstantDeclaratorRest

ConstantDeclaratorRest:
    BracketsOpt = VariableInitializer

MethodDeclaratorRest:
    FormalParameters BracketsOpt [ throws QualifiedIdentifierList ] ( MethodBody | ; )

VoidMethodDeclaratorRest:
    FormalParameters [ throws QualifiedIdentifierList ] ( MethodBody | ; )

InterfaceMethodDeclaratorRest:
    FormalParameters BracketsOpt [ throws QualifiedIdentifierList ] ;

VoidInterfaceMethodDeclaratorRest:
    FormalParameters [ throws QualifiedIdentifierList ] ;

ConstructorDeclaratorRest:
    FormalParameters [ throws QualifiedIdentifierList ] MethodBody

QualifiedIdentifierList:
    QualifiedIdentifier { , QualifiedIdentifier }

FormalParameters:
    ( [ FormalParameter { , FormalParameter } ] )

FormalParameter:
    [ final ] Type [ VariableDeclaratorId ]

VariableDeclaratorId:
    Identifier BracketsOpt

MethodBody:
    InstructionBlock

```

A.3 Instructions

```

InstructionBlock:
    { { LabeledInstruction } { ExceptionHandler } }

LabeledInstruction:
    Address : Instruction

ExceptionHandler:
    Address - Address : CatchType => Address

CatchType:
    ClassType
    *

Address:
    DecimalNumeral

```

Instruction:

```

nop
push OperandType Constant
pop NbWords
dup NbWords NbWords
swap NbWords NbWords
numop OperandType NumericOperator [ OperandType ]
load OperandType LocalVariableIndex
store OperandType LocalVariableIndex
inc OperandType LocalVariableIndex IntegerConstant
goto Address
if OperandType ComparisonOperator [ NullComparison ] goto Address
lookupswitch OperandType [ MatchTableEntries ] , default => Address
tableswitch OperandType IndexTableEntry , default => Address
new ReferenceType
checkcast ReferenceType
instanceof ReferenceType
getstatic Name
putstatic Name
getfield [ this ] Name
putfield [ this ] Name
invokedefinite Signature
invokevirtual Signature
invokeinterface Signature
return [ OperandType ]
arraylength
arrayload OperandType
arraystore OperandType
throw
jsr Address
ret LocalVariableIndex

```

OperandType: one of

r b s i

Constant:

NullLiteral
NumericLiteral

NbWords:

DecimalNumeral

ComparisonOperator: one of

eq ne gt le lt ne

NullComparison: one of

0 null

NumericOperator: one of

neg add sub mul div rem and or xor shl shr ushr tob cmp

LocalVariableIndex:

DecimalNumeral

MatchTableEntries:

MatchTableEntries { , MatchTableEntry }

MatchTableEntry:

NumericLiteral => Address

IndexTableEntry:

NumericLiteral => { Address }

Signature:

Name ([TypeList])