

JavaScript is disabled on your browser.

- Prev
- Next
- Frames
- No Frames
- All Classes

A B C D E F G H I J K L M N O P R S T U V W Z

## A

**A0** - Static variable in class jminusminus.NPhysicalRegister  
Argument 1.

**A1** - Static variable in class jminusminus.NPhysicalRegister  
Argument 2.

**A2** - Static variable in class jminusminus.NPhysicalRegister  
Argument 3.

**A3** - Static variable in class jminusminus.NPhysicalRegister  
Argument 4.

**AALOAD** - Static variable in class jminusminus.CLConstants  
AALOAD instruction.

**AASTORE** - Static variable in class jminusminus.CLConstants  
AASTORE instruction.

**ABSTRACT** - Static variable in interface jminusminus.JavaCCParserConstants

**abstractMethods()** - Method in class jminusminus.Type

Return a list of this class' abstract methods? It does has abstract methods if (1) Any method declared in the class is abstract, or (2) Its superclass has an abstract method which is not overridden here.

**ACC\_ABSTRACT** - Static variable in class jminusminus.CLConstants  
abstract access flag.

**ACC\_ANNOTATION** - Static variable in class jminusminus.CLConstants  
annotation access flag.

**ACC\_BRIDGE** - Static variable in class jminusminus.CLConstants  
bridge access flag.

**ACC\_ENUM** - Static variable in class jminusminus.CLConstants  
enum access flag.

**ACC\_FINAL** - Static variable in class jminusminus.CLConstants  
final access flag.

**ACC\_INTERFACE** - Static variable in class jminusminus.CLConstants  
interface access flag.

**ACC\_NATIVE** - Static variable in class jminusminus.CLConstants  
native access flag.

**ACC\_PRIVATE** - Static variable in class jminusminus.CLConstants  
private access flag.

**ACC\_PROTECTED** - Static variable in class jminusminus.CLConstants  
protected access flag.

**ACC\_PUBLIC** - Static variable in class jminusminus.CLConstants  
public access flag.

**ACC\_STATIC** - Static variable in class jminusminus.CLConstants

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

static access flag.

**ACC\_STRICT** - Static variable in class jminusminus.CLConstants

strict access flag.

**ACC\_SUPER** - Static variable in class jminusminus.CLConstants

super access flag.

**ACC\_SYNCHRONIZED** - Static variable in class jminusminus.CLConstants

synchronized access flag.

**ACC\_SYNTHETIC** - Static variable in class jminusminus.CLConstants

synthetic access flag.

**ACC\_TRANSIENT** - Static variable in class jminusminus.CLConstants

transient access flag.

**ACC\_VARARGS** - Static variable in class jminusminus.CLConstants

varargs access flag.

**ACC\_VOLATILE** - Static variable in class jminusminus.CLConstants

volatile access flag.

**accessFlags** - Variable in class jminusminus.CLFile

ClassFile.access\_flags item.

**accessFlags** - Variable in class jminusminus.CLMemberInfo

member\_info.access\_flags item.

**accessFlagToInt(String)** - Static method in class jminusminus.CLFile

Return the integer value (mask) corresponding to the specified access flag.

**ACONST\_NULL** - Static variable in class jminusminus.CLConstants

ACONST\_NULL instruction.

**active** - Variable in class jminusminus.NBasicBlock

Is this block active?

**add\_escapes(String)** - Method in exception jminusminus.ParseException

Used to convert raw characters to their escaped version when these raw version cannot be used as part of an ASCII string literal.

**addArrayInstruction(int, String)** - Method in class jminusminus.CLEmitter

Add an array instruction.

**addBranchInstruction(int, String)** - Method in class jminusminus.CLEmitter

Add a branch instruction.

**addClass(ArrayList<String>, String, String, ArrayList<String>, boolean)** - Method in class jminusminus.CLEmitter

Add a class or interface.

**addClassAttribute(CLAttributeInfo)** - Method in class jminusminus.CLEmitter

Add the specified class attribute to the attribyte section of the class.

**addCodeAttribute(CLAttributeInfo)** - Method in class jminusminus.CLEmitter

Add the specified code attribute to the attribute section of the code for the method last added.

**addCPIItem(CLCPInfo)** - Method in class jminusminus.CLConstantPool

Add the specified (non null) item to the constant pool table and return its index.

**addEntry(int, String, IDefn)** - Method in class jminusminus.Context

Add an entry to the symbol table, binding a name to its definition in the current context.

**addEscapes(String)** - Static method in error jminusminus.TokenMgrError

Replaces unprintable characters by their escaped (or unicode escaped) equivalents in the given string

**addExceptionHandler(String, String, String, String)** - Method in class jminusminus.CLEmitter

Add an exception handler.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**addField(ArrayList<String>, String, String, boolean)** - Method in class jminusminus.CLEmitter

Add a field without initialization.

**addField(ArrayList<String>, String, String, boolean, int)** - Method in class jminusminus.CLEmitter

Add an int, short, char, byte, or boolean field with initialization.

**addField(ArrayList<String>, String, boolean, float)** - Method in class jminusminus.CLEmitter

Add a float field with initialization.

**addField(ArrayList<String>, String, boolean, long)** - Method in class jminusminus.CLEmitter

Add a long field with initialization.

**addField(ArrayList<String>, String, boolean, double)** - Method in class jminusminus.CLEmitter

Add a double field with initialization.

**addField(ArrayList<String>, String, boolean, String)** - Method in class jminusminus.CLEmitter

Add a String type field with initialization.

**addFieldAttribute(CLEmitterInfo)** - Method in class jminusminus.CLEmitter

Add the specified field attribute the attribute section of the field last added.

**addIINCInstruction(int, int)** - Method in class jminusminus.CLEmitter

Add an IINC instruction to increment a variable by a constant.

**addInnerClass(ArrayList<String>, String, String, String)** - Method in class jminusminus.CLEmitter

Add an inner class.

**addLabel(String)** - Method in class jminusminus.CLEmitter

Add a jump label to the code section of the method being added.

**addLDCInstruction(int)** - Method in class jminusminus.CLEmitter

Add an LDC instruction to load an int constant on the operand stack.

**addLDCInstruction(float)** - Method in class jminusminus.CLEmitter

Add an LDC instruction to load a float constant on the operand stack.

**addLDCInstruction(long)** - Method in class jminusminus.CLEmitter

Add an LDC instruction to load a long constant on the operand stack.

**addLDCInstruction(double)** - Method in class jminusminus.CLEmitter

Add an LDC instruction to load a double constant on the operand stack.

**addLDCInstruction(String)** - Method in class jminusminus.CLEmitter

Add an LDC instruction to load a String constant on the operand stack.

**addLOOKUPSWITCHInstruction(String, int, TreeMap<Integer, String>)** - Method in class jminusminus.CLEmitter

Add a LOOKUPSWITCH instruction -- used for switch statements.

**addMemberAccessInstruction(int, String, String, String)** - Method in class jminusminus.CLEmitter

Add a member (field & method) access instruction.

**addMethod(ArrayList<String>, String, String, ArrayList<String>, boolean)** - Method in class jminusminus.CLEmitter

Add a method.

**addMethodAttribute(CLEmitterInfo)** - Method in class jminusminus.CLEmitter

Add the specified method attribute to the attribute section of the method last added.

**addMULTIANEWARRAYInstruction(String, int)** - Method in class jminusminus.CLEmitter

Add a MULTIANEWARRAY instruction for creating multi-dimensional arrays.

**addNoArgInstruction(int)** - Method in class jminusminus.CLEmitter

Add a no argument instruction.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**addOneArgInstruction(int, int)** - Method in class jminusminus.CLEmitter

Add a one argument instruction.

**addOrExtendNRange(NRange)** - Method in class jminusminus.NInterval

Add a new range to the existing ranges.

**addReferenceInstruction(int, String)** - Method in class jminusminus.CLEmitter

Add a reference (object) instruction.

**addTABLESWITCHInstruction(String, int, int, ArrayList<String>)** - Method in class jminusminus.CLEmitter

Add a TABLESWITCH instruction -- used for switch statements.

**addType(int, Type)** - Method in class jminusminus.Context

Add the type to the environment.

**addUsePosition(Integer, InstructionType)** - Method in class jminusminus.NInterval

Register a use (read or write)>

**adjustBeginLineColumn(int, int)** - Method in class jminusminus.SimpleCharStream

Method to adjust line and column numbers for the start of a token.

**allocatePhysicalRegisters()** - Method in class jminusminus.NControlFlowGraph

Replace references to virtual registers in LIR instructions with references to physical registers.

**allocatePhysicalRegisters()** - Method in class jminusminus.NLIRArithmetic

**allocatePhysicalRegisters()** - Method in class jminusminus.NLIRConditionalJump

**allocatePhysicalRegisters()** - Method in class jminusminus.NLIRInstruction

Replace references to virtual registers in this LIR instruction with references to physical registers.

**allocatePhysicalRegisters()** - Method in class jminusminus.NLIRIntConstant

**allocatePhysicalRegisters()** - Method in class jminusminus.NLIRInvoke

**allocatePhysicalRegisters()** - Method in class jminusminus.NLIRMove

**allocatePhysicalRegisters()** - Method in class jminusminus.NLIRStore

**allocatePhysicalRegisters()** - Method in class jminusminus.NLIRStringConstant

**allocation()** - Method in class jminusminus.NGraphRegisterAllocator

Build intervals with register allocation information in them.

**allocation()** - Method in class jminusminus.NLinearRegisterAllocator

Perform the linear register allocation, assigning physical registers to virtual registers.

**allocation()** - Method in class jminusminus.NNaiveRegisterAllocator

Build intervals with (naive) register allocation information in them.

**allocation()** - Method in class jminusminus.NRegisterAllocator

The work horse that does the allocation, implemented in the concrete sub-classes of NRegisterAllocator.

**ALOAD** - Static variable in class jminusminus.CLConstants

ALOAD instruction.

**ALOAD\_0** - Static variable in class jminusminus.CLConstants

ALOAD\_0 instruction.

**ALOAD\_1** - Static variable in class jminusminus.CLConstants

ALOAD\_1 instruction.

**ALOAD\_2** - Static variable in class jminusminus.CLConstants

ALOAD\_2 instruction.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**ALOAD\_3** - Static variable in class `jminusminus.CLConstants`

`ALOAD_3` instruction.

**AmbiguousName** - Class in `jminusminus`

Ambiguous names are meant to deal with snippets like

**AmbiguousName(int, String)** - Constructor for class `jminusminus.AmbiguousName`

Construct an encapsulation of the ambiguous portion of a snippet like `x.y.z`.

**analyze(Context)** - Method in class `jminusminus.JArrayExpression`

Perform semantic analysis on an array indexing expression such as `A[i]`.

**analyze(Context)** - Method in class `jminusminus.JArrayInitializer`

Analysis of array initializer involves making sure that the type of the initials is the same as the component type.

**analyze(Context)** - Method in class `jminusminus.JAssignOp`

Analyze the lhs and rhs, checking that types match, and set the result type.

**analyze(Context)** - Method in class `jminusminus.JAST`

Perform semantic analysis on this AST.

**analyze(Context)** - Method in class `jminusminus.JBlock`

Analyzing a block consists of creating a new nested context for that block and analyzing each of its statements within that context.

**analyze(Context)** - Method in class `jminusminus.JCastOp`

Analyzing a cast expression means, resolving the type (to which we are casting), checking the legality of the cast, and computing a (possibly null) conversion for use in code generation.

**analyze(Context)** - Method in class `jminusminus.JClassDeclaration`

Perform semantic analysis on the class and all of its members within the given context.

**analyze(Context)** - Method in class `jminusminus.JComparison`

The analysis of a comparison operation consists of analyzing its two operands, and making sure they both have the same numeric type.

**analyze(Context)** - Method in class `jminusminus.JCompilationUnit`

Perform semantic analysis on the AST in the specified context.

**analyze(Context)** - Method in class `jminusminus.JConstructorDeclaration`

Analysis for a constructor declaration is very much like that for a method declaration.

**analyze(Context)** - Method in class `jminusminus.JEmptyStatement`

**analyze(Context)** - Method in class `jminusminus.JEqualOp`

Analyzing an equality expression means analyzing its operands and checking that the types match.

**analyze(Context)** - Method in class `jminusminus.JExpression`

The analysis of any `JExpression` returns a `JExpression`.

**analyze(Context)** - Method in class `jminusminus.JFieldDeclaration`

Analysis of field declaration involves rewriting initializations (if any) as assignment statements.

**analyze(Context)** - Method in class `jminusminus.JFieldSelection`

Analyzing a field selection expression involves, (1) reclassifying any ambiguous part, (2) analyzing the target, (3) treating "length" field of arrays specially, or computing the Field object, (4) checking the access rules, and (5) computing the resultant type.

**analyze(Context)** - Method in class `jminusminus.JFormalParameter`

No analysis done here.

**analyze(Context)** - Method in class `jminusminus.JIfStatement`

Analyzing the if-statement means analyzing its components and checking that the test is boolean.

**analyze(Context)** - Method in class `jminusminus.JInstanceOfOp`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Analysis of an instance of operation requires analyzing the expression to be tested, resolving the type we are testing for, and determining if the test is legal, or if the answer can be determined at compile time.

**analyze(Context)** - Method in class `jminusminus.JLiteralChar`

Analyzing a char literal is trivial.

**analyze(Context)** - Method in class `jminusminus.JLiteralFalse`

Analyzing a boolean literal is trivial.

**analyze(Context)** - Method in class `jminusminus.JLiteralInt`

Analyzing an int literal is trivial.

**analyze(Context)** - Method in class `jminusminus.JLiteralNull`

Analyzing the null literal is trivial.

**analyze(Context)** - Method in class `jminusminus.JLiteralString`

Analyzing a String literal is trivial.

**analyze(Context)** - Method in class `jminusminus.JLiteralTrue`

Analyzing a boolean literal is trivial.

**analyze(Context)** - Method in class `jminusminus.JLogicalAndOp`

Analyzing a logical AND expression involves analyzing its operands and insuring they are boolean; the result type is of course boolean.

**analyze(Context)** - Method in class `jminusminus.JLogicalNotOp`

Analyzing a logical NOT operation means analyzing its operand, insuring it's a boolean, and setting the result to boolean.

**analyze(Context)** - Method in class `jminusminus.JMessageExpression`

Analysis of a message expression involves: (1) reclassifying any ambiguous part, (2) analyzing and computing the types for the actual arguments, (3) determining the type we are currently in (for checking access), (4) analyzing the target and determining its type, (5) finding the appropriate Method, (6) checking accessibility, and (7) determining the result type.

**analyze(Context)** - Method in class `jminusminus.JMethodDeclaration`

Analysis for a method declaration involves (1) creating a new method context (that records the return type; this is used in the analysis of the method body), (2) bumping up the offset (for instance methods), (3) declaring the formal parameters in the method context, and (4) analyzing the method's body.

**analyze(Context)** - Method in class `jminusminus.JMultiplyOp`

Analyzing the \* operation involves analyzing its operands, checking types, and determining the result type.

**analyze(Context)** - Method in class `jminusminus.JNegateOp`

Analyzing the negation operation involves analyzing its operand, checking its type and determining the result type.

**analyze(Context)** - Method in class `jminusminus.JNewArrayOp`

Analysis of a new array operation involves resolving its type and analyzing the array bounds and checking their types.

**analyze(Context)** - Method in class `jminusminus.JNewOp`

To analyze the new operation, we (1) resolve the type, (2) analyze its arguments, (3) check accessibility of the type, (3) find the appropriate Constructor.

**analyze(Context)** - Method in class `jminusminus.JPlusAssignOp`

Analyze the lhs and rhs, rewrite rhs as lhs + rhs (string concatenation) if lhs is a String, and set the result type.

**analyze(Context)** - Method in class `jminusminus.JPlusOp`

Analysis involves first analyzing the operands.

**analyze(Context)** - Method in class `jminusminus.JPostDecrementOp`

Analyze the operand as a lhs (since there is a side effect), check types and determine the type of the result.

**analyze(Context)** - Method in class jminusminus.JPreIncrementOp

Analyze the operand as a lhs (since there is a side effect), check types and determine the type of the result.

**analyze(Context)** - Method in class jminusminus.JReturnStatement

Analysis distinguishes between our being in a constructor or in a regular method in checking return types.

**analyze(Context)** - Method in class jminusminus.JStatementExpression

Analysis involves analyzing the encapsulated expression if indeed it is a statement expression, i.e., one with a side effect.

**analyze(Context)** - Method in class jminusminus.JStringConcatenationOp

Analysis is simple here.

**analyze(Context)** - Method in class jminusminus.JSubtractOp

Analyzing the - operation involves analyzing its operands, checking types, and determining the result type.

**analyze(Context)** - Method in class jminusminus.JSuper

Analysis involves determining the super class to that in which we are in; this becomes the type.

**analyze(Context)** - Method in class jminusminus.JSuperConstruction

Analyzing a super constructor statement involves (1) setting the type, (2) analyzing the actual arguments, and (3) checking that this construction statement is properly invoked (as the first statement in another constructor).

**analyze(Context)** - Method in class jminusminus.JThis

Analysis involves simply determining the type in which we are, since that determines the type of this target.

**analyze(Context)** - Method in class jminusminus.JThisConstruction

Analyzing a this constructor statement involves (1) setting the type, (2) analyzing the actual arguments, (3) checking that this construction statement is properly invoked (as the first statement in another constructor), and (4) finding the appropriate Constructor

**analyze(Context)** - Method in class jminusminus.JVariable

Analyzing identifiers involves resolving them in the context.

**analyze(Context)** - Method in class jminusminus.JVariableDeclaration

We declare the variable(s).

**analyze(Context)** - Method in class jminusminus.JVariableDeclarator

No analysis is done here.

**analyze(Context)** - Method in class jminusminus.JWhileStatement

Analysis involves analyzing the test, checking its type and analyzing the body statement.

**analyze(Context)** - Method in class jminusminus.JWildExpression

Simply set the type to ANY (a wild type matching everything).

**analyzeLhs(Context)** - Method in class jminusminus.JArrayExpression

Analyzing the array expression as an Lvalue is like analyzing it for its Rvalue.

**analyzeLhs(Context)** - Method in class jminusminus.JFieldSelection

Analyze the field selection expression for use on the lhs of an assignment.

**analyzeLhs(Context)** - Method in interface jminusminus.JLhs

Analyze the lhs of an assignment.

**analyzeLhs(Context)** - Method in class jminusminus.JVariable

Analyze the identifier as used on the lhs of an assignment.

**ANEWARRAY** - Static variable in class jminusminus.CLConstants

ANEWARRAY instruction.

**annotations** - Variable in class jminusminus.CLParameterAnnotationInfo

parameter\_annotations\_table\_entry.annotations item.

**annotations** - Variable in class jminusminus.CLRuntimeInvisibleAnnotationsAttribute

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

RuntimeInvisibleAnnotations\_attribute.annotations item.

**annotations** - Variable in class jminusminus.CLRuntimeVisibleAnnotationsAttribute  
RuntimeVisibleAnnotations\_attribute.annotations item.

**annotationValue** - Variable in class jminusminus.CLElementValue  
element\_value.annotation\_value item.

**ANY** - Static variable in class jminusminus.Type  
The "any" type (denotes wild expressions).

**ARETURN** - Static variable in class jminusminus.CLConstants  
ARETURN instruction.

**arg** - Variable in class jminusminus.JavaCCParser.JJCalls

**arg** - Variable in class jminusminus.JUnaryExpression  
The operand.

**argTypesAsString(Type[])** - Static method in class jminusminus.Type  
Convert an array of argument types to a string representation of a parenthesized list of the types, eg, (int, boolean, java.lang.String).

**argTypesMatch(Class<?>[], Class<?>[])** - Static method in class jminusminus.Type  
Do argument types match? A helper used for finding candidate methods and constructors.

**arguments** - Variable in class jminusminus.NHIRInvoke  
List of HIR ids of arguments for the method.

**arguments** - Variable in class jminusminus.NHIRPhiFunction  
List of HIR ids of arguments for the phi function.

**argumentTypeForAppend()** - Method in class jminusminus.Type  
The String representation for a type being appended to a StringBuffer for + and += over strings.

**ARRAYLENGTH** - Static variable in class jminusminus.CLConstants  
ARRAYLENGTH instruction.

**arrayRef** - Variable in class jminusminus.NHIRALoad  
HIR id of the array reference.

**arrayRef** - Variable in class jminusminus.NHIRASTore  
HIR id of the array reference.

**ArrayTypeNames** - Class in jminusminus  
The (temporary) representation of an array's type.

**ArrayTypeNames(Type)** - Constructor for class jminusminus.ArrayTypeNames  
Construct an array's type given its component type.

**ASSIGN** - Static variable in interface jminusminus.JavaCCParserConstants

**ASTORE** - Static variable in class jminusminus.CLConstants  
ASTORE instruction.

**ASTORE\_0** - Static variable in class jminusminus.CLConstants  
ASTORE\_0 instruction.

**ASTORE\_1** - Static variable in class jminusminus.CLConstants  
ASTORE\_1 instruction.

**ASTORE\_2** - Static variable in class jminusminus.CLConstants  
ASTORE\_2 instruction.

**ASTORE\_3** - Static variable in class jminusminus.CLConstants  
ASTORE\_3 instruction.

**AT** - Static variable in class jminusminus.NPhysicalRegister  
Reserved for assembler.

**ATHROW** - Static variable in class jminusminus.CLConstants

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



ATHROW instruction.

**ATT\_ANNOTATION\_DEFAULT** - Static variable in class jminusminus.CLConstants  
Identifies AnnotationDefault attribute.

**ATT\_CODE** - Static variable in class jminusminus.CLConstants  
Identifies Code attribute.

**ATT\_CONSTANT\_VALUE** - Static variable in class jminusminus.CLConstants  
Identifies ConstantValue attribute.

**ATT\_DEPRECATED** - Static variable in class jminusminus.CLConstants  
Identifies Deprecated attribute.

**ATT\_ENCLOSING\_METHOD** - Static variable in class jminusminus.CLConstants  
Identifies EnclosingMethod attribute.

**ATT\_EXCEPTIONS** - Static variable in class jminusminus.CLConstants  
Identifies Exceptions attribute.

**ATT\_INNER\_CLASSES** - Static variable in class jminusminus.CLConstants  
Identifies InnerClasses attribute.

**ATT\_LINE\_NUMBER\_TABLE** - Static variable in class jminusminus.CLConstants  
Identifies LineNumberTable attribute.

**ATT\_LOCAL\_VARIABLE\_TABLE** - Static variable in class jminusminus.CLConstants  
Identifies LocalVariableTable attribute.

**ATT\_LOCAL\_VARIABLE\_TYPE\_TABLE** - Static variable in class jminusminus.CLConstants  
Identifies LocalVariableTypeTable attribute.

**ATT\_RUNTIME\_INVISIBLE\_ANNOTATIONS** - Static variable in class jminusminus.CLConstants  
Identifies RuntimeInvisibleAnnotations attribute.

**ATT\_RUNTIME\_INVISIBLE\_PARAMETER\_ANNOTATIONS** - Static variable in class jminusminus.CLConstants  
Identifies RuntimeInvisibleParameterAnnotations attribute.

**ATT\_RUNTIME\_VISIBLE\_ANNOTATIONS** - Static variable in class jminusminus.CLConstants  
Identifies RuntimeVisibleAnnotations attribute.

**ATT\_RUNTIME\_VISIBLE\_PARAMETER\_ANNOTATIONS** - Static variable in class jminusminus.CLConstants  
Identifies RuntimeVisibleParameterAnnotations attribute.

**ATT\_SIGNATURE** - Static variable in class jminusminus.CLConstants  
Identifies Signature attribute.

**ATT\_SOURCE\_DEBUG\_EXTENSION** - Static variable in class jminusminus.CLConstants  
Identifies SourceDebugExtension attribute.

**ATT\_SOURCE\_FILE** - Static variable in class jminusminus.CLConstants  
Identifies SourceFile attribute.

**ATT\_SYNTHETIC** - Static variable in class jminusminus.CLConstants  
Identifies Synthetic attribute.

**attributeLength** - Variable in class jminusminus.CLAtributeInfo  
attribute\_info.attribute\_length item.

**attributeNameIndex** - Variable in class jminusminus.CLAtributeInfo  
attribute\_info.attribute\_name\_index item.

**attributes** - Variable in class jminusminus.CLCodeAttribute  
Code\_attribute.attributes item.

**attributes** - Variable in class jminusminus.CLFile  
ClassFile.attributes item.

**attributes** - Variable in class jminusminus.CLMemberInfo  
member\_info.attributes item.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**attributesCount** - Variable in class jminusminus.CLCODEAttribute  
Code\_attribute.attributes\_count item.

**attributesCount** - Variable in class jminusminus.CLFile  
ClassFile.attributes\_count item.

**attributesCount** - Variable in class jminusminus.CLMemberInfo  
member\_info.attributes\_count item.

**available** - Variable in class jminusminus.SimpleCharStream

## B

**b** - Variable in class jminusminus.CLConstantUtf8Info  
CONSTANT\_Utf8\_info.bytes item.

**backup(int)** - Method in class jminusminus.SimpleCharStream

**BALOAD** - Static variable in class jminusminus.CLConstants  
BALOAD instruction.

**basicBlocks** - Variable in class jminusminus.NControlFlowGraph  
List of blocks forming the cfg for the method.

**BASTORE** - Static variable in class jminusminus.CLConstants  
BASTORE instruction.

**beginColumn** - Variable in class jminusminus.Token  
beginLine and beginColumn describe the position of the first character of this token;  
endLine and endColumn describe the position of the last character of this token.

**beginLine** - Variable in class jminusminus.Token  
beginLine and beginColumn describe the position of the first character of this token;  
endLine and endColumn describe the position of the last character of this token.

**BeginToken()** - Method in class jminusminus.SimpleCharStream

**BIPUSH** - Static variable in class jminusminus.CLConstants  
BIPUSH instruction.

**block** - Variable in class jminusminus.NHIRInstruction  
The block containing this instruction.

**block** - Variable in class jminusminus.NLIRInstruction  
The block containing this instruction.

**blockAt(int)** - Method in class jminusminus.NControlFlowGraph  
The basic block at a particular instruction id.

**blockId** - Static variable in class jminusminus.NControlFlowGraph  
block identifier.

**body** - Variable in class jminusminus.JMethodDeclaration  
Method body.

**BOOLEAN** - Static variable in interface jminusminus.JavaCCParserConstants

**BOOLEAN** - Static variable in class jminusminus.Type  
The primitive type, boolean.

**BOXED\_BOOLEAN** - Static variable in class jminusminus.Type  
java.lang.Boolean.

**BOXED\_CHAR** - Static variable in class jminusminus.Type  
java.lang.Character.

**BOXED\_INT** - Static variable in class jminusminus.Type  
java.lang.Integer.

**Boxing** - Class in `jminusminus`

Boxing requires invoking the appropriate conversion method from the (Java) API.

**Boxing(Type, Type)** - Constructor for class `jminusminus.Boxing`

Construct a Boxing converter.

**bufcolumn** - Variable in class `jminusminus.SimpleCharStream`

**buffer** - Variable in class `jminusminus.SimpleCharStream`

**bufline** - Variable in class `jminusminus.SimpleCharStream`

**bufpos** - Variable in class `jminusminus.SimpleCharStream`

**bufsize** - Variable in class `jminusminus.SimpleCharStream`

**buildIntervals()** - Method in class `jminusminus.NRegisterAllocator`

Build the intervals for a control flow graph.

**bwdBranches** - Variable in class `jminusminus.NBasicBlock`

Number of backward branches to this block.

**byteAt(int, int)** - Method in class `jminusminus.CLInstruction`

Return the byte from i at position byteNum.

**ByteClassLoader** - Class in `jminusminus`

A class loader to be able to load a class from a byte stream.

**ByteClassLoader()** - Constructor for class `jminusminus.ByteClassLoader`

**bytes()** - Method in class `jminusminus.CLConstantFloatInfo`

Return CONSTANT\_Float\_info.bytes item.

**bytes()** - Method in class `jminusminus.CLConstantIntegerInfo`

Return CONSTANT\_Integer\_info.bytes item.

## C

Add WeChat powcoder

**CALOAD** - Static variable in class `jminusminus.CLConstants`

CALOAD instruction.

**CASTORE** - Static variable in class `jminusminus.CLConstants`

CASTORE instruction.

**catchType** - Variable in class `jminusminus.CLEnvironment`

The class of exceptions that this exception handler is designated to catch.

**catchType** - Variable in class `jminusminus.CLEnvironmentInfo`

exception\_table\_entry.catch\_type item.

**category** - Variable in class `jminusminus.CLInsInfo`

The category under which instruction belongs.

**cfg** - Variable in class `jminusminus.NBasicBlock`

The control flow graph (cfg) that this block belongs to.

**cfg** - Variable in class `jminusminus.NRegisterAllocator`

The control flow graph for a method.

**CHAR** - Static variable in interface `jminusminus.JavaCCParserConstants`

**CHAR** - Static variable in class `jminusminus.Type`

The primitive type, char.

**CHAR\_LITERAL** - Static variable in interface `jminusminus.JavaCCParserConstants`

**CharReader** - Class in `jminusminus`

A buffered character reader.

**CharReader(String)** - Constructor for class `jminusminus.CharReader`

Construct a CharReader from a file name.

**checkAccess(int, Member)** - Method in class `jminusminus.Type`

Check the accessibility of a member from this type (that is, this type is the referencing type).

**checkAccess(int, Type)** - Method in class `jminusminus.Type`

Check the accessibility of a target type (from this type)

**checkAccess(int, Class, Class)** - Static method in class `jminusminus.Type`

Check the accessibility of a type.

**CHECKCAST** - Static variable in class `jminusminus.CLConstants`

CHECKCAST instruction.

**childAt(int)** - Method in class `jminusminus.NInterval`

The child interval at a given instruction index.

**childAtOrEndingBefore(NBasicBlock)** - Method in class `jminusminus.NInterval`

A child of this interval which is live or ends before the given basic block's end.

**childAtOrStartingAfter(NBasicBlock)** - Method in class `jminusminus.NInterval`

The child of this interval which is live or starts after the given basic block's start

**children** - Variable in class `jminusminus.NInterval`

Children of this interval.

**CLAbsorber** - Class in `jminusminus`

CLAbsorbers for reading a Java class file an in-memory CLFile representation and printing it out to STDOUT in a format similar to that of javap.

**CLAbsorber(String)** - Constructor for class `jminusminus.CLAbsorber`

Construct a CLAbsorber object given the (fully-qualified) name of the class file to read.

**CLAnnotation** - Class in `jminusminus`

Representation of annotation structure (JVM Spec Section 4.8.15).

**CLAnnotation(int, int, ArrayList<CLElementValuePair>)** - Constructor for class `jminusminus.CLAnnotation`

Construct a CLAnnotation object.

**CLAnnotationDefaultAttribute** - Class in `jminusminus`

Representation of AnnotationDefault\_attribute structure (JVM Spec Section 4.8.2).

**CLAnnotationDefaultAttribute(int, long, CLElementValue)** - Constructor for class `jminusminus.CLAnnotationDefaultAttribute`

Construct a CLAnnotationDefaultAttribute object.

**CLArithmeticInstruction** - Class in `jminusminus`

Representation for ARITHMETIC1 and ARITHMETIC2 instructions.

**CLArithmeticInstruction(int, int)** - Constructor for class `jminusminus.CLArithmeticInstruction`

Construct a CLArithmeticInstruction object for ARITHMETIC1 instructions.

**CLArithmeticInstruction(int, int, int, int, boolean)** - Constructor for class `jminusminus.CLArithmeticInstruction`

Construct a CLArithmeticInstruction object for IINC instruction.

**CLArrayInstruction** - Class in `jminusminus`

Representation for ARRAY1, ARRAY2 and ARRAY3 instructions.

**CLArrayInstruction(int, int, int)** - Constructor for class `jminusminus.CLArrayInstruction`

Construct a CLArrayInstruction object for ARRAY1 instructions.

**CLArrayInstruction(int, int, int, int)** - Constructor for class `jminusminus.CLArrayInstruction`

Construct a CLArrayInstruction object for ARRAY2 instructions.

**CLArrayInstruction(int, int)** - Constructor for class `jminusminus.CLArrayInstruction`

Construct a CLArrayInstruction object for ARRAY3 instructions.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**CLASS** - Static variable in interface `jminusminus.JavaCCParserConstants`

**classAccessFlagsToString(int)** - Static method in class `jminusminus.CLFile`

Return (as a string) the class access permissions and properties contained in the specified mask of flags.

**ClassContext** - Class in `jminusminus`

Represents the context (scope, environment, symbol table) for a type, eg a class, in j--.

**ClassContext(JAST, Context)** - Constructor for class `jminusminus.ClassContext`

Construct a class context.

**classContext** - Variable in class `jminusminus.Context`

The surrounding class context.

**classContext()** - Method in class `jminusminus.Context`

Return the surrounding class context.

**classes** - Variable in class `jminusminus.CLInnerClassesAttribute`

InnerClasses\_attribute.classes item.

**classFile()** - Method in class `jminusminus.CLAbsorber`

Return the CLFile representation of the class that was read.

**classIndex** - Variable in class `jminusminus.CLConstantMemberRefInfo`

CONSTANT\_Memberref\_info.class\_index item.

**classIndex** - Variable in class `jminusminus.CLEnclosingMethodAttribute`

EnclosingMethod\_attribute.class\_index item.

**classInfoIndex** - Variable in class `jminusminus.CLElementValue`

element\_value.class\_info\_index item.

**classRep()** - Method in class `jminusminus.Type`

Return the class representation for a type, appropriate for dealing with the Java reflection API.

**CLAttributeInfo** - Class in `jminusminus`

Representation of attribute\_info structure (JVM Spec Section 4.8).

**CLAttributeInfo(int, long)** - Constructor for class `jminusminus.CLAttributeInfo`

Construct an CLAttributeInfo object.

**CLBitInstruction** - Class in `jminusminus`

Representation for BIT instructions.

**CLBitInstruction(int, int)** - Constructor for class `jminusminus.CLBitInstruction`

Construct a CLBitInstruction object.

**CLBranchStack** - Class in `jminusminus`

This class is used for control flow analysis to compute maximum depth of operand stack for a method.

**CLBranchStack()** - Constructor for class `jminusminus.CLBranchStack`

Construct a CLBranchStack object.

**CLBranchTarget** - Class in `jminusminus`

Instances of this class form the elements of the CLBranchStack which is used for control flow analysis to compute maximum depth of operand stack for a method.

**CLBranchTarget(CLInstruction, int)** - Constructor for class `jminusminus.CLBranchTarget`

Construct a CLBranchTarget object.

**CLCodeAttribute** - Class in `jminusminus`

Representation of Code\_attribute structure (JVM Spec Section 4.8.2).

**CLCodeAttribute(int, long, int, int, long, ArrayList<Integer>, int, ArrayList<CLExceptionInfo>, int, ArrayList<CLAttributeInfo>)** - Constructor for class `jminusminus.CLCodeAttribute`

Construct a CLCodeAttribute object.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



**CLComparisonInstruction** - Class in `jminusminus`

Representation for COMPARISON instructions.

**CLComparisonInstruction(int, int)** - Constructor for class `jminusminus.CLComparisonInstruction`

Construct a `CLComparisonInstruction` object.

**CLConstantClassInfo** - Class in `jminusminus`

Representation of `CONSTANT_Class_info` structure (JVM Spec Section 4.5.1).

**CLConstantClassInfo(int)** - Constructor for class `jminusminus.CLConstantClassInfo`

Construct a `CLConstantClassInfo` object.

**CLConstantDoubleInfo** - Class in `jminusminus`

Representation of `CONSTANT_Double_info` structure (JVM Spec Section 4.5.5).

**CLConstantDoubleInfo(double)** - Constructor for class `jminusminus.CLConstantDoubleInfo`

Construct a `CLConstantDoubleInfo` object.

**CLConstantFieldRefInfo** - Class in `jminusminus`

Representation of `CONSTANT_Fieldref_info` structure (JVM Spec Section 4.5.2).

**CLConstantFieldRefInfo(int, int)** - Constructor for class `jminusminus.CLConstantFieldRefInfo`

Construct a `CLConstantFieldRefInfo` object.

**CLConstantFloatInfo** - Class in `jminusminus`

Representation of `CONSTANT_Float_info` structure (JVM Spec Section 4.5.4).

**CLConstantFloatInfo(float)** - Constructor for class `jminusminus.CLConstantFloatInfo`

Construct a `CLConstantFloatInfo` object.

**CLConstantIntegerInfo** - Class in `jminusminus`

Representation of `CONSTANT_Integer_info` structure (JVM Spec Section 4.5.4).

**CLConstantIntegerInfo(int)** - Constructor for class `jminusminus.CLConstantIntegerInfo`

Construct a `CLConstantIntegerInfo` object.

**CLConstantInterfaceMethodRefInfo** - Class in `jminusminus`

Representation of `CONSTANT_InterfaceMethodref_info` structure (JVM Spec Section 4.5.2).

**CLConstantInterfaceMethodRefInfo(int, int)** - Constructor for class `jminusminus.CLConstantInterfaceMethodRefInfo`

Construct a `CLConstantInterfaceMethodRefInfo` object.

**CLConstantLongInfo** - Class in `jminusminus`

Representation of `CONSTANT_Long_info` structure (JVM Spec Section 4.5.5).

**CLConstantLongInfo(long)** - Constructor for class `jminusminus.CLConstantLongInfo`

Construct a `CLConstantLongInfo` object.

**CLConstantMemberRefInfo** - Class in `jminusminus`

Abstract super class of `CONSTANT_Fieldref_info`, `CONSTANT_Methodref_info`, `CONSTANT_InterfaceMethodref_info` structures (JVM Spec Section 4.5.2).

**CLConstantMemberRefInfo(int, int, short)** - Constructor for class `jminusminus.CLConstantMemberRefInfo`

Construct a `CLConstantMemberRefInfo` object.

**CLConstantMethodRefInfo** - Class in `jminusminus`

Representation of `CONSTANT_Methodref_info` structure (JVM Spec Section 4.5.2).

**CLConstantMethodRefInfo(int, int)** - Constructor for class `jminusminus.CLConstantMethodRefInfo`

Construct a `CLConstantMethodRefInfo` object.

**CLConstantNameAndTypeInfo** - Class in `jminusminus`

Representation of `CONSTANT_NameAndType_info` structure (JVM Spec Section 4.5.6).

**CLConstantNameAndTypeInfo(int, int)** - Constructor for class `jminusminus.CLConstantNameAndTypeInfo`

Construct a `CLConstantNameAndTypeInfo` object.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Construct a `CLConstantNameAndTypeInfo` object.

**CLConstantPool** - Class in `jminusminus`

Representation of a class' constant\_pool table (JVM Spec Section 4.5).

**CLConstantPool()** - Constructor for class `jminusminus.CLConstantPool`

Construct a `CLConstantPool` object.

**CLConstants** - Class in `jminusminus`

Constants used within `CL*.java` files.

**CLConstants()** - Constructor for class `jminusminus.CLConstants`

**CLConstants.Category** - Enum in `jminusminus`

We classify the JVM instructions into the following categories.

**CLConstantStringInfo** - Class in `jminusminus`

Representation of `CONSTANT_String_info` structure (JVM Spec Section 4.5.3).

**CLConstantStringInfo(int)** - Constructor for class `jminusminus.CLConstantStringInfo`

Construct a `CLConstantStringInfo` object.

**CLConstantUtf8Info** - Class in `jminusminus`

Representation of `CONSTANT_Utf8_info` structure (JVM Spec Section 4.5.7).

**CLConstantUtf8Info(byte[])** - Constructor for class `jminusminus.CLConstantUtf8Info`

Construct a `CLConstantUtf8Info` object.

**CLConstantValueAttribute** - Class in `jminusminus`

Representation of `ConstantValue_attribute` structure (JVM Spec Section 4.8.2).

**CLConstantValueAttribute(long, long, int)** - Constructor for class `jminusminus.CLConstantValueAttribute`

Construct a `CLConstantValueAttribute` object.

**CLConversionInstruction** - Class in `jminusminus`

Representation for `CONVERSION` instructions.

**CLConversionInstruction(int, int)** - Constructor for class `jminusminus.CLConversionInstruction`

Construct a `CLConversionInstruction` object.

**CLCPIInfo** - Class in `jminusminus`

Representation of `cp_info` structure (JVM Spec Section 4.5).

**CLCPIInfo()** - Constructor for class `jminusminus.CLCPIInfo`

**CLDeprecatedAttribute** - Class in `jminusminus`

Representation of `Deprecated_attribute` structure (JVM Spec Section 4.8.14).

**CLDeprecatedAttribute(int, long)** - Constructor for class `jminusminus.CLDeprecatedAttribute`

Construct a `CLDeprecatedAttribute` object.

**CLElementValue** - Class in `jminusminus`

Representation of `element_value` union (JVM Spec Section 4.8.15.1).

**CLElementValue(short, int)** - Constructor for class `jminusminus.CLElementValue`

Construct a `CLElementValue` object.

**CLElementValue(int, int)** - Constructor for class `jminusminus.CLElementValue`

Construct a `CLElementValue` object.

**CLElementValue(int)** - Constructor for class `jminusminus.CLElementValue`

Construct a `CLElementValue` object.

**CLElementValue(CLAnnotation)** - Constructor for class `jminusminus.CLElementValue`

Construct a `CLElementValue` object.

**CLElementValue(int, ArrayList<CLElementValue>)** - Constructor for class `jminusminus.CLElementValue`

Construct a `CLElementValue` object.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**CLElementValuePair** - Class in `jminusminus`

Representation of the `element_value_pairs` table entry (JVM Spec Section 4.8.15).

**CLElementValuePair(int, CLElementValue)** - Constructor for class `jminusminus.CLElementValuePair`

Construct a `CLElementValuePair` object.

**CLEmitter** - Class in `jminusminus`

This class provides a high level interface for creating (in-memory and file based) representation of Java classes.

**CLEmitter(boolean)** - Constructor for class `jminusminus.CLEmitter`

Construct a `CLEmitter` instance.

**CLEnclosingMethodAttribute** - Class in `jminusminus`

Representation of `EnclosingMethod_attribute` structure (JVM Spec Section 4.8.6).

**CLEnclosingMethodAttribute(int, long, int, int)** - Constructor for class `jminusminus.CLEnclosingMethodAttribute`

Construct a `CLEnclosingMethodAttribute` object.

**CLException** - Class in `jminusminus`

Representation of an exception handler.

**CLException(String, String, String, String)** - Constructor for class `jminusminus.CLException`

Construct a `CLException` object.

**CLExceptionInfo** - Class in `jminusminus`

Representation of `exception_table` entry structure (JVM Spec Section 4.8.3).

**CLExceptionInfo(int, int, int, int)** - Constructor for class `jminusminus.CLExceptionInfo`

Construct a `CLExceptionInfo` object.

**CLExceptionsAttribute** - Class in `jminusminus`

Representation of `Exceptions_attribute` structure (JVM Spec Section 4.8.4).

**CLExceptionsAttribute(int, long, int, ArrayList<Integer>)** - Constructor for class `jminusminus.CLExceptionsAttribute`

Construct a `CLExceptionsAttribute` object.

**CLFieldInfo** - Class in `jminusminus`

Representation of `field_info` structure (JVM Spec Section 4.6).

**CLFieldInfo(int, int, int, int, ArrayList<CLAttributeInfo>)** - Constructor for class `jminusminus.CLFieldInfo`

Construct a `CLFieldInfo` object.

**CLFieldInstruction** - Class in `jminusminus`

Representation for `FIELD` instructions.

**CLFieldInstruction(int, int, int, int)** - Constructor for class `jminusminus.CLFieldInstruction`

Construct a `CLFieldInstruction` object.

**clFile()** - Method in class `jminusminus.CLEmitter`

Return the `CLFile` instance corresponding to the class built by this emitter.

**CLFile** - Class in `jminusminus`

Representation of the `ClassFile` structure (JVM Spec Section 4.2).

**CLFile()** - Constructor for class `jminusminus.CLFile`

**clFiles()** - Method in class `jminusminus.JCompilationUnit`

Return the list of `CLFile` objects corresponding to the type declarations in this compilation unit.

**CLFlowControlInstruction** - Class in `jminusminus`

Representation for `FLOW_CONTROL1`, `FLOW_CONTROL2`, `FLOW_CONTROL3` and `FLOW_CONTROL4` instructions.

**CLFlowControlInstruction(int, int, String)** - Constructor for class `jminusminus.CLFlowControlInstruction`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Construct a CLFlowControlInstruction object for FLOW\_CONTROL1 instructions.

**CLFlowControlInstruction(int, int, boolean)** - Constructor for class jminusminus.CLFlowControlInstruction

Construct a CLFlowControlInstruction object for RET instruction.

**CLFlowControlInstruction(int, int, String, int, int, ArrayList<String>)** - Constructor for class jminusminus.CLFlowControlInstruction

Construct a CLFlowControlInstruction object for TABLESWITCH instruction.

**CLFlowControlInstruction(int, int, String, int, TreeMap<Integer, String>)** - Constructor for class jminusminus.CLFlowControlInstruction

Construct a CLFlowControlInstruction object for LOOKUPSWITCH instruction.

**CLInnerClassesAttribute** - Class in jminusminus

Representation of InnerClasses\_attribute structure (JVM Spec Section 4.8.5).

**CLInnerClassesAttribute(int, long, int, ArrayList<CLInnerClassInfo>)** - Constructor for class jminusminus.CLInnerClassesAttribute

Construct a CLInnerClassesAttribute object.

**CLInnerClassInfo** - Class in jminusminus

Representation of classes table entry structure (JVM Spec Section 4.8.5).

**CLInnerClassInfo(int, int, int, int)** - Constructor for class jminusminus.CLInnerClassInfo

Construct a CLInnerClassInfo object.

**CLInputStream** - Class in jminusminus

Inherits from java.io.DataInputStream and provides an extra function for reading unsigned int from the input stream, which is required for reading Java class files.

**CLInputStream(InputStream)** - Constructor for class jminusminus.CLInputStream

Construct a CLInputStream object from the specified input stream.

**CLInsInfo** - Class in jminusminus

This class stores static information about a JVM instruction.

**CLInsInfo(int, String, int, int, int, CLConstants.Category)** - Constructor for class jminusminus.CLInsInfo

Construct a CLInsInfo object.

**CLInstruction** - Class in jminusminus

Representation of a JVM instruction.

**CLInstruction()** - Constructor for class jminusminus.CLInstruction

**CLLineNumberInfo** - Class in jminusminus

Representation of line\_number\_table entry structure (JVM Spec Section 4.8.11).

**CLLineNumberInfo(int, int)** - Constructor for class jminusminus.CLLineNumberInfo

Construct a CLLineNumberInfo object.

**CLLineNumberTableAttribute** - Class in jminusminus

Representation of LineNumberTable\_attribute structure (JVM Spec Section 4.8.11).

**CLLineNumberTableAttribute(int, long, int, ArrayList<CLLineNumberInfo>)** - Constructor for class jminusminus.CLLineNumberTableAttribute

Construct a CLLineNumberTableAttribute object.

**CLLoadStoreInstruction** - Class in jminusminus

Representation for LOAD\_STORE1, LOAD\_STORE2, LOAD\_STORE3 and LOAD\_STORE4 instructions.

**CLLoadStoreInstruction(int, int)** - Constructor for class jminusminus.CLLoadStoreInstruction

Construct a CLLoadStoreInstruction object for LOAD\_STORE1 instructions.

**CLLoadStoreInstruction(int, int, int, boolean)** - Constructor for class jminusminus.CLLoadStoreInstruction

Construct a CLLoadStoreInstruction object for LOAD\_STORE2 instructions.

**CLLoadStoreInstruction(int, int, int)** - Constructor for class

**jminusminus.CCLoadStoreInstruction**

Construct a CCLoadStoreInstruction object for LOAD\_STORE3 and LOAD\_STORE4 instructions.

**CLLocalVariableInfo** - Class in jminusminus

Representation of local\_variable\_table entry structure (JVM Spec Section 4.8.12).

**CLLocalVariableInfo(int, int, int, int, int)** - Constructor for class

jminusminus.CCLocalVariableInfo

Construct a CCLocalVariableInfo object.

**CLLocalVariableTableAttribute** - Class in jminusminus

Representation of LocalVariableTable\_attribute structure (JVM Spec Section 4.8.12).

**CLLocalVariableTableAttribute(int, long, int, ArrayList<CLLocalVariableInfo>)** -

Constructor for class jminusminus.CCLocalVariableTableAttribute

Construct a CCLocalVariableTableAttribute object.

**CLLocalVariableTypeInfo** - Class in jminusminus

Representation of local\_variable\_type\_table entry structure (JVM Spec Section 4.8.13).

**CLLocalVariableTypeInfo(int, int, int, int, int)** - Constructor for class

jminusminus.CCLocalVariableTypeInfo

Construct a CCLocalVariableTypeInfo object.

**CLLocalVariableTypeTableAttribute** - Class in jminusminus

Representation of LocalVariableTypeTable\_attribute structure (JVM Spec Section 4.8.12).

**CLLocalVariableTypeTableAttribute(int, long, int, ArrayList<CLLocalVariableTypeInfo>)** -

Constructor for class jminusminus.CCLocalVariableTypeTableAttribute

Construct a CCLocalVariableTypeTableAttribute object.

**CLMemberInfo** - Class in jminusminus

Abstraction of field\_info and method\_info structures (JVM Spec Section 4.6, 4.7).

**CLMemberInfo(int, int, int, int, ArrayList<CLAttributeInfo>)** - Constructor for class

jminusminus.CLMemberInfo

Construct a CLMemberInfo object.

**CLMethodInfo** - Class in jminusminus

Representation of method\_info structure (JVM Spec Section 4.7).

**CLMethodInfo(int, int, int, int, ArrayList<CLAttributeInfo>)** - Constructor for class

jminusminus.CLMMethodInfo

Construct a CLMethodInfo object.

**CLMethodInstruction** - Class in jminusminus

Representation for METHOD1 and METHOD2 instructions.

**CLMethodInstruction(int, int, int, int)** - Constructor for class

jminusminus.CLMMethodInstruction

Construct a CLMethodInstruction object for METHOD1 instructions.

**CLMethodInstruction(int, int)** - Constructor for class jminusminus.CLMMethodInstruction

Construct a CLMethodInstruction object for METHOD2 instructions.

**CLMiscInstruction** - Class in jminusminus

Representation for MISC instructions.

**CLMiscInstruction(int, int)** - Constructor for class jminusminus.CLMiscInstruction

Construct a CLMiscInstruction object.

**CLObjectInstruction** - Class in jminusminus

Representation for OBJECT instructions.

**CLObjectInstruction(int, int, int)** - Constructor for class jminusminus.CLObjectInstruction

Construct a CLObjectInstruction object.

**close()** - Method in class jminusminus.CharReader

Close the file.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



**close(int)** - Static method in class `spim.SPIM`

Wrapper for `SPIM.close()`.

**CLOutputStream** - Class in `jminusminus`

Inherits from `java.out.DataOutputStream` and provides an extra function for writing unsigned int to the output stream, which is required for writing Java class files.

**CLOutputStream(OutputStream)** - Constructor for class `jminusminus.CLOutputStream`

Construct a `CLOutputStream` from the specified output stream.

**CLParameterAnnotationInfo** - Class in `jminusminus`

Representation of `parameter_annotations_table` entry structure (JVM Spec Section 4.8.17).

**CLParameterAnnotationInfo(int, ArrayList<CLAnnotation>)** - Constructor for class `jminusminus.CLParameterAnnotationInfo`

Construct a `ParameterAnnotationInfo` object.

**CLPath** - Class in `jminusminus`

This class can be used to locate and load system, extension, and user-defined class files from directories and zip (jar) files.

**CLPath()** - Constructor for class `jminusminus.CLPath`

Construct a `CLPath` object.

**CLPath(String, String)** - Constructor for class `jminusminus.CLPath`

Construct a `CLPath` object.

**CLRRuntimeInvisibleAnnotationsAttribute** - Class in `jminusminus`

Representation of `RuntimeInvisibleAnnotations_attribute` structure (JVM Spec Section 4.8.16).

**CLRRuntimeInvisibleAnnotationsAttribute(int, long, int, ArrayList<CLAnnotation>)** - Constructor for class `jminusminus.CLRRuntimeInvisibleAnnotationsAttribute`

Construct a `CLRRuntimeInvisibleAnnotationsAttribute` object.

**CLRRuntimeInvisibleParameterAnnotationsAttribute** - Class in `jminusminus`

Representation of `RuntimeInvisibleParameterAnnotations_attribute` structure (JVM Spec Section 4.8.18).

**CLRRuntimeInvisibleParameterAnnotationsAttribute(int, long, short, ArrayList<CLParameterAnnotationInfo>)** - Constructor for class `jminusminus.CLRRuntimeInvisibleParameterAnnotationsAttribute`

Construct a `CLRRuntimeInvisibleParameterAnnotationsAttribute` object.

**CLRRuntimeVisibleAnnotationsAttribute** - Class in `jminusminus`

Representation of `RuntimeVisibleAnnotations_attribute` structure (JVM Spec Section 4.8.15).

**CLRRuntimeVisibleAnnotationsAttribute(int, long, int, ArrayList<CLAnnotation>)** - Constructor for class `jminusminus.CLRRuntimeVisibleAnnotationsAttribute`

Construct a `CLRRuntimeVisibleAnnotationsAttribute` object.

**CLRRuntimeVisibleParameterAnnotationsAttribute** - Class in `jminusminus`

Representation of `RuntimeVisibleParameterAnnotations_attribute` structure (JVM Spec Section 4.8.17).

**CLRRuntimeVisibleParameterAnnotationsAttribute(int, long, short, ArrayList<CLParameterAnnotationInfo>)** - Constructor for class `jminusminus.CLRRuntimeVisibleParameterAnnotationsAttribute`

Construct a `CLRRuntimeVisibleParameterAnnotationsAttribute` object.

**CLSignatureAttribute** - Class in `jminusminus`

Representation of `Signature_attribute` structure (JVM Spec Section 4.8.8).

**CLSignatureAttribute(int, long, int)** - Constructor for class `jminusminus.CLSignatureAttribute`

Construct a `CLSignatureAttribute` object.

**CLSourceDebugExtensionAttribute** - Class in `jminusminus`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Representation of SourceDebugExtension\_attribute structure (JVM Spec Section 4.8.10).

**CLSourceDebugExtensionAttribute(int, long, byte[])** - Constructor for class jminusminus.CLSourceDebugExtensionAttribute

Construct a CLSourceDebugExtensionAttribute object.

**CLSourceFileAttribute** - Class in jminusminus

Representation of SourceFile\_attribute structure (JVM Spec Section 4.8.9).

**CLSourceFileAttribute(int, long, int)** - Constructor for class jminusminus.CLSourceFileAttribute

Construct a CLSourceFileAttribute object.

**CLStackInstruction** - Class in jminusminus

Representation for STACK instructions.

**CLStackInstruction(int, int)** - Constructor for class jminusminus.CLStackInstruction

Construct a CLStackInstruction object.

**CLSyntheticAttribute** - Class in jminusminus

Representation of Synthetic\_attribute structure (JVM Spec Section 4.8.7).

**CLSyntheticAttribute(int, long)** - Constructor for class jminusminus.CLSyntheticAttribute

Construct a CLSyntheticAttribute object.

**code** - Variable in class jminusminus.CLCodeAttribute

Code\_attribute.code item.

**codegen(CLEmitter)** - Method in class jminusminus.Boxing

**codegen(CLEmitter)** - Method in interface jminusminus.Converter  
Emit code necessary to convert (cast) a source type to a target type.

**codegen(CLEmitter)** - Method in class jminusminus.I2C

**codegen(CLEmitter)** - Method in class jminusminus.Identity

**codegen(CLEmitter)** - Method in class jminusminus.JArrayExpression

Perform code generation from the JArrayExpression using the specified code emitter.

**codegen(CLEmitter)** - Method in class jminusminus.JArrayInitializer

Perform code generation necessary to construct the initializing array and leave it on top of the stack.

**codegen(CLEmitter)** - Method in class jminusminus.JAssignOp

Code generation for an assignment involves, generating code for loading any necessary Lvalue onto the stack, for loading the Rvalue, for (unless a statement) copying the Rvalue to its proper place on the stack, and for doing the store.

**codegen(CLEmitter)** - Method in class jminusminus.JAST

Perform code generation for this AST.

**codegen(CLEmitter)** - Method in class jminusminus.JBlock

Generating code for a block consists of generating code for each of its statements.

**codegen(CLEmitter)** - Method in class jminusminus.JBooleanBinaryExpression

Generate code for the case where we actually want a boolean value (true or false) computed onto the stack, eg for assignment to a boolean variable.

**codegen(CLEmitter)** - Method in class jminusminus.JCastOp

Generating code for a cast expression involves generating code for the original expr, and then for any necessary conversion.

**codegen(CLEmitter)** - Method in class jminusminus.JClassDeclaration

Generate code for the class declaration.

**codegen(CLEmitter)** - Method in class jminusminus.JCompilationUnit

Generating code for a compilation unit means generating code for each of the type declarations.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**codegen(CLEmitter)** - Method in class jminusminus.JConstructorDeclaration

Generate code for the constructor declaration.

**codegen(CLEmitter)** - Method in class jminusminus.JEmptyStatement

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JEqualOp

Branching code generation for == operation.

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JExpression

Perform (short-circuit) code generation for a boolean expression, given the code emitter, a target label, and whether we branch to that label on true or on false.

**codegen(CLEmitter)** - Method in class jminusminus.JFieldDeclaration

Code generation for field declaration involves generate field the header.

**codegen(CLEmitter)** - Method in class jminusminus.JFieldSelection

Generate the code necessary to load the Rvalue for this field selection.

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JFieldSelection

The semantics of j-- require that we implement short-circuiting branching in implementing field selections.

**codegen(CLEmitter)** - Method in class jminusminus.JFormalParameter

No code generated here.

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JGreaterThanOp

Branching code generation for > operation.

**codegen(CLEmitter)** - Method in class jminusminus.JIfStatement

Code generation for an if statement.

**codegen(CLEmitter)** - Method in class jminusminus.JInstanceOfOp

Generate code for the type test.

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JInstanceOfOp

Short-circuiting branching for instanceof.

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JLessEqualOp

Branching code generation for <= operation.

**codegen(CLEmitter)** - Method in class jminusminus.JLiteralChar

Generating code for a char literal means generating code to push it onto the stack.

**codegen(CLEmitter)** - Method in class jminusminus.JLiteralFalse

Generating code for a boolean literal means generating code to push it onto the stack.

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JLiteralFalse

Generating branch code for a boolean literal is trivial; it's either empty or an unconditional branch.

**codegen(CLEmitter)** - Method in class jminusminus.JLiteralInt

Generating code for an int literal means generating code to push it onto the stack.

**codegen(CLEmitter)** - Method in class jminusminus.JLiteralNull

Generating code for a null literal means generating code to push it onto the stack.

**codegen(CLEmitter)** - Method in class jminusminus.JLiteralString

Generating code for a string literal means generating code to push it onto the stack.

**codegen(CLEmitter)** - Method in class jminusminus.JLiteralTrue

Generating code for a boolean literal means generating code to push it onto the stack.

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JLiteralTrue

Generating branch code for a boolean literal is trivial; it's either empty or an unconditional branch.

**codegen(CLEmitter, String, boolean)** - Method in class jminusminus.JLogicalAndOp

The semantics of j-- require that we implement short-circuiting branching in implementing the logical AND.

**codegen(CLEmitter)** - Method in class jminusminus.JLogicalNotOp

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Generate code for the case where we actually want a boolean value (true or false) computed onto the stack, eg for assignment to a boolean variable.

**codegen(CLEmitter, String, boolean)** - Method in class `jminusminus.JLogicalNotOp`

The code generation necessary for branching simply flips the condition on which we branch.

**codegen(CLEmitter)** - Method in class `jminusminus.JMessageExpression`

Code generation for a message expression involves generating code for loading the target onto the stack, generating code to load the actual arguments onto the stack, and then invoking the named Method.

**codegen(CLEmitter, String, boolean)** - Method in class `jminusminus.JMessageExpression`

The semantics of `j--` require that we implement short-circuiting branching in implementing message expressions.

**codegen(CLEmitter)** - Method in class `jminusminus.JMethodDeclaration`

Generate code for the method declaration.

**codegen(CLEmitter)** - Method in class `jminusminus.JMultiplyOp`

Generating code for the `*` operation involves generating code for the two operands, and then the multiplication instruction.

**codegen(CLEmitter)** - Method in class `jminusminus.JNegateOp`

Generating code for the negation operation involves generating code for the operand, and then the negation instruction.

**codegen(CLEmitter)** - Method in class `jminusminus.JNewArrayOp`

Generate code to push the bounds on the stack and then generate the appropriate array creation instruction.

**codegen(CLEmitter)** - Method in class `jminusminus.JNewOp`

Generating code for a new operation involves generating the `NEW` instruction for creating the object on the stack, then generating the code for the actual arguments, and then code for invoking the constructor (the initialization method).

**codegen(CLEmitter)** - Method in class `jminusminus.JPlusAssignOp`

Code generation for `+=` involves generating code for loading any necessary l-value onto the stack, for (unless a string concatenation) loading the r-value, for (unless a statement) copying the r-value to its proper place on the stack, and for doing the store.

**codegen(CLEmitter)** - Method in class `jminusminus.JPlusOp`

Any string concatenation has been rewritten as a `JStringConcatenationOp` (in `analyze()`), so code generation here involves simply generating code for loading the operands onto the stack and then generating the appropriate add instruction.

**codegen(CLEmitter)** - Method in class `jminusminus.JPostDecrementOp`

In generating code for a post-decrement operation, we treat simple variable (`JVariable`) operands specially since the JVM has an increment instruction.

**codegen(CLEmitter)** - Method in class `jminusminus.JPreIncrementOp`

In generating code for a pre-increment operation, we treat simple variable (`JVariable`) operands specially since the JVM has an increment instruction.

**codegen(CLEmitter)** - Method in class `jminusminus.JReturnStatement`

Generate code for the return statement.

**codegen(CLEmitter)** - Method in class `jminusminus.JStatementExpression`

Generating code for the statement expression involves simply generating code for the encapsulated expression.

**codegen(CLEmitter)** - Method in class `jminusminus.JStringConcatenationOp`

Code generation generates code for creating a `StringBuilder` atop the runtime stack, appending the operands (which might contain nested concatenations; these are handled by `cascadingCodegen()`), and then for converting the `StringBuilder` to a `String`.

**codegen(CLEmitter)** - Method in class `jminusminus.JSubtractOp`

Generating code for the `-` operation involves generating code for the two operands, and

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



then the subtraction instruction.

**codegen(CLEmitter)** - Method in class `jminusminus.JSuper`

Load "this" onto the stack (even if we treat it as its super class).

**codegen(CLEmitter)** - Method in class `jminusminus.JSuperConstruction`

Code generation involves generating code to load the actual arguments onto the stack, and then the code for invoking the constructor.

**codegen(CLEmitter)** - Method in class `jminusminus.JThis`

Simply generate code to load "this" onto the stack.

**codegen(CLEmitter)** - Method in class `jminusminus.JThisConstruction`

Code generation involves generating the code for loading the actual arguments onto the stack, and then for invoking this constructor.

**codegen(CLEmitter)** - Method in class `jminusminus.JVariable`

Generate code to load value of variable on stack.

**codegen(CLEmitter, String, boolean)** - Method in class `jminusminus.JVariable`

The semantics of `j--` require that we implement short-circuiting branching in implementing the identifier expression.

**codegen(CLEmitter)** - Method in class `jminusminus.JVariableDeclaration`

Local variable initializations (rewritten as assignments in `analyze()`) are generated here.

**codegen(CLEmitter)** - Method in class `jminusminus.JVariableDeclarator`

No code generation is done here.

**codegen(CLEmitter)** - Method in class `jminusminus.JWhileStatement`

Generate code for the while loop

**codegen(CLEmitter)** - Method in class `jminusminus.JWildExpression`

No code generation.

**codegen(CLEmitter)** - Method in class `jminusminus.NarrowReference`

**codegen(CLEmitter)** - Method in class `jminusminus.UnBoxing`

**codegenDuplicateRvalue(CLEmitter)** - Method in class `jminusminus.JArrayExpression`

Generate the code required for duplicating the Rvalue that is on the stack because it is to be used in a surrounding expression, as in `a[i] = x =` or `x = y--`.

**codegenDuplicateRvalue(CLEmitter)** - Method in class `jminusminus.JFieldSelection`

Generate the code required for duplicating the Rvalue that is on the stack because it is to be used in a surrounding expression, as in `a[i] = x =` or `x = y--`.

**codegenDuplicateRvalue(CLEmitter)** - Method in interface `jminusminus.JLhs`

Generate the code required for duplicating the Rvalue that is on the stack because it is to be used in a surrounding expression, as in `a[i] = x =` or `x = y--`.

**codegenDuplicateRvalue(CLEmitter)** - Method in class `jminusminus.JVariable`

Generate the code required for duplicating the Rvalue that is on the stack because it is to be used in a surrounding expression, as in `a[i] = x =` or `x = y--`.

**codegenInitializations(CLEmitter)** - Method in class `jminusminus.JFieldDeclaration`

Generate code for any field initializations (now rewritten as assignment statements).

**codegenLoadLhsLvalue(CLEmitter)** - Method in class `jminusminus.JArrayExpression`

Generate the code required for setting up an Lvalue, eg for use in an assignment.

**codegenLoadLhsLvalue(CLEmitter)** - Method in class `jminusminus.JFieldSelection`

Generate the code required for setting up an Lvalue, eg, for use in an assignment.

**codegenLoadLhsLvalue(CLEmitter)** - Method in interface `jminusminus.JLhs`

The up front code necessary for implementing an assignment; it generates code to load onto the stack any part of the lhs variable that must be there.

**codegenLoadLhsLvalue(CLEmitter)** - Method in class `jminusminus.JVariable`

Generate the code required for setting up an Lvalue, eg for use in an assignment.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



**codegenLoadLhsRvalue(CLEmitter)** - Method in class jminusminus.JArrayExpression  
Generate the code required for loading an Rvalue for this variable, eg for use in a +=.

**codegenLoadLhsRvalue(CLEmitter)** - Method in class jminusminus.JFieldSelection  
Generate the code required for loading an Rvalue for this variable, eg for use in a +=.

**codegenLoadLhsRvalue(CLEmitter)** - Method in interface jminusminus.JLhs  
Generate the code required for loading an Rvalue for this variable, as in a +=.

**codegenLoadLhsRvalue(CLEmitter)** - Method in class jminusminus.JVariable  
Generate the code required for loading an Rvalue for this variable, eg for use in a +=.

**codegenStore(CLEmitter)** - Method in class jminusminus.JArrayExpression  
Generate the code required for doing the actual assignment.

**codegenStore(CLEmitter)** - Method in class jminusminus.JFieldSelection  
Generate the code required for doing the actual assignment.

**codegenStore(CLEmitter)** - Method in interface jminusminus.JLhs  
Generate the code required for doing the actual assignment.

**codegenStore(CLEmitter)** - Method in class jminusminus.JVariable  
Generate the code required for doing the actual assignment.

**codeLength** - Variable in class jminusminus.CLCodeAttribute  
Code\_attribute.code\_length item.

**column** - Variable in class jminusminus.SimpleCharStream

**COMMA** - Static variable in interface jminusminus.JavaCCParserConstants

**compareTo(NInterval)** - Method in class jminusminus.NInterval

Compare start positions (for ordering intervals).

**compareTo(NRange)** - Method in class jminusminus.NRange

One liveness range comes before another if its start position comes before the other's start position.

**compilationUnit** - Static variable in class jminusminus.JAST

Current compilation unit (set in JCompilationUnit).

**compilationUnit()** - Method in class jminusminus.JavaCCParser

**compilationUnit()** - Method in class jminusminus.Parser

Parse a compilation unit (a program file) and construct an AST for it.

**CompilationUnitContext** - Class in jminusminus

The compilation unit context is always the outermost context, and is where imported types and locally defined types (classes) are declared.

**CompilationUnitContext()** - Constructor for class jminusminus.CompilationUnitContext

Construct a new compilation unit context.

**compilationUnitContext** - Variable in class jminusminus.Context

The compilation unit context (for the whole source program or file).

**compilationUnitContext()** - Method in class jminusminus.Context

Return the surrounding compilation unit context.

**componentType()** - Method in class jminusminus.ArrayTypeName

Return the (component) type of its elements.

**componentType()** - Method in class jminusminus.Type

An array type's component type.

**computeDominators(NBasicBlock, NBasicBlock)** - Method in class jminusminus.NControlFlowGraph

Compute the dominator of each block in this cfg recursively given the starting block and its predecessor.

**confirmMethodHasReturn()** - Method in class jminusminus.MethodContext

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Record fact that (non-void) method has at least one return.

**CONSTANT\_Class** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_Class\_info constant pool structure.

**CONSTANT\_Double** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_Double\_info constant pool structure.

**CONSTANT\_Fieldref** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_Fieldref\_info constant pool structure.

**CONSTANT\_Float** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_Float\_info constant pool structure.

**CONSTANT\_Integer** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_Integer\_info constant pool structure.

**CONSTANT\_InterfaceMethodref** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_InterfaceMethodref\_info constant pool structure.

**CONSTANT\_Long** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_Long\_info constant pool structure.

**CONSTANT\_Methodref** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_Methodref\_info constant pool structure.

**CONSTANT\_NameAndType** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_NameAndType\_info constant pool structure.

**CONSTANT\_String** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_String\_info constant pool structure.

**CONSTANT\_Utf8** - Static variable in class jminusminus.CLConstants

Identifies CONSTANT\_Utf8\_info constant pool structure.

**constantClassInfo(String)** - Method in class jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of CLConstantClassInfo.

**constantDoubleInfo(double)** - Method in class jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of CLConstantDoubleInfo.

**constantFieldRefInfo(String, String, String)** - Method in class jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of CLConstantFieldRefInfo.

**constantFloatInfo(float)** - Method in class jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of CLConstantFloatInfo.

**constantIntegerInfo(int)** - Method in class jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of CLConstantIntegerInfo.

**constantInterfaceMethodRefInfo(String, String, String)** - Method in class

jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of

CLConstantInterfaceMethodRefInfo.

**constantLongInfo(long)** - Method in class jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of CLConstantLongInfo.

**constantMethodRefInfo(String, String, String)** - Method in class

jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of CLConstantMethodRefInfo.

**constantNameAndTypeInfo(String, String)** - Method in class jminusminus.CLConstantPool

Return the constant pool index of a singleton instance of CLConstantNameAndTypeInfo.

**constantPool()** - Method in class jminusminus.CLEmitter

Return the constant pool of the class being built.

**constantPool** - Variable in class jminusminus.CLFile

ClassFile.constant\_pool item.

**constantPoolCount** - Variable in class jminusminus.CLFile

ClassFile.constant\_pool\_count item.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**constantStringInfo(String)** - Method in class `jminusminus.CLConstantPool`  
Return the constant pool index of a singleton instance of `CLConstantStringInfo`.

**constantUtf8Info(String)** - Method in class `jminusminus.CLConstantPool`  
Return the constant pool index of a singleton instance of `CLConstantUtf8Info`.

**constantValueIndex** - Variable in class `jminusminus.CLConstantValueAttribute`  
`ConstantValue_attribute.constantvalue_index` item.

**constNameIndex** - Variable in class `jminusminus.CLElementValue`  
`element_value.enum_const_value.const_name_index` item.

**Constructor** - Class in `jminusminus`  
A Constructor knows its JVM descriptor.

**Constructor(Constructor)** - Constructor for class `jminusminus.Constructor`  
Construct a Constructor from its internal representation.

**constructor** - Variable in class `jminusminus.Constructor`  
Internal representation of this constructor.

**CONSTRUCTOR** - Static variable in class `jminusminus.Type`  
A type marker indicating a constructor (having no return type).

**constructorFor(Type[])** - Method in class `jminusminus.Type`  
Find an appropriate constructor in this type, given it's argument types.

**constValueIndex** - Variable in class `jminusminus.CLElementValue`  
`element_value.const_value_index` item.

**Context** - Class in `jminusminus`  
A Context encapsulates the environment in which an AST is analyzed.

**Context(Context, ClassContext, CompilationUnitContext)** - Constructor for class `jminusminus.Context`  
Construct a Context.

**context** - Variable in class `jminusminus.JMethodDeclaration`  
Built in `analyze()`.

**Conversions** - Class in `jminusminus`  
A 2-D table of conversions, from one type to another.

**Conversions()** - Constructor for class `jminusminus.Conversions`  
Construct a table of conversions and populate it.

**Converter** - Interface in `jminusminus`  
A Converter encapsulates any (possibly none) code necessary to perform a cast operation.

**cplIndex** - Variable in class `jminusminus.CLCPInfo`  
Index of this object into the constant pool.

**cplItem(int)** - Method in class `jminusminus.CLConstantPool`  
Return the constant pool item at the specified index, or null if the index is invalid.

**createLabel()** - Method in class `jminusminus.CLEmitter`  
Construct and return a unique jump label.

**curChar** - Variable in class `jminusminus.JavaCCParserTokenManager`

**curLexState** - Variable in class `jminusminus.JavaCCParserTokenManager`

**currentToken** - Variable in exception `jminusminus.ParseException`  
This is the last token that has been consumed successfully.

## D

**d** - Variable in class `jminusminus.CLConstantDoubleInfo`  
The double precision floating-point number.

**D2F** - Static variable in class jminusminus.CLConstants  
D2F instruction.

**D2I** - Static variable in class jminusminus.CLConstants  
D2I instruction.

**D2L** - Static variable in class jminusminus.CLConstants  
D2L instruction.

**DADD** - Static variable in class jminusminus.CLConstants  
DADD instruction.

**DALOAD** - Static variable in class jminusminus.CLConstants  
DALOAD instruction.

**DASTORE** - Static variable in class jminusminus.CLConstants  
DASTORE instruction.

**data** - Variable in class jminusminus.NControlFlowGraph  
SPIM code for string literals added to the data segment.

**DCMPG** - Static variable in class jminusminus.CLConstants  
DCMPG instruction.

**DCMPL** - Static variable in class jminusminus.CLConstants  
DCMPL instruction.

**DCONST\_0** - Static variable in class jminusminus.CLConstants  
DCONST\_0 instruction.

**DCONST\_1** - Static variable in class jminusminus.CLConstants  
DCONST\_1 instruction.

**DDIV** - Static variable in class jminusminus.CLConstants  
DDIV instruction.

**debugExtension** - Variable in class jminusminus.SourceDebugExtensionAttribute  
SourceDebugExtension.debug\_extension item.

**debugStream** - Variable in class jminusminus.JavaCCParserTokenManager

**DEC** - Static variable in interface jminusminus.JavaCCParserConstants

**declareThisType(Context)** - Method in class jminusminus.JClassDeclaration  
Declare this class in the parent (compilation unit) context.

**declareThisType(Context)** - Method in interface jminusminus.JTypeDecl  
Even before preAnalyze(), declare this type in the parent context so that it is available in the preAnalyze() of other types.

**declaringType()** - Method in class jminusminus.Constructor  
Return the type declaring this constructor.

**declaringType()** - Method in class jminusminus.Member  
Return the type in which this member was declared.

**DEFAULT** - Static variable in interface jminusminus.JavaCCParserConstants

**defaultLexState** - Variable in class jminusminus.JavaCCParserTokenManager

**defaultValue** - Variable in class jminusminus.CLAnnotationDefaultAttribute  
AnnotationDefault\_attribute.defaultValue item.

**definingClass** - Variable in class jminusminus.JConstructorDeclaration  
Defining class

**definingType()** - Method in class jminusminus.Context  
The type that defines this context (used principally for checking accessibility).

**definition()** - Method in class jminusminus.ClassContext  
Return the AST node of the type defined by this class.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**desc** - Variable in class jminusminus.NControlFlowGraph  
Descriptor of the method this cfg corresponds to.

**descriptor** - Variable in class jminusminus.JMethodDeclaration  
Computed by preAnalyze().

**descriptorIndex** - Variable in class jminusminus.CLConstantNameAndTypeInfo  
CONSTANT\_NameAndType\_info.descriptor\_index item.

**descriptorIndex** - Variable in class jminusminus.CLLocalVariableInfo  
local\_variable\_table\_entry.descriptor\_index item.

**descriptorIndex** - Variable in class jminusminus.CLMemberInfo  
member\_info.descriptor\_index item.

**destination** - Variable in class jminusminus.NHIRGoto  
The destination block to unconditionally jump to.

**destinationDir(String)** - Method in class jminusminus.CLEmitter  
Set the destination directory for the class file to the specified value.

**destinationDir(String)** - Method in class jminusminus.NEmitter  
Set the destination directory for the SPIM files to the specified value.

**detectLoops(NBasicBlock, NBasicBlock)** - Method in class jminusminus.NControlFlowGraph  
Implements loop detection algorithm to figure out if the specified block is a loop head or a loop tail.

**DIGIT** - Static variable in interface jminusminus.JavaCCParserConstants

**dim** - Variable in class jminusminus.NHIRNewArray  
Dimension of the array.

**dim** - Variable in class jminusminus.NLIRNewArray  
Dimension of the array.

**disable\_tracing()** - Method in class jminusminus.JavaCCParser

**DLOAD** - Static variable in class jminusminus.CLConstants  
DLOAD instruction.

**DLOAD\_0** - Static variable in class jminusminus.CLConstants  
DLOAD\_0 instruction.

**DLOAD\_1** - Static variable in class jminusminus.CLConstants  
DLOAD\_1 instruction.

**DLOAD\_2** - Static variable in class jminusminus.CLConstants  
DLOAD\_2 instruction.

**DLOAD\_3** - Static variable in class jminusminus.CLConstants  
DLOAD\_3 instruction.

**DMUL** - Static variable in class jminusminus.CLConstants  
DMUL instruction.

**DNEG** - Static variable in class jminusminus.CLConstants  
DNEG instruction.

**dom** - Variable in class jminusminus.NBasicBlock  
The dominator of this block.

**Done()** - Method in class jminusminus.SimpleCharStream

**DOT** - Static variable in interface jminusminus.JavaCCParserConstants

**DREM** - Static variable in class jminusminus.CLConstants  
DREM instruction.

**DRETURN** - Static variable in class jminusminus.CLConstants  
DRETURN instruction.



**DSTORE** - Static variable in class jminusminus.CLConstants  
DSTORE instruction.

**DSTORE\_0** - Static variable in class jminusminus.CLConstants  
DSTORE\_0 instruction.

**DSTORE\_1** - Static variable in class jminusminus.CLConstants  
DSTORE\_1 instruction.

**DSTORE\_2** - Static variable in class jminusminus.CLConstants  
DSTORE\_2 instruction.

**DSTORE\_3** - Static variable in class jminusminus.CLConstants  
DSTORE\_3 instruction.

**DSUB** - Static variable in class jminusminus.CLConstants  
DSUB instruction.

**DUP** - Static variable in class jminusminus.CLConstants  
DUP instruction.

**DUP2** - Static variable in class jminusminus.CLConstants  
DUP2 instruction.

**DUP2\_X1** - Static variable in class jminusminus.CLConstants  
DUP2\_X1 instruction.

**DUP2\_X2** - Static variable in class jminusminus.CLConstants  
DUP2\_X2 instruction.

**DUP\_X1** - Static variable in class jminusminus.CLConstants  
DUP\_X1 instruction.

**DUP\_X2** - Static variable in class jminusminus.CLConstants  
DUP\_X2 instruction.

**DYNAMIC** - Static variable in class jminusminus.CLConstants  
Denotes values that are not statically known.

Assignment Project Exam Help

<https://powcoder.com>

## E

Add WeChat powcoder

**elementNameIndex** - Variable in class jminusminus.CLElementValuePair  
element\_value\_pairs\_table\_entry.element\_name\_index item.

**elementValuePairs** - Variable in class jminusminus.CLAnnotation  
annotation.element\_value\_pairs item.

**eliminateRedundantPhiFunctions()** - Method in class jminusminus.NControlFlowGraph  
Eliminate redundant phi functions of the form  $x = (y, x, x, \dots, x)$  with  $y$ .

**ELSE** - Static variable in interface jminusminus.JavaCCParserConstants

**ELT\_ANNOTATION** - Static variable in class jminusminus.CLConstants  
Identifies annotation type of annotation element value.

**ELT\_ARRAY** - Static variable in class jminusminus.CLConstants  
Identifies array type of annotation element value.

**ELT\_B** - Static variable in class jminusminus.CLConstants  
Identifies boolean type of annotation element value.

**ELT\_C** - Static variable in class jminusminus.CLConstants  
Identifies char type of annotation element value.

**ELT\_c** - Static variable in class jminusminus.CLConstants  
Identifies class type of annotation element value.

**ELT\_D** - Static variable in class jminusminus.CLConstants  
Identifies double type of annotation element value.

**ELT\_e** - Static variable in class jminusminus.CLConstants  
Identifies enum type of annotation element value.

**ELT\_F** - Static variable in class jminusminus.CLConstants

Identifies float type of annotation element value.

**ELT\_I** - Static variable in class jminusminus.CLConstants

Identifies int type of annotation element value.

**ELT\_J** - Static variable in class jminusminus.CLConstants

Identifies long type of annotation element value.

**ELT\_S** - Static variable in class jminusminus.CLConstants

Identifies short type of annotation element value.

**ELT\_s** - Static variable in class jminusminus.CLConstants

Identifies String type of annotation element value.

**ELT\_Z** - Static variable in class jminusminus.CLConstants

Identifies boolean type of annotation element value.

**EMPTY\_STACK** - Static variable in class jminusminus.CLConstants

Stack units for the instructions that empty the operand stack.

**enable\_tracing()** - Method in class jminusminus.JavaCCParser

**endColumn** - Variable in class jminusminus.Token

beginLine and beginColumn describe the position of the first character of this token;  
endLine and endColumn describe the position of the last character of this token.

**endLabel** - Variable in class jminusminus.CLEntry

this instruction.

**endLine** - Variable in class jminusminus.Token

beginLine and beginColumn describe the position of the first character of this token;  
endLine and endColumn describe the position of the last character of this token.

**endPC** - Variable in class jminusminus.CLEntry

endLabel is resolved to this value.

**endPC** - Variable in class jminusminus.CLEntryInfo

exception\_table\_entry.end\_pc item.

**endsAtBlock()** - Method in class jminusminus.Interval

The basic block in which this interval's end position falls.

**entries** - Variable in class jminusminus.Context

Map of (local variable, formal parameters, type) names to their definitions.

**EOF** - Static variable in interface jminusminus.JavaCCParserConstants

**EOFCH** - Static variable in class jminusminus.CharReader

A representation of the end of file as a character.

**EOFCH** - Static variable in class jminusminus.Scanner

End of file character.

**eol** - Variable in exception jminusminus.ParseException

The end of line string for this machine.

**EQUAL** - Static variable in interface jminusminus.JavaCCParserConstants

**equals(Object)** - Method in class jminusminus.CLConstantClassInfo

**equals(Object)** - Method in class jminusminus.CLConstantDoubleInfo

**equals(Object)** - Method in class jminusminus.CLConstantFloatInfo

**equals(Object)** - Method in class jminusminus.CLConstantIntegerInfo

**equals(Object)** - Method in class jminusminus.CLConstantLongInfo

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**equals(Object)** - Method in class jminusminus.CLConstantMemberRefInfo

**equals(Object)** - Method in class jminusminus.CLConstantNameAndTypeInfo

**equals(Object)** - Method in class jminusminus.CLConstantStringInfo

**equals(Object)** - Method in class jminusminus.CLConstantUtf8Info

**equals(Object)** - Method in class jminusminus.CLLineNumberInfo

Return true if this LineNumber\_info object is "equal to" the specified LineNumber\_info object, false otherwise.

**equals(Method)** - Method in class jminusminus.Method

Method equality is defined HERE as having override-equivalent signatures.

**equals(NBasicBlock)** - Method in class jminusminus.NBasicBlock

Is this block the same as the other block? Two blocks are the same if their ids are the same.

**equals(NHIRInstruction)** - Method in class jminusminus.NHIRInstruction

Return true if this instruction is the same as the other, false otherwise.

**equals(NInterval)** - Method in class jminusminus.NInterval

Two intervals are equal if they have the same virtual register ID.

**equals(Type)** - Method in class jminusminus.Type

Type equality is based on the equality of descriptors.

**ERROR** - Static variable in interface jminusminus.JavaCCParserConstants

**errorCode** - Variable in error jminusminus.TokenMgrError

Indicates the reason why the exception is thrown.

**errorHasOccured()** - Method in class jminusminus.LookaheadScanner

Has an error occurred up to now in lexical analysis?

**errorHasOccurred()** - Method in class jminusminus.CLAbsorber

Return true if an error had occurred while reading the class; false otherwise.

**errorHasOccurred()** - Method in class jminusminus.CLEmitter

Has an emitter error occurred up to now?

**errorHasOccurred()** - Static method in class jminusminus.JavaCCMain

Return true if an error occurred during compilation; false otherwise.

**errorHasOccurred()** - Method in class jminusminus.JavaCCParser

Has a parser error occurred up to now?

**errorHasOccurred()** - Method in class jminusminus.JCompilationUnit

Has a semantic error occurred up to now?

**errorHasOccurred()** - Static method in class jminusminus.Main

Return true if an error occurred during compilation; false otherwise.

**errorHasOccurred()** - Method in class jminusminus.NEmitter

Has an emitter error occurred up to now?

**errorHasOccurred()** - Method in class jminusminus.Parser

Has a parser error occurred up to now?

**errorHasOccurred()** - Method in class jminusminus.Scanner

Has an error occurred up to now in lexical analysis?

**ESC** - Static variable in interface jminusminus.JavaCCParserConstants

**escapeSpecialXMLChars(String)** - Static method in class jminusminus.Util

Escape the special XML characters in the specified string and return the escaped string.

**exceptionIndexTable** - Variable in class jminusminus.CLEExceptionsAttribute

Exceptions\_attribute.exception\_index\_table item.

**exceptionTable** - Variable in class jminusminus.CLCodeAttribute

Code\_attribute.exception\_table item.

**exceptionTableLength** - Variable in class jminusminus.CLCodeAttribute

Code\_attribute.exception\_table\_length item.

**exit()** - Static method in class spim.SPIM

Wrapper for SPIM.exit().

**exit2(int)** - Static method in class spim.SPIM

Wrapper for SPIM.exit2().

**ExpandBuff(boolean)** - Method in class jminusminus.SimpleCharStream

**expectedTokenSequences** - Variable in exception jminusminus.ParseException

Each entry in this array is an array of integers.

**expr** - Variable in class jminusminus.JStatementExpression

The expression.

**EXTENDS** - Static variable in interface jminusminus.JavaCCParserConstants

## F

**f** - Variable in class jminusminus.CLConstantFloatInfo

The floating point number.

**F2D** - Static variable in class jminusminus.CLConstants

F2D instruction.

**F2I** - Static variable in class jminusminus.CLConstants

F2I instruction.

**F2L** - Static variable in class jminusminus.CLConstants

F2L instruction.

**FADD** - Static variable in class jminusminus.CLConstants

FADD instruction.

**FALOAD** - Static variable in class jminusminus.CLConstants

FALOAD instruction.

**FALSE** - Static variable in interface jminusminus.JavaCCParserConstants

**FASTORE** - Static variable in class jminusminus.CLConstants

FASTORE instruction.

**FCMPG** - Static variable in class jminusminus.CLConstants

FCMPG instruction.

**FCMPL** - Static variable in class jminusminus.CLConstants

FCMPL instruction.

**FCONST\_0** - Static variable in class jminusminus.CLConstants

FCONST\_0 instruction.

**FCONST\_1** - Static variable in class jminusminus.CLConstants

FCONST\_1 instruction.

**FCONST\_2** - Static variable in class jminusminus.CLConstants

FCONST\_2 instruction.

**FDIV** - Static variable in class jminusminus.CLConstants

FDIV instruction.

**Field** - Class in jminusminus

A Field knows its type.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**Field(Field)** - Constructor for class jminusminus.Field

Construct a Field is constructed from its internal representation.

**fieldAccessFlagsToString(int)** - Static method in class jminusminus.CLFile

Return (as a string) the field access permissions and properties contained in the specified mask of flags.

**fieldFor(String)** - Method in class jminusminus.Type

Return the Field having this name.

**fields** - Variable in class jminusminus.CLFile

ClassFile.fields item.

**fieldsCount** - Variable in class jminusminus.CLFile

ClassFile.fields\_count item.

**fileName()** - Method in class jminusminus.CharReader

Return the file name.

**fileName(String)** - Method in class jminusminus.JavaCCParser

Set the name of the file that is being parsed.

**fileName()** - Method in class jminusminus.LookaheadScanner

Return the name of the source file.

**fileName()** - Method in class jminusminus.Scanner

The name of the source file.

**FillBuff()** - Method in class jminusminus.SimpleCharStream

**find(CLCPInfo)** - Method in class jminusminus.CLConstantPool

Return the constant pool index of the specified item if it exists in the pool, -1 otherwise.

**first** - Variable in class jminusminus.JavaCCParser.JJCalls

**firstRangeStart()** - Method in class jminusminus.NInterval

The start position for the first range.

**firstUsage()** - Method in class jminusminus.NInterval

The first use position in this interval.

**FLOAD** - Static variable in class jminusminus.CLConstants

FLOAD instruction.

**FLOAD\_0** - Static variable in class jminusminus.CLConstants

FLOAD\_0 instruction.

**FLOAD\_1** - Static variable in class jminusminus.CLConstants

FLOAD\_1 instruction.

**FLOAD\_2** - Static variable in class jminusminus.CLConstants

FLOAD\_2 instruction.

**FLOAD\_3** - Static variable in class jminusminus.CLConstants

FLOAD\_3 instruction.

**FMUL** - Static variable in class jminusminus.CLConstants

FMUL instruction.

**FNEG** - Static variable in class jminusminus.CLConstants

FNEG instruction.

**FP** - Static variable in class jminusminus.NPhysicalRegister

Frame pointer.

**FREM** - Static variable in class jminusminus.CLConstants

FREM instruction.

**FRETURN** - Static variable in class jminusminus.CLConstants

FRETURN instruction.

**FSTORE** - Static variable in class jminusminus.CLConstants

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



FSTORE instruction.

**FSTORE\_0** - Static variable in class jminusminus.CLConstants

FSTORE\_0 instruction.

**FSTORE\_1** - Static variable in class jminusminus.CLConstants

FSTORE\_1 instruction.

**FSTORE\_2** - Static variable in class jminusminus.CLConstants

FSTORE\_2 instruction.

**FSTORE\_3** - Static variable in class jminusminus.CLConstants

FSTORE\_3 instruction.

**FSUB** - Static variable in class jminusminus.CLConstants

FSUB instruction.

**fwdBranches** - Variable in class jminusminus.NBasicBlock

Number of forward branches to this block.

## G

**gen** - Variable in class jminusminus.JavaCCParser.JJCalls

**generateParseException()** - Method in class jminusminus.JavaCCParser

**get(Type, Type)** - Method in class jminusminus.Conversions

Retrieve a converter for converting from some original type to a target type; the converter may be empty (requiring no code for run-time execution).

**getBeginColumn()** - Method in class jminusminus.SimpleCharStream

**getBeginLine()** - Method in class jminusminus.SimpleCharStream

**getColumn()** - Method in class jminusminus.SimpleCharStream

Deprecated.

**getEndColumn()** - Method in class jminusminus.SimpleCharStream

**getEndLine()** - Method in class jminusminus.SimpleCharStream

**GETFIELD** - Static variable in class jminusminus.CLConstants

GETFIELD instruction.

**getFirstLIRInstId()** - Method in class jminusminus.NBasicBlock

The instruction identifier for the first LIR instruction.

**GetImage()** - Method in class jminusminus.SimpleCharStream

**getInstruction(int)** - Method in class jminusminus.NBasicBlock

Iterates through the lir array of this block, returning an NLIRInstruction with the specified id.

**getLastLIRInstId()** - Method in class jminusminus.NBasicBlock

The instruction identifier for the last LIR instruction.

**getLine()** - Method in class jminusminus.SimpleCharStream

Deprecated.

**getMessage()** - Method in exception jminusminus.ParseException

This method has the standard behavior when this object has been created using the standard constructors.

**getMessage()** - Method in error jminusminus.TokenMgrError

You can also modify the body of this method to customize your error messages.

**getNextToken()** - Method in class jminusminus.JavaCCParser

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**getNextToken()** - Method in class jminusminus.JavaCCParserTokenManager

**getNextToken()** - Method in class jminusminus.Scanner

Scan the next token from input.

**GETSTATIC** - Static variable in class jminusminus.CLConstants

GETSTATIC instruction.

**GetSuffix(int)** - Method in class jminusminus.SimpleCharStream

**getTabSize(int)** - Method in class jminusminus.SimpleCharStream

**getToken(int)** - Method in class jminusminus.JavaCCParser

**GOTO** - Static variable in class jminusminus.CLConstants

GOTO instruction.

**GOTO\_W** - Static variable in class jminusminus.CLConstants

GOTO\_W instruction.

**GP** - Static variable in class jminusminus.NPhysicalRegister

Pointer to global area.

**GT** - Static variable in interface jminusminus.JavaCCParserConstants

## H

**handlerLabel** - Variable in class jminusminus.CLEntryHandler

Instruction after this label is first instruction of the handler.

**handlerPC** - Variable in class jminusminus.CLEntryHandler

handlerLabel is resolved to this value.

**handlerPC** - Variable in class jminusminus.CLEntryHandler

exception\_table\_entry.handler\_pc item.

**highBytes()** - Method in class jminusminus.CLConstantDoubleInfo

Return CONSTANT\_Double\_info.high\_bytes item.

**highBytes()** - Method in class jminusminus.CLConstantLongInfo

Return CONSTANT\_Long\_info.high\_bytes item.

**hir** - Variable in class jminusminus.NBasicBlock

List of high-level (HIR) instructions in this block.

**hirId** - Static variable in class jminusminus.NControlFlowGraph

HIR instruction identifier.

**hirMap** - Variable in class jminusminus.NControlFlowGraph

Maps HIR instruction ids in this cfg to HIR instructions.

**hirMnemonic** - Static variable in class jminusminus.NHIRInstruction

Maps JVM opcode to a string mnemonic for HIR instructions.

**hirToLir()** - Method in class jminusminus.NControlFlowGraph

Convert the hir instructions in this cfg to lir instructions.

## I

**i** - Variable in class jminusminus.CLConstantIntegerInfo

The int number.

**I2B** - Static variable in class jminusminus.CLConstants

I2B instruction.

**I2C** - Static variable in class jminusminus.CLConstants

I2C instruction.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**I2C** - Class in [jminusminus](#)

Converting from an int to a char requires an I2C instruction.

**I2C()** - Constructor for class [jminusminus.I2C](#)

**I2D** - Static variable in class [jminusminus.CLConstants](#)  
I2D instruction.

**I2F** - Static variable in class [jminusminus.CLConstants](#)  
I2F instruction.

**I2L** - Static variable in class [jminusminus.CLConstants](#)  
I2L instruction.

**I2S** - Static variable in class [jminusminus.CLConstants](#)  
I2S instruction.

**IADD** - Static variable in class [jminusminus.CLConstants](#)  
IADD instruction.

**IALOAD** - Static variable in class [jminusminus.CLConstants](#)  
IALOAD instruction.

**IAND** - Static variable in class [jminusminus.CLConstants](#)  
IAND instruction.

**IASTORE** - Static variable in class [jminusminus.CLConstants](#)  
IASTORE instruction.

**ICONST\_0** - Static variable in class [jminusminus.CLConstants](#)  
ICONST\_0 instruction.

**ICONST\_1** - Static variable in class [jminusminus.CLConstants](#)  
ICONST\_1 instruction.

**ICONST\_2** - Static variable in class [jminusminus.CLConstants](#)  
ICONST\_2 instruction.

**ICONST\_3** - Static variable in class [jminusminus.CLConstants](#)  
ICONST\_3 instruction.

**ICONST\_4** - Static variable in class [jminusminus.CLConstants](#)  
ICONST\_4 instruction.

**ICONST\_5** - Static variable in class [jminusminus.CLConstants](#)  
ICONST\_5 instruction.

**ICONST\_M1** - Static variable in class [jminusminus.CLConstants](#)  
ICONST\_M1 instruction.

**id** - Variable in class [jminusminus.NBasicBlock](#)  
Unique identifier of this block.

**id()** - Method in class [jminusminus.NBasicBlock](#)  
Return a string identifier of this block.

**id** - Variable in class [jminusminus.NHIRInstruction](#)  
Unique identifier of this instruction.

**id()** - Method in class [jminusminus.NHIRInstruction](#)  
Return the identifier of this instruction with the short type name prefixed.

**id** - Variable in class [jminusminus.NLIRInstruction](#)  
Unique identifier of this instruction.

**IDefn** - Interface in [jminusminus](#)

The IDefn type is used to implement definitions of those things (local variables, formal arguments, types) that are named in some context (or scope).

**iDefn()** - Method in class [jminusminus.JVariable](#)  
Return the identifier's definition.

**IDENTIFIER** - Static variable in interface [jminusminus.JavaCCParserConstants](#)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**Identity** - Static variable in interface jminusminus.Converter

For identity conversion (no run-time code needed).

**Identity** - Class in jminusminus

The identity conversion requires no run-time code.

**Identity()** - Constructor for class jminusminus.Identity

**idsFree(int)** - Method in class jminusminus.NBasicBlock

Checks to see if there is an LIRInstruction with this id in the block's lir.

**IDIV** - Static variable in class jminusminus.CLConstants

IDIV instruction.

**IF** - Static variable in interface jminusminus.JavaCCParserConstants

**IF\_ACMPEQ** - Static variable in class jminusminus.CLConstants

IF\_ACMPEQ instruction.

**IF\_ACMUNE** - Static variable in class jminusminus.CLConstants

IF\_ACMUNE instruction.

**IF\_ICMPEQ** - Static variable in class jminusminus.CLConstants

IF\_ICMPEQ instruction.

**IF\_ICMPGE** - Static variable in class jminusminus.CLConstants

IF\_ICMPGE instruction.

**IF\_ICMPGT** - Static variable in class jminusminus.CLConstants

IF\_ICMPGT instruction.

**IF\_ICMPLE** - Static variable in class jminusminus.CLConstants

IF\_ICMPLE instruction.

**IF\_ICMPLT** - Static variable in class jminusminus.CLConstants

IF\_ICMPLT instruction.

**IF\_ICMPNE** - Static variable in class jminusminus.CLConstants

IF\_ICMPNE instruction.

**IFEQ** - Static variable in class jminusminus.CLConstants

IFEQ instruction.

**IFGE** - Static variable in class jminusminus.CLConstants

IFGE instruction.

**IFGT** - Static variable in class jminusminus.CLConstants

IFGT instruction.

**IFLE** - Static variable in class jminusminus.CLConstants

IFLE instruction.

**IFLT** - Static variable in class jminusminus.CLConstants

IFLT instruction.

**IFNE** - Static variable in class jminusminus.CLConstants

IFNE instruction.

**IFNONNULL** - Static variable in class jminusminus.CLConstants

IFNONNULL instruction.

**IFNULL** - Static variable in class jminusminus.CLConstants

IFNULL instruction.

**IINC** - Static variable in class jminusminus.CLConstants

IINC instruction.

**ILOAD** - Static variable in class jminusminus.CLConstants

ILOAD instruction.

**ILOAD\_0** - Static variable in class jminusminus.CLConstants

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

ILOAD\_0 instruction.

**ILOAD\_1** - Static variable in class jminusminus.CLConstants

ILOAD\_1 instruction.

**ILOAD\_2** - Static variable in class jminusminus.CLConstants

ILOAD\_2 instruction.

**ILOAD\_3** - Static variable in class jminusminus.CLConstants

ILOAD\_3 instruction.

**image** - Variable in class jminusminus.Token

The string image of the token.

**image()** - Method in class jminusminus.TokenInfo

Return the semantic text associated with the token.

**image()** - Method in enum jminusminus.TokenKind

Return the image of the token.

**IMPORT** - Static variable in interface jminusminus.JavaCCParserConstants

**IMUL** - Static variable in class jminusminus.CLConstants

IMUL instruction.

**inBuf** - Variable in class jminusminus.SimpleCharStream

**INC** - Static variable in interface jminusminus.JavaCCParserConstants

**indentLeft()** - Method in class jminusminus.PrettyPrinter

Indent left.

**indentRight()** - Method in class jminusminus.PrettyPrinter

Indent right.

**index** - Variable in class jminusminus.CLLocalVariableInfo

local\_variable\_table\_entry.index item.

**index** - Variable in class jminusminus.CLLocalVariableTypeInfo

local\_variable\_type\_table\_entry.index item.

**index** - Variable in class jminusminus.NHIRALoad

HIR id of the array index.

**index** - Variable in class jminusminus.NHIRASTore

HIR id of the array index.

**INEG** - Static variable in class jminusminus.CLConstants

INEG instruction.

**inferType()** - Method in class jminusminus.NHIRPhiFunction

Infer type for this phi function.

**initialize()** - Method in class jminusminus.LocalVariableDefn

Initialize this local variable.

**initializeByteClassLoader()** - Static method in class jminusminus.CLEmitter

Set a new ByteClassLoader for loading classes from byte streams.

**initializer()** - Method in class jminusminus.JVariableDeclarator

Return the variable initializer.

**innerClassAccessFlags** - Variable in class jminusminus.CLIInnerClassInfo

classes\_table\_entry.inner\_class\_access\_flags item.

**innerClassAccessFlagsToString(int)** - Static method in class jminusminus.CLFile

Return (as a string) the inner class access permissions and properties contained in the specified mask of flags.

**innerClassInfoIndex** - Variable in class jminusminus.CLIInnerClassInfo

classes\_table\_entry.inner\_class\_info\_index item.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



**innerNameIndex** - Variable in class jminusminus.CLIInnerClassInfo  
classes\_table\_entry.inner\_name\_index item.

**input\_stream** - Variable in class jminusminus.JavaCCParserTokenManager

**inputStream** - Variable in class jminusminus.SimpleCharStream

**insertLIRInst(NLIRInstruction)** - Method in class jminusminus.NBasicBlock

Inserts an NLIRInstruction to the appropriate place in this block's lir array based on its id -- preserving order by id.

**instanceFieldInitializations()** - Method in class jminusminus.JClassDeclaration

The initializations for instance fields (now expressed as assignment statments).

**INSTANCEOF** - Static variable in class jminusminus.CLConstants

INSTANCEOF instruction.

**INSTANCEOF** - Static variable in interface jminusminus.JavaCCParserConstants

**instructionInfo** - Static variable in class jminusminus.CLInstruction

For each JVM instruction, this array stores its opcode, mnemonic, number of operands (DYNAMIC for instructions with variable attribute count), local variable index (IRRELEVANT where not applicable), stack units, and instruction category.

**InstructionType** - Enum in jminusminus

The types of possible uses.

**INT** - Static variable in interface jminusminus.JavaCCParserConstants

**INT** - Static variable in class jminusminus.Type

The primitive type, int.

**INT\_LITERAL** - Static variable in interface jminusminus.JavaCCParserConstants

**interfaces** - Variable in class jminusminus.CLFile

ClassFile.interfaces item.

**interfacesCount** - Variable in class jminusminus.CLFile

ClassFile.interfaces\_count item.

**intervals** - Variable in class jminusminus.NControlFlowGraph

Intervals allocated by the register allocation algorithm.

**INVALID\_LEXICAL\_STATE** - Static variable in error jminusminus.TokenMgrError

Tried to change to an invalid lexical state.

**INVOKEDYNAMIC** - Static variable in class jminusminus.CLConstants

INVOKEDYNAMIC instruction.

**INVOKEINTERFACE** - Static variable in class jminusminus.CLConstants

INVOKEINTERFACE instruction.

**INVOKESPECIAL** - Static variable in class jminusminus.CLConstants

INVOKESPECIAL instruction.

**INVOKESTATIC** - Static variable in class jminusminus.CLConstants

INVOKESTATIC instruction.

**INVOKEVIRTUAL** - Static variable in class jminusminus.CLConstants

INVOKEVIRTUAL instruction.

**IOR** - Static variable in class jminusminus.CLConstants

IOR instruction.

**IREM** - Static variable in class jminusminus.CLConstants

IREM instruction.

**IRETURN** - Static variable in class jminusminus.CLConstants

IRETURN instruction.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**IRRELEVANT** - Static variable in class jminusminus.CLConstants

Denotes values that are irrelevant to certain instructions.

**isAbstract** - Variable in class jminusminus.JMethodDeclaration

Is method abstract.

**isAbstract()** - Method in class jminusminus.Member

Has this member been declared with the abstract modifier?

**isAbstract()** - Method in class jminusminus.Type

Is this type declared abstract?

**isArray()** - Method in class jminusminus.Type

Is this an Array type?

**isChild()** - Method in class jminusminus.NInterval

Is this interval a child interval?

**isFinal()** - Method in class jminusminus.Member

Has this member been declared with the final modifier? Note, we cannot declare anything final as it is not in our lexicon.

**isFinal()** - Method in class jminusminus.Type

Is this type declared final?

**ISHL** - Static variable in class jminusminus.CLConstants

ISHL instruction.

**ISHR** - Static variable in class jminusminus.CLConstants

ISHR instruction.

**isInitialized** - Variable in class jminusminus.JMethodDeclaration

**isInitialized()** - Method in class jminusminus.LocalVariableDefn

Has this local variable been initialized?

**isInterface()** - Method in class jminusminus.Type

Is this an interface type?

**isJavaAssignableFrom(Type)** - Method in class jminusminus.Type

Is this a supertype of that?

**isLeader** - Variable in class jminusminus.NTuple

Is this tuple the leader of the block containing it.

**isLiveAt(int)** - Method in class jminusminus.NInterval

Check if this vreg is alive at a given index.

**isLookingAhead** - Variable in class jminusminus.LookaheadScanner

Whether we are looking ahead.

**isLoopHead** - Variable in class jminusminus.NBasicBlock

Is this block a loop head?

**isLoopTail** - Variable in class jminusminus.NBasicBlock

Is this block a loop tail?

**isParent()** - Method in class jminusminus.NInterval

Is this interval a parent interval? (I.e., does it have children?)

**isPrimitive()** - Method in class jminusminus.Type

Is this a primitive type?

**isPrivate** - Variable in class jminusminus.JMethodDeclaration

Is method private.

**isPrivate()** - Method in class jminusminus.Member

Has this member been declared with the private modifier?

**isProtected()** - Method in class jminusminus.Member

Has this member been declared with the protected modifier?

**isPublic()** - Method in class jminusminus.Member

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Has this member been declared with the public modifier?

**isReference()** - Method in class `jminusminus.Type`

Is this a reference type?

**isStatementExpression** - Variable in class `jminusminus.JExpression`

Whether or not this expression is a statement.

**isStatementExpression()** - Method in class `jminusminus.JExpression`

Is this a statementExpression?

**isStatic** - Variable in class `jminusminus.JMethodDeclaration`

Is method static.

**isStatic()** - Method in class `jminusminus.Member`

Has this member been declared with the static modifier?

**isStatic()** - Method in class `jminusminus.MethodContext`

Is this method static?

**ISTORE** - Static variable in class `jminusminus.CLConstants`

ISTORE instruction.

**ISTORE\_0** - Static variable in class `jminusminus.CLConstants`

ISTORE\_0 instruction.

**ISTORE\_1** - Static variable in class `jminusminus.CLConstants`

ISTORE\_1 instruction.

**ISTORE\_2** - Static variable in class `jminusminus.CLConstants`

ISTORE\_2 instruction.

**ISTORE\_3** - Static variable in class `jminusminus.CLConstants`

ISTORE\_3 instruction.

**ISUB** - Static variable in class `jminusminus.CLConstants`

ISUB instruction.

**isValid(int)** - Static method in class `jminusminus.CLInstruction`

Return true if the opcode is valid; false otherwise.

**IUSHR** - Static variable in class `jminusminus.CLConstants`

IUSHR instruction.

**IXOR** - Static variable in class `jminusminus.CLConstants`

IXOR instruction.

## J

**JArrayExpression** - Class in `jminusminus`

The AST for an array indexing operation.

**JArrayExpression(int, JExpression, JExpression)** - Constructor for class `jminusminus.JArrayExpression`

Construct an AST node for an array indexing operation.

**JArrayInitializer** - Class in `jminusminus`

The AST node for an array initializer.

**JArrayInitializer(int, Type, ArrayList<JExpression>)** - Constructor for class `jminusminus.JArrayInitializer`

Construct an AST node for an array initializer given the (expected) array type and initial values.

**JAssignment** - Class in `jminusminus`

The AST node for an assignment statement.

**JAssignment(int, String, JExpression, JExpression)** - Constructor for class `jminusminus.JAssignment`

Construct an AST node for an assignment operation.

**JAssignOp** - Class in `jminusminus`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

The AST node for an assignment (=) expression.

**JAssignOp(int, JExpression, JExpression)** - Constructor for class jminusminus.JAssignOp  
Construct the AST node for an assignment (=) expression given the lhs and rhs operands.

**JAST** - Class in jminusminus

The abstract superclass of all nodes in the abstract syntax tree (AST).

**JAST(int)** - Constructor for class jminusminus.JAST

Construct an AST node the given its line number in the source file.

**JavaCCMain** - Class in jminusminus

Driver class for j-- compiler using JavaCC front-end.

**JavaCCMain()** - Constructor for class jminusminus.JavaCCMain

**JavaCCParser** - Class in jminusminus

Parser generated by JavaCC.

**JavaCCParser(InputStream)** - Constructor for class jminusminus.JavaCCParser

**JavaCCParser(InputStream, String)** - Constructor for class jminusminus.JavaCCParser

**JavaCCParser(Reader)** - Constructor for class jminusminus.JavaCCParser

**JavaCCParser(JavaCCParserTokenManager)** - Constructor for class jminusminus.JavaCCParser

**JavaCCParser.JJCalls** - Class in jminusminus

**JavaCCParser.JJCalls()** - Constructor for class jminusminus.JavaCCParser.JJCalls

**JavaCCParserConstants** - Interface in jminusminus

**JavaCCParserTokenManager** - Class in jminusminus

**JavaCCParserTokenManager(SimpleCharStream)** - Constructor for class jminusminus.JavaCCParserTokenManager

**JavaCCParserTokenManager(SimpleCharStream, int)** - Constructor for class jminusminus.JavaCCParserTokenManager

**JBinaryExpression** - Class in jminusminus

The AST node for a binary expression.

**JBinaryExpression(int, String, JExpression, JExpression)** - Constructor for class jminusminus.JBinaryExpression

Construct an AST node for a binary expression given its line number, the binary operator, and lhs and rhs operands.

**JBlock** - Class in jminusminus

The AST node for a block, which delimits a nested level of scope.

**JBlock(int, ArrayList<JStatement>)** - Constructor for class jminusminus.JBlock

Construct an AST node for a block given its line number, and the list of statements forming the block body.

**JBooleanBinaryExpression** - Class in jminusminus

Most binary expressions that return booleans can be recognized by their syntax.

**JBooleanBinaryExpression(int, String, JExpression, JExpression)** - Constructor for class jminusminus.JBooleanBinaryExpression

Construct an AST node for a boolean binary expression.

**JCastOp** - Class in jminusminus

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

The AST for an cast expression, which has both a cast (a type) and the expression to be cast.

**JCastOp(int, Type, JExpression)** - Constructor for class `jminusminus.JCastOp`

Construct an AST node for a cast operation from its line number, cast, and expression.

**JClassDeclaration** - Class in `jminusminus`

A class declaration has a list of modifiers, a name, a super class and a class block; it distinguishes between instance fields and static (class) fields for initialization, and it defines a type.

**JClassDeclaration(int, ArrayList<String>, String, Type, ArrayList<JMember>)** - Constructor for class `jminusminus.JClassDeclaration`

Construct an AST node for a class declaration given the line number, list of class modifiers, name of the class, its super class type, and the class block.

**JComparison** - Class in `jminusminus`

The AST node for a comparison expression.

**JComparison(int, String, JExpression, JExpression)** - Constructor for class `jminusminus.JComparison`

Create an AST node for a comparison expression.

**JCompilationUnit** - Class in `jminusminus`

The abstract syntax tree (AST) node representing a compilation unit, and so the root of the AST.

**JCompilationUnit(String, int, TypeName, ArrayList<TypeName>, ArrayList<JAST>)** - Constructor for class `jminusminus.JCompilationUnit`

Construct an AST node for a compilation unit given a file name, class directory, line number, package name, list of imports, and type declarations.

**JConstructorDeclaration** - Class in `jminusminus`

The AST node for a constructor declaration.

**JConstructorDeclaration(int, ArrayList<String>, String, ArrayList<JFormalParameter>, JBlock)** - Constructor for class `jminusminus.JConstructorDeclaration`

Construct an AST node for a constructor declaration given the line number, modifiers, constructor name, formal parameters, and the constructor body.

**JEmptyStatement** - Class in `jminusminus`

The (dummy) AST node for representing the empty statement.

**JEmptyStatement(int)** - Constructor for class `jminusminus.JEmptyStatement`

Construct an AST node for an empty statement.

**JEqualOp** - Class in `jminusminus`

The AST node for an equality (==) expression.

**JEqualOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JEqualOp`

Construct an AST node for an equality expression.

**JExpression** - Class in `jminusminus`

The AST node for an expression.

**JExpression(int)** - Constructor for class `jminusminus.JExpression`

Construct an AST node for an expression given its line number.

**JFieldDeclaration** - Class in `jminusminus`

The AST node for a field declaration.

**JFieldDeclaration(int, ArrayList<String>, ArrayList<JVariableDeclarator>)** - Constructor for class `jminusminus.JFieldDeclaration`

Construct an AST node for a field declaration given the line number, modifiers, and the variable declarators.

**JFieldSelection** - Class in `jminusminus`

The AST node for a field selection operation.

**JFieldSelection(int, JExpression, String)** - Constructor for class `jminusminus.JFieldSelection`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Construct an AST node for a field selection without an ambiguous part.

**JFieldSelection(int, AmbiguousName, JExpression, String)** - Constructor for class `jminusminus.JFieldSelection`

Construct an AST node for a field selection having an ambiguous part.

**JFormalParameter** - Class in `jminusminus`

The AST node for a formal parameter declaration.

**JFormalParameter(int, String, Type)** - Constructor for class `jminusminus.JFormalParameter`

Construct an AST node for a formal parameter declaration given its line number, name, and type.

**JGreaterThanOp** - Class in `jminusminus`

The AST node for a greater-than (>) expression.

**JGreaterThanOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JGreaterThanOp`

Construct an AST node for a greater-than expression given its line number, and the lhs and rhs operands.

**JIfStatement** - Class in `jminusminus`

The AST node for an if-statement.

**JIfStatement(int, JExpression, JStatement, JStatement)** - Constructor for class `jminusminus.JIfStatement`

Construct an AST node for an if-statement given its line number, the test expression, the consequent, and the alternate.

**JInstanceOfOp** - Class in `jminusminus`

The AST node for an instanceof expression, having two arguments: an expression and a reference type.

**JInstanceOfOp(int, JExpression, Type)** - Constructor for class `jminusminus.JInstanceOfOp`

Construct an AST node for an instanceof expression given its line number, the relational expression and reference type.

**jj\_input\_stream** - Variable in class `jminusminus.JavaCCParser`

**jj\_nt** - Variable in class `jminusminus.JavaCCParser`

**jjbitVec0** - Static variable in class `jminusminus.JavaCCParserTokenManager`

**jjFillToken()** - Method in class `jminusminus.JavaCCParserTokenManager`

**jjmatchedKind** - Variable in class `jminusminus.JavaCCParserTokenManager`

**jjmatchedPos** - Variable in class `jminusminus.JavaCCParserTokenManager`

**jjnewStateCnt** - Variable in class `jminusminus.JavaCCParserTokenManager`

**jjnextStates** - Static variable in class `jminusminus.JavaCCParserTokenManager`

**jjround** - Variable in class `jminusminus.JavaCCParserTokenManager`

**jjstrLiteralImages** - Static variable in class `jminusminus.JavaCCParserTokenManager`

**jjtoSkip** - Static variable in class `jminusminus.JavaCCParserTokenManager`

**jjtoToken** - Static variable in class `jminusminus.JavaCCParserTokenManager`

**JLessEqualOp** - Class in `jminusminus`

The AST node for a less-than-or-equal-to (<=) expression.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**JLessEqualOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JLessEqualOp`

Construct an AST node for a less-than-or-equal-to expression given its line number, and the lhs and rhs operands.

**JLhs** - Interface in `jminusminus`

The type of any expression that can appear on the lhs of an assignment statement, i.e., `JVariable`, `JFieldSelection`, `JArrayExpression`.

**JLiteralChar** - Class in `jminusminus`

The AST node for a char literal.

**JLiteralChar(int, String)** - Constructor for class `jminusminus.JLiteralChar`

Construct an AST node for a char literal given its line number and text representation.

**JLiteralFalse** - Class in `jminusminus`

The AST node for the boolean "false" literal.

**JLiteralFalse(int)** - Constructor for class `jminusminus.JLiteralFalse`

Construct an AST node for a "false" literal given its line number.

**JLiteralInt** - Class in `jminusminus`

The AST node for an int literal.

**JLiteralInt(int, String)** - Constructor for class `jminusminus.JLiteralInt`

Construct an AST node for an int literal given its line number and string representation.

**JLiteralNull** - Class in `jminusminus`

The AST node for the null literal.

**JLiteralNull(int)** - Constructor for class `jminusminus.JLiteralNull`

Construct an AST node for the null literal given its line number.

**JLiteralString** - Class in `jminusminus`

The AST node for a string literal.

**JLiteralString(int, String)** - Constructor for class `jminusminus.JLiteralString`

Construct an AST node for a string literal given its line number and string representation.

**JLiteralTrue** - Class in `jminusminus`

The AST node for the boolean "true" literal.

**JLiteralTrue(int)** - Constructor for class `jminusminus.JLiteralTrue`

Construct an AST node for a "true" literal given its line number.

**JLogicalAndOp** - Class in `jminusminus`

The AST node for a logical AND (&&) expression.

**JLogicalAndOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JLogicalAndOp`

Construct an AST node for a logical AND expression given its line number, and lhs and rhs operands.

**JLogicalNotOp** - Class in `jminusminus`

The AST node for a logical NOT (!) expression.

**JLogicalNotOp(int, JExpression)** - Constructor for class `jminusminus.JLogicalNotOp`

Construct an AST for a logical NOT expression given its line number, and the operand.

**JMember** - Interface in `jminusminus`

An interface supported by all class (or later, interface) members.

**JMessageExpression** - Class in `jminusminus`

The AST node for a message expression that has a target, optionally an ambiguous part, a message name, and zero or more actual arguments.

**JMessageExpression(int, JExpression, String, ArrayList<JExpression>)** - Constructor for class `jminusminus.JMessageExpression`

Construct an AST node for a message expression without an ambiguous part.

**JMessageExpression(int, JExpression, AmbiguousName, String,**

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**ArrayList<JExpression>)** - Constructor for class `jminusminus.JMessageExpression`

Construct an AST node for a message expression having an ambiguous part.

**JMethodDeclaration** - Class in `jminusminus`

The AST node for a method declaration.

**JMethodDeclaration(int, ArrayList<String>, String, Type, ArrayList<JFormalParameter>, JBlock)** - Constructor for class `jminusminus.JMethodDeclaration`

Construct an AST node for a method declaration given the line number, method name, return type, formal parameters, and the method body.

`jminusminus` - package `jminusminus`

This package contains the scanner, parser, AST node representations, bytecode emitter, JavaCC description file, and driver classes for the j-- compiler.

**JMultiplyOp** - Class in `jminusminus`

The AST node for a multiplication (\*) expression.

**JMultiplyOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JMultiplyOp`

Construct an AST for a multiplication expression given its line number, and the lhs and rhs operands.

**JNegateOp** - Class in `jminusminus`

The AST node for a unary negation (-) expression.

**JNegateOp(int, JExpression)** - Constructor for class `jminusminus.JNegateOp`

Construct an AST node for a negation expression given its line number, and the operand.

**JNewArrayOp** - Class in `jminusminus`

The AST node for a "new" array operation.

**JNewArrayOp(int, Type, ArrayList<JExpression>)** - Constructor for class `jminusminus.JNewArrayOp`

Construct an AST node for a "new" array operation.

**JNewOp** - Class in `jminusminus`

The AST node for a "new" expression.

**JNewOp(int, Type, ArrayList<JExpression>)** - Constructor for class `jminusminus.JNewOp`

Construct an AST node for a "new" expression.

**JPlusAssignOp** - Class in `jminusminus`

The AST node for a += expression.

**JPlusAssignOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JPlusAssignOp`

Construct the AST node for a += expression given its lhs and rhs operands.

**JPlusOp** - Class in `jminusminus`

The AST node for a plus (+) expression.

**JPlusOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JPlusOp`

Construct an AST node for an addition expression given its line number, and the lhs and rhs operands.

**JPostDecrementOp** - Class in `jminusminus`

The AST node for an expr--.

**JPostDecrementOp(int, JExpression)** - Constructor for class `jminusminus.JPostDecrementOp`

Construct an AST node for an expr-- expression given its line number, and the operand.

**JPreIncrementOp** - Class in `jminusminus`

The AST node for a ++expr expression.

**JPreIncrementOp(int, JExpression)** - Constructor for class `jminusminus.JPreIncrementOp`

Construct an AST node for a ++expr given its line number, and the operand.

**JReturnStatement** - Class in `jminusminus`

The AST node for a return-statement.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**JReturnStatement(int, JExpression)** - Constructor for class `jminusminus.JReturnStatement`  
Construct an AST node for a return-statement given its line number, and the expression that is returned.

**JSR** - Static variable in class `jminusminus.CLConstants`  
JSR instruction.

**JSR\_W** - Static variable in class `jminusminus.CLConstants`  
JSR\_W instruction.

**JStatement** - Class in `jminusminus`  
The AST node for a statement (includes expressions).

**JStatement(int)** - Constructor for class `jminusminus.JStatement`  
Construct an AST node for a statement given its line number.

**JStatementExpression** - Class in `jminusminus`  
The AST node for an expression that appears as a statement.

**JStatementExpression(int, JExpression)** - Constructor for class `jminusminus.JStatementExpression`  
Construct an AST node for a statement expression given its line number, and expression.

**JStringConcatenationOp** - Class in `jminusminus`  
The AST node for a string concatenation operation.

**JStringConcatenationOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JStringConcatenationOp`  
Construct an AST node for a string concatenation expression given its line number, and the lhs and rhs operands.

**JSubtractOp** - Class in `jminusminus`  
The AST node for a subtraction (-) expression.

**JSubtractOp(int, JExpression, JExpression)** - Constructor for class `jminusminus.JSubtractOp`  
Construct an AST node for a subtraction expression given its line number, and lhs and rhs operands.

**JSuper** - Class in `jminusminus`  
The AST node for a "super" expression.

**JSuper(int)** - Constructor for class `jminusminus.JSuper`  
Construct an AST node for a "super" expression given its line number.

**JSuperConstruction** - Class in `jminusminus`  
The AST node for a super(...) constructor.

**JSuperConstruction(int, ArrayList<JExpression>)** - Constructor for class `jminusminus.JSuperConstruction`  
Construct an AST node for a super(...) constructor given its line number and arguments.

**JThis** - Class in `jminusminus`  
The AST for a "this" expression.

**JThis(int)** - Constructor for class `jminusminus.JThis`  
Construct an AST node for a "this" expression given its line number.

**JThisConstruction** - Class in `jminusminus`  
The AST node for a this(...) constructor.

**JThisConstruction(int, ArrayList<JExpression>)** - Constructor for class `jminusminus.JThisConstruction`  
Construct the AST node for a this(...) constructor given its line number and arguments.

**JTypeDecl** - Interface in `jminusminus`  
An interface supported by class (or later, interface) declarations.

**jumpToOffset()** - Method in class `jminusminus.CLFlowControlInstruction`  
Return the pc of instruction to jump to.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**JUnaryExpression** - Class in `jminusminus`

The AST node for a unary expression.

**JUnaryExpression(int, String, JExpression)** - Constructor for class `jminusminus.JUnaryExpression`

Construct an AST node for a unary expression given its line number, the unary operator, and the operand.

**JVariable** - Class in `jminusminus`

The AST node for an identifier used as a primary expression.

**JVariable(int, String)** - Constructor for class `jminusminus.JVariable`

Construct the AST node for a variable given its line number and name.

**JVariableDeclaration** - Class in `jminusminus`

The AST node for a local variable declaration.

**JVariableDeclaration(int, ArrayList<String>, ArrayList<JVariableDeclarator>)** - Constructor for class `jminusminus.JVariableDeclaration`

Construct an AST node for a variable declaration given the line number, modifiers, and the variable declarators.

**JVariableDeclarator** - Class in `jminusminus`

The AST node for a variable declarator, which declares a name, its type and (possibly) provides an initialization.

**JVariableDeclarator(int, String, Type, JExpression)** - Constructor for class `jminusminus.JVariableDeclarator`

Construct an AST node for a variable declarator given the line number, variable name, its type, and the initializer.

**jvmName()** - Method in class `jminusminus.Type`

The JVM representation for this type's name.

**jvmName()** - Method in class `jminusminus.TypeName`

Return the JVM name for this (identifier) type.

**JWhileStatement** - Class in `jminusminus`

The AST node for a while-statement.

**JWhileStatement(int, JExpression, JStatement)** - Constructor for class `jminusminus.JWhileStatement`

Construct an AST node for a while-statement given its line number, the test expression, and the body.

**JWildExpression** - Class in `jminusminus`

The AST node for a "wild" expression.

**JWildExpression(int)** - Constructor for class `jminusminus.JWildExpression`

Construct an AST node for a "wild" expression given its line number.

## K

**K0** - Static variable in class `jminusminus.NPhysicalRegister`

Reserved for OS kernel.

**K1** - Static variable in class `jminusminus.NPhysicalRegister`

Reserved for OS kernel.

**kind** - Variable in class `jminusminus.Token`

An integer that describes the kind of this token.

**kind()** - Method in class `jminusminus.TokenInfo`

Return the token's kind.

## L

**L** - Variable in class `jminusminus.CLConstantLongInfo`



The long number.

**L2D** - Static variable in class jminusminus.CLConstants

L2D instruction.

**L2F** - Static variable in class jminusminus.CLConstants

L2F instruction.

**L2I** - Static variable in class jminusminus.CLConstants

L2I instruction.

**labelPrefix** - Variable in class jminusminus.NControlFlowGraph

Used to construct jump labels in spim output.

**LADD** - Static variable in class jminusminus.CLConstants

LADD instruction.

**LALOAD** - Static variable in class jminusminus.CLConstants

LALOAD instruction.

**LAND** - Static variable in class jminusminus.CLConstants

LAND instruction.

**LAND** - Static variable in interface jminusminus.JavaCCParserConstants

**lastNRangeStop()** - Method in class jminusminus.NInterval

The stop position for the last range.

**LASTORE** - Static variable in class jminusminus.CLConstants

LASTORE instruction.

**LBRACK** - Static variable in interface jminusminus.JavaCCParserConstants

**LCMP** - Static variable in class jminusminus.CLConstants

LCMP instruction.

**LCONST\_0** - Static variable in class jminusminus.CLConstants

LCONST\_0 instruction.

**LCONST\_1** - Static variable in class jminusminus.CLConstants

LCONST\_1 instruction.

**LCURLY** - Static variable in interface jminusminus.JavaCCParserConstants

**LDC** - Static variable in class jminusminus.CLConstants

LDC instruction.

**LDC2\_W** - Static variable in class jminusminus.CLConstants

LDC2\_W instruction.

**LDC\_W** - Static variable in class jminusminus.CLConstants

LDC\_W instruction.

**LDIV** - Static variable in class jminusminus.CLConstants

LDIV instruction.

**LE** - Static variable in interface jminusminus.JavaCCParserConstants

**length()** - Method in class jminusminus.CLConstantUtf8Info

Return CONSTANT\_Utf8\_info.length item.

**length** - Variable in class jminusminus.CLLocalVariableInfo

local\_variable\_table\_entry.length item.

**length** - Variable in class jminusminus.CLLocalVariableTypeInfo

local\_variable\_type\_table\_entry.length item.

**LETTER** - Static variable in interface jminusminus.JavaCCParserConstants

**LEXICAL\_ERROR** - Static variable in error jminusminus.TokenMgrError

Lexical error occurred.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**LexicalError(boolean, int, int, int, String, char)** - Static method in error  
jminusminus.TokenMgrError

Returns a detailed message for the Error when it is thrown by the token manager to indicate a lexical error.

**lexStateNames** - Static variable in class jminusminus.JavaCCParserTokenManager

**lhs** - Variable in class jminusminus.JBinaryExpression  
The lhs operand.

**lhs** - Variable in class jminusminus.NHIRArithmetic  
Lhs HIR id.

**lhs** - Variable in class jminusminus.NHIRConditionalJump  
Lhs HIR id.

**line()** - Method in class jminusminus.CharReader  
The current line number in the source file, starting at 1.

**line** - Variable in class jminusminus.JAST  
Line in which the source for the AST was found.

**line()** - Method in class jminusminus.JAST  
Return the line in which the source for the AST was found.

**line** - Variable in class jminusminus.SimpleCharStream

**line()** - Method in class jminusminus.TokenInfo  
Return the line number associated with the token

**line()** - Method in class jminusminus.TypeName  
Return the line in which the identifier occurs in the source file.

**lineNumber** - Variable in class jminusminus.CLLineNumberInfo  
line\_number\_table\_entry.line\_number item

**lineNumberTable** - Variable in class jminusminus.CLLineNumberTableAttribute  
LineNumberTable\_attribute.line\_number\_table item.

**lineNumberTableLength** - Variable in class jminusminus.CLLineNumberTableAttribute  
LineNumberTable\_attribute.line\_number\_table\_length item.

**lir** - Variable in class jminusminus.NBasicBlock  
List of low-level (LIR) instructions in this block.

**lir** - Variable in class jminusminus.NHIRInstruction  
The LIR instruction corresponding to this HIR instruction.

**lirId** - Static variable in class jminusminus.NControlFlowGraph  
HIR instruction identifier.

**lirMnemonic** - Static variable in class jminusminus.NLIRInstruction  
Maps JVM opcode to a string mnemonic for LIR instructions.

**liveDef** - Variable in class jminusminus.NBasicBlock  
All virtual registers locally defined within this block.

**liveIn** - Variable in class jminusminus.NBasicBlock  
All virtual registers live in the block.

**liveOut** - Variable in class jminusminus.NBasicBlock  
All virtual registers live outside the block.

**liveUse** - Variable in class jminusminus.NBasicBlock  
All virtual registers used before definition within this block.

**LLOAD** - Static variable in class jminusminus.CLConstants  
LLOAD instruction.

**LLOAD\_0** - Static variable in class jminusminus.CLConstants  
LLOAD\_0 instruction.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**LLOAD\_1** - Static variable in class jminusminus.CLConstants

LLOAD\_1 instruction.

**LLOAD\_2** - Static variable in class jminusminus.CLConstants

LLOAD\_2 instruction.

**LLOAD\_3** - Static variable in class jminusminus.CLConstants

LLOAD\_3 instruction.

**LMUL** - Static variable in class jminusminus.CLConstants

LMUL instruction.

**LNEG** - Static variable in class jminusminus.CLConstants

LNEG instruction.

**LNOT** - Static variable in interface jminusminus.JavaCCParserConstants

**loadClass(String, boolean)** - Method in class jminusminus.ByteClassLoader

**loadClass(String)** - Method in class jminusminus.CLPath

Return a CLInputStream instance for the class with specified name (fully-qualified; tokens separated by '/') or null if the class was not found.

**local** - Variable in class jminusminus.NHIRLoadLocal

Local variable index.

**local** - Variable in class jminusminus.NHIRLocal

Local variable index.

**local** - Variable in class jminusminus.NHIRPhiFunction

Local variable index.

**local** - Variable in class jminusminus.NLIRLoadLocal

Local variable index.

**LocalContext** - Class in jminusminus

A local context is a context (scope) in which local variables (including formal parameters) can be declared.

**LocalContext(Context)** - Constructor for class jminusminus.LocalContext

Construct a local context.

**locals** - Variable in class jminusminus.NBasicBlock

The state array for this block that maps local variable index to the HIR instruction that last affected it.

**LocalVariableDefn** - Class in jminusminus

The definition for a local variable (including formal parameters).

**LocalVariableDefn(Type, int)** - Constructor for class jminusminus.LocalVariableDefn

Construct a local variable definition for a local variable.

**localVariableIndex** - Variable in class jminusminus.CLInsInfo

Index of the local variable that this instruction refers to; applies only to instructions that operate on local variables.

**localVariableIndex** - Variable in class jminusminus.CLInstruction

Index of the local variable that this instruction refers to; applies only to instructions that operate on local variables.

**localVariableIndex()** - Method in class jminusminus.CLInstruction

Return the local variable index for this instruction.

**localVariableTable** - Variable in class jminusminus.CLLocalVariableTableAttribute

LocalVariableTable\_attribute.local\_variable\_table item.

**localVariableTableLength** - Variable in class jminusminus.CLLocalVariableTableAttribute

LocalVariableTable\_attribute.local\_variable\_table\_length item.

**localVariableTypeTable** - Variable in class jminusminus.CLLocalVariableTypeTableAttribute

LocalVariableTypeTable\_attribute.local\_variable\_type\_table item.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**localVariableTypeTableLength** - Variable in class `jminusminus.CLLocalVariableTypeTableAttribute`

`LocalVariableTypeTable_attribute`.

**LookaheadScanner** - Class in `jminusminus`

A lexical analyzer for j-- that interfaces with the hand-written parser (`Parser.java`).

**LookaheadScanner(String)** - Constructor for class `jminusminus.LookaheadScanner`

Construct a `LookaheadScanner` from a file name.

**lookingAhead** - Variable in class `jminusminus.JavaCCParser`

**lookup(String)** - Method in class `jminusminus.Context`

Return the definition for a name in the environment.

**LOOKUPSWITCH** - Static variable in class `jminusminus.CLConstants`

`LOOKUPSWITCH` instruction.

**lookupType(String)** - Method in class `jminusminus.Context`

Return the definition for a type name in the environment.

**LOOP\_DETECTED** - Static variable in error `jminusminus.TokenMgrError`

Detected (and bailed out of) an infinite loop in the token manager.

**loopDepth** - Variable in class `jminusminus.NBasicBlock`

Depth of a loop.

**loopIndex** - Variable in class `jminusminus.NBasicBlock`

Index of a loop.

**loopIndex** - Static variable in class `jminusminus.NControlFlowGraph`

Loop identifier.

**LOR** - Static variable in class `jminusminus.CLConstants`

`LOR` instruction.

**lowBytes()** - Method in class `jminusminus.CLConstantDoubleInfo`

Return `CONSTANT_Double_info.low_bytes` item.

**lowBytes()** - Method in class `jminusminus.CLConstantLongInfo`

Return `CONSTANT_Long_info.low_bytes` item.

**LPAREN** - Static variable in interface `jminusminus.JavaCCParserConstants`

**LREM** - Static variable in class `jminusminus.CLConstants`

`LREM` instruction.

**LRETURN** - Static variable in class `jminusminus.CLConstants`

`LRETURN` instruction.

**LSHL** - Static variable in class `jminusminus.CLConstants`

`LSHL` instruction.

**LSHR** - Static variable in class `jminusminus.CLConstants`

`LSHR` instruction.

**LSTORE** - Static variable in class `jminusminus.CLConstants`

`LSTORE` instruction.

**LSTORE\_0** - Static variable in class `jminusminus.CLConstants`

`LSTORE_0` instruction.

**LSTORE\_1** - Static variable in class `jminusminus.CLConstants`

`LSTORE_1` instruction.

**LSTORE\_2** - Static variable in class `jminusminus.CLConstants`

`LSTORE_2` instruction.

**LSTORE\_3** - Static variable in class `jminusminus.CLConstants`

`LSTORE_3` instruction.

**LSUB** - Static variable in class `jminusminus.CLConstants`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

LSUB instruction.

**IType** - Variable in class `jminusminus.NHIRInstruction`

Long type name for this instruction.

**LUSHR** - Static variable in class `jminusminus.CLConstants`

LUSHR instruction.

**LXOR** - Static variable in class `jminusminus.CLConstants`

LXOR instruction.

## M

**MAGIC** - Static variable in class `jminusminus.CLConstants`

Magic number (0xCAFEBAFE) identifying the class file format.

**magic** - Variable in class `jminusminus.CLFile`

`ClassFile.magic` item.

**main(String[])** - Static method in class `jminusminus.CLAbsorber`

Driver for `CLAbsorber`.

**main(String[])** - Static method in class `jminusminus.JavaCCMain`

Entry point.

**Main** - Class in `jminusminus`

Driver class for j-- compiler using hand-written front-end.

**Main()** - Constructor for class `jminusminus.Main`

**main(String[])** - Static method in class `jminusminus.Main`

Entry point.

**MAJOR\_VERSION** - Static variable in class `jminusminus.CLConstants`

Major version for the class files that j-- compiles.

**majorVersion** - Variable in class `jminusminus.CLFile`

`ClassFile.major_version` item.

**markProperUseOfConstructor()** - Method in class `jminusminus.JSuperConstruction`

Used in `JConstructorDeclaration` to mark `super(...)` as being properly placed, ie, as the first statement in its body.

**markProperUseOfConstructor()** - Method in class `jminusminus.JThisConstruction`

Used in `JConstructorDeclaration` to mark `this(...)` as being properly placed, ie, as the first statement in its body.

**matchesExpected(Type)** - Method in class `jminusminus.Type`

Does this type match the expected type? For now, "matches" means "equals".

**MAX\_COUNT** - Static variable in class `jminusminus.NPhysicalRegister`

Maximum number of physical registers used for allocation, starting at T0.

**maxIntervals** - Variable in class `jminusminus.NControlFlowGraph`

The total number of intervals.

**maxLocals** - Variable in class `jminusminus.CLCodeAttribute`

`Code_attribute.max_locals` item.

**maxNextCharInd** - Variable in class `jminusminus.SimpleCharStream`

**maxStack** - Variable in class `jminusminus.CLCodeAttribute`

`Code_attribute.max_stack` item.

**member()** - Method in class `jminusminus.Constructor`

**member()** - Method in class `jminusminus.Field`

**Member** - Class in `jminusminus`

A wrapper for members (eg Fields, Methods, Constructors) in the Java API.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



**Member()** - Constructor for class `jminusminus.Member`

**member()** - Method in class `jminusminus.Member`

Return the member's internal representation.

**member()** - Method in class `jminusminus.Method`

Return the member's internal representation.

**Method** - Class in `jminusminus`

A Method knows its descriptor (its signature in JVM format), and its return type.

**Method(Method)** - Constructor for class `jminusminus.Method`

Construct a Method is constructed from its internal representation.

**methodAccessFlagsToString(int)** - Static method in class `jminusminus.CLFile`

Return (as a string) the method access permissions and properties contained in the specified mask of flags.

**methodContext()** - Method in class `jminusminus.Context`

Return the closest surrounding method context.

**MethodContext** - Class in `jminusminus`

A method context is where formal parameters are declared.

**MethodContext(Context, boolean, Type)** - Constructor for class `jminusminus.MethodContext`

Construct a method context.

**methodFor(String, Type[])** - Method in class `jminusminus.Type`

Find an appropriate method in this type, given a message (method) name and it's argument types.

**methodHasReturn()** - Method in class `jminusminus.MethodContext`

Does this (non-void) method have at least one return?

**methodIndex** - Variable in class `jminusminus.CLEnclosingMethodAttribute`

`EnclosingMethod_attribute.method_index` item.

**methodReturntype()** - Method in class `jminusminus.MethodContext`

Return the return type of this method.

**methods** - Variable in class `jminusminus.CLFile`

`ClassFile.methods` item.

**methodsCount** - Variable in class `jminusminus.CLFile`

`ClassFile.methods_count` item.

**MINOR\_VERSION** - Static variable in class `jminusminus.CLConstants`

Minor version for the class files that j-- compiles.

**minorVersion** - Variable in class `jminusminus.CLFile`

`ClassFile.minor_version` item.

**MINUS** - Static variable in interface `jminusminus.JavaCCParserConstants`

**mnemonic** - Variable in class `jminusminus.CLInsInfo`

Mnemonic for this instruction.

**mnemonic** - Variable in class `jminusminus.CLInstruction`

Mnemonic for this instruction.

**mnemonic()** - Method in class `jminusminus.CLInstruction`

Return the mnemonic for this instruction.

**mnemonic** - Variable in class `jminusminus.NTuple`

String representation (mnemonic) of the instruction.

**mods()** - Method in class `jminusminus.JFieldDeclaration`

Return the list of modifiers.

**mods** - Variable in class `jminusminus.JMethodDeclaration`

Method modifiers.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**mods()** - Method in class jminusminus.JVariableDeclaration

Return the list of modifiers.

**MONITORENTER** - Static variable in class jminusminus.CLConstants

MONITORENTER instruction.

**MONITOREXIT** - Static variable in class jminusminus.CLConstants

MONITOREXIT instruction.

**MULTIANEWARRAY** - Static variable in class jminusminus.CLConstants

MULTIANEWARRAY instruction.

**mustMatchExpected(int, Type)** - Method in class jminusminus.Type

An assertion that this type matches the specified type.

**mustMatchOneOf(int, Type...)** - Method in class jminusminus.Type

An assertion that this type matches one of the specified types.

## N

**name()** - Method in class jminusminus.JClassDeclaration

Return the class name.

**name()** - Method in class jminusminus.JFormalParameter

Return the parameter's name.

**name** - Variable in class jminusminus.JMethodDeclaration

Method name.

**name()** - Method in interface jminusminus.JTypeDecl

Return the name of this type declaration.

**name()** - Method in class jminusminus.JVariable

Return the identifier name.

**name()** - Method in class jminusminus.JVariableDeclaration

Return the variable name.

**name()** - Method in class jminusminus.Member

Return the member's (simple) name.

**name** - Variable in class jminusminus.NControlFlowGraph

Name of the method this cfg corresponds to.

**name** - Variable in class jminusminus.NHIRGetField

Name of the field being accessed.

**name** - Variable in class jminusminus.NHIRInvoke

Name of the method being invoked.

**name** - Variable in class jminusminus.NHIRPutField

Name of the field being accessed.

**name** - Variable in class jminusminus.NLIRGetField

Name of the field being accessed.

**name** - Variable in class jminusminus.NLIRInvoke

Name of the method being invoked.

**name** - Variable in class jminusminus.NLIRPutField

Name of the field being accessed.

**name** - Variable in class jminusminus.NRegister

Register name.

**name()** - Method in class jminusminus.NRegister

Return the name of this register.

**nameAndTypeIndex** - Variable in class jminusminus.CLConstantMemberRefInfo

CONSTANT\_Memberref\_info.name\_and\_type\_index item.

**nameIndex** - Variable in class jminusminus.CLConstantClassInfo

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

CONSTANT\_Class\_info.name\_index item.

**nameIndex** - Variable in class jminusminus.CLCConstantNameAndTypeInfo

CONSTANT\_NameAndType\_info.name\_index item.

**nameIndex** - Variable in class jminusminus.CLLocalVariableInfo

local\_variable\_table\_entry.name\_index item.

**nameIndex** - Variable in class jminusminus.CLLocalVariableTypeInfo

local\_variable\_type\_table\_entry.name\_index item.

**nameIndex** - Variable in class jminusminus.CLMemberInfo

member\_info.name\_index item.

**names()** - Method in class jminusminus.Context

The names declared in this context.

**NarrowReference** - Class in jminusminus

A narrowing conversion on a reference type requires a run-time check on the type.

**NarrowReference(Type)** - Constructor for class jminusminus.NarrowReference

Construct a narrowing converter.

**NBasicBlock** - Class in jminusminus

Representation of a block within a control flow graph.

**NBasicBlock(NControlFlowGraph, int)** - Constructor for class jminusminus.NBasicBlock

Construct a block given its unique identifier.

**NControlFlowGraph** - Class in jminusminus

Representation of a control flow graph (cfg) for a method.

**NControlFlowGraph(CLCConstantPool, CLMethodInfo)** - Constructor for class jminusminus.NControlFlowGraph

Construct an NControlFlowGraph object for a method given the constant pool for the class containing the method and the object containing information about the method.

**NEmitter** - Class in jminusminus

A class for generating native SPIM code.

**NEmitter(String, ArrayList<CLFile>, String)** - Constructor for class jminusminus.NEmitter

Construct an NEmitter instance.

**nestedCodegen(CLEmitter)** - Method in class jminusminus.JStringConcatenationOp

Like a codegen() but we needn't (and shouldn't) create a StringBuilder nor convert the result to a String, as that will be done in a parent.

**NEW** - Static variable in class jminusminus.CLConstants

NEW instruction.

**NEW** - Static variable in interface jminusminus.JavaCCParserConstants

**NEWARRAY** - Static variable in class jminusminus.CLConstants

NEWARRAY instruction.

**newFirstRangeStart(int)** - Method in class jminusminus.NInterval

Sets the start value of the very first range.

**newToken(int)** - Static method in class jminusminus.Token

Returns a new Token object, by default.

**next** - Variable in class jminusminus.JavaCCParser.JJCalls

**next()** - Method in class jminusminus.LookaheadScanner

Scan to the next token in the input.

**next** - Variable in class jminusminus.Token

A reference to the next regular (non-special) token from the input stream.

**nextChar()** - Method in class jminusminus.CharReader

Scan the next character.

Assignment Project Exam Help  
<https://powcoder.com>  
Add WeChat powcoder

**nextIntersection(NInterval)** - Method in class `jminusminus.NInterval`

Looks for the very first position where an intersection with another interval occurs.

**nextOffset()** - Method in class `jminusminus.LocalContext`

Allocate a new offset (eg for a parameter or local variable).

**nextUsageOverlapping(NInterval)** - Method in class `jminusminus.NInterval`

The next use position of this interval after the first range start of the foreign interval.

**NGraphRegisterAllocator** - Class in `jminusminus`

Implements register allocation using graph coloring algorithm.

**NGraphRegisterAllocator(NControlFlowGraph)** - Constructor for class `jminusminus.NGraphRegisterAllocator`

Construct a `NGraphRegisterAllocator`.

**NHIRALoad** - Class in `jminusminus`

HIR instruction representing JVM array load instructions.

**NHIRALoad(NBasicBlock, int, int, int, int, String, String)** - Constructor for class `jminusminus.NHIRALoad`

Construct an `NHIRALoad` instruction.

**NHIRArithmetic** - Class in `jminusminus`

HIR instruction corresponding to the JVM arithmetic instructions.

**NHIRArithmetic(NBasicBlock, int, int, int, int)** - Constructor for class `jminusminus.NHIRArithmetic`

Construct an `NHIRArithmetic` instruction.

**NHIRASTore** - Class in `jminusminus`

HIR instruction representing JVM array store instructions.

**NHIRASTore(NBasicBlock, int, int, int, int, int, String, String)** - Constructor for class `jminusminus.NHIRASTore`

Construct an `NHIRASTore` instruction.

**NHIRConditionalJump** - Class in `jminusminus`

HIR instruction representing an conditional jump instructions in JVM.

**NHIRConditionalJump(NBasicBlock, int, int, int, int, NBasicBlock, NBasicBlock)** - Constructor for class `jminusminus.NHIRConditionalJump`

Construct an `NHIRConditionalJump` instruction.

**NHIRGetField** - Class in `jminusminus`

HIR instruction representing JVM (get) field instructions.

**NHIRGetField(NBasicBlock, int, int, String, String, String, String)** - Constructor for class `jminusminus.NHIRGetField`

Construct an `NHIRGetField` instruction.

**NHIRGoto** - Class in `jminusminus`

HIR instruction representing an unconditional jump instruction in JVM.

**NHIRGoto(NBasicBlock, int, NBasicBlock)** - Constructor for class `jminusminus.NHIRGoto`

Construct an `NHIRGoto` instruction.

**NHIRInstruction** - Class in `jminusminus`

High-level intermediate representation (HIR) of a JVM instruction.

**NHIRInstruction(NBasicBlock, int)** - Constructor for class `jminusminus.NHIRInstruction`

Construct an `NHIRInstruction` object.

**NHIRInstruction(NBasicBlock, int, String, String)** - Constructor for class `jminusminus.NHIRInstruction`

Construct an `NHIRInstruction` object.

**NHIRIntConstant** - Class in `jminusminus`

HIR instruction corresponding to the JVM instructions representing integer constants.

**NHIRIntConstant(NBasicBlock, int, int)** - Constructor for class `jminusminus.NHIRIntConstant`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Construct an NHIRIntConstant instruction.

**NHIRInvoke** - Class in jminusminus

HIR instruction representing method invocation instructions in JVM.

**NHIRInvoke(NBasicBlock, int, int, String, String, ArrayList<Integer>, String, String)** -

Constructor for class jminusminus.NHIRInvoke

Construct an NHIRInvoke instruction.

**NHIRLoadLocal** - Class in jminusminus

HIR instruction representing a formal parameter.

**NHIRLoadLocal(NBasicBlock, int, int, String, String)** - Constructor for class

jminusminus.NHIRLoadLocal

Construct an NHIRLoadLocal instruction.

**NHIRLocal** - Class in jminusminus

HIR instruction representing a local (not formal) variable.

**NHIRLocal(NBasicBlock, int, int, String, String)** - Constructor for class

jminusminus.NHIRLocal

Construct an NHIRLocal instruction.

**NHIRNewArray** - Class in jminusminus

HIR instruction representing JVM array creation instructions.

**NHIRNewArray(NBasicBlock, int, int, int, String, String)** - Constructor for class

jminusminus.NHIRNewArray

Construct an NHIRNewArray instruction.

**NHIRPhiFunction** - Class in jminusminus

HIR instruction representing phi functions.

**NHIRPhiFunction(NBasicBlock, int, ArrayList<Integer>, int)** - Constructor for class

jminusminus.NHIRPhiFunction

Construct an NHIRPhiFunction instruction.

**NHIRPutField** - Class in jminusminus

HIR instruction representing JVM (put) field instructions.

**NHIRPutField(NBasicBlock, int, int, String, String, String, String, int)** - Constructor for class

jminusminus.NHIRPutField

Construct an NHIRPutField instruction.

**NHIRReturn** - Class in jminusminus

HIR instruction representing a JVM return instruction.

**NHIRReturn(NBasicBlock, int, int, int)** - Constructor for class jminusminus.NHIRReturn

Construct an NHIRReturn instruction.

**NHIRStringConstant** - Class in jminusminus

HIR instruction corresponding to the JVM instructions representing string constants.

**NHIRStringConstant(NBasicBlock, int, String)** - Constructor for class

jminusminus.NHIRStringConstant

Construct an NHIRStringConstant instruction.

**NInterval** - Class in jminusminus

A lifetime interval, recording the interval of LIR code for which the corresponding virtual register contains a useful value.

**NInterval(int, NControlFlowGraph)** - Constructor for class jminusminus.NInterval

Construct a NInterval with the given virtual register ID for the given control flow graph.

**NInterval(int, NControlFlowGraph, ArrayList<NRange>, NInterval)** - Constructor for class

jminusminus.NInterval

This second constructor is used in instantiating children of a split interval.

**NLinearRegisterAllocator** - Class in jminusminus

Implements the Linear Scan register allocation algorithm.

**NLinearRegisterAllocator(NControlFlowGraph)** - Constructor for class

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



**jminusminus.NLinearRegisterAllocator**

Construct a linear register allocator for the given control flow graph.

**NLIRALoad** - Class in **jminusminus**

LIR instruction representing JVM array load instructions.

**NLIRALoad(NBasicBlock, int, int, NLIRInstruction, NLIRInstruction, String, String)** -

Constructor for class **jminusminus.NLIRALoad**

Construct an **NLIRALoad** instruction.

**NLIRArithmetic** - Class in **jminusminus**

LIR instruction corresponding to the JVM arithmetic instructions.

**NLIRArithmetic(NBasicBlock, int, int, NLIRInstruction, NLIRInstruction)** - Constructor for class **jminusminus.NLIRArithmetic**

Construct an **NLIRArithmetic** instruction.

**NLIRASTore** - Class in **jminusminus**

LIR instruction representing JVM array store instructions.

**NLIRASTore(NBasicBlock, int, int, NLIRInstruction, NLIRInstruction, NLIRInstruction, String, String)** - Constructor for class **jminusminus.NLIRASTore**

Construct an **NLIRASTore** instruction.

**NLIRConditionalJump** - Class in **jminusminus**

LIR instruction representing an conditional jump instructions in JVM.

**NLIRConditionalJump(NBasicBlock, int, NLIRInstruction, NLIRInstruction, int, NBasicBlock, NBasicBlock)** - Constructor for class **jminusminus.NLIRConditionalJump**

Construct an **NLIRConditionalJump** instruction.

**NLIRGetField** - Class in **jminusminus**

LIR instruction representing JVM (get) field instructions.

**NLIRGetField(NBasicBlock, int, int, String, String, String, String)** - Constructor for class **jminusminus.NLIRGetField**

Construct an **NLIRGetField** instruction.

**NLIRGoto** - Class in **jminusminus**

LIR instruction representing an conditional jump instructions in JVM.

**NLIRGoto(NBasicBlock, int, NBasicBlock)** - Constructor for class **jminusminus.NLIRGoto**

Construct an **NLIRGoto** instruction.

**NLIRInstruction** - Class in **jminusminus**

Low-level intermediate representation (LIR) of a JVM instruction.

**NLIRInstruction(NBasicBlock, int)** - Constructor for class **jminusminus.NLIRInstruction**

Construct an **NLIRInstruction**.

**NLIRIntConstant** - Class in **jminusminus**

LIR instruction corresponding to the JVM instructions representing integer constants.

**NLIRIntConstant(NBasicBlock, int, int)** - Constructor for class **jminusminus.NLIRIntConstant**

Construct an **NLIRIntConstant** instruction.

**NLIRInvoke** - Class in **jminusminus**

LIR instruction representing method invocation instructions in JVM.

**NLIRInvoke(NBasicBlock, int, int, String, String, ArrayList<NRegister>, String, String)** - Constructor for class **jminusminus.NLIRInvoke**

Construct an **NLIRInvoke** instruction.

**NLIRLoad** - Class in **jminusminus**

LIR instruction representing a load from memory to register.

**NLIRLoad(NBasicBlock, int, int, OffsetFrom, NRegister)** - Constructor for class **jminusminus.NLIRLoad**

Construct an **NLIRLoad** instruction.

**NLIRLoadLocal** - Class in **jminusminus**

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

LIR instruction representing a formal parameter.

**NLIRLoadLocal(NBasicBlock, int, int, String, String)** - Constructor for class `jminusminus.NLIRLoadLocal`

Construct an **NLIRLoadLocal** instruction.

**NLIRMove** - Class in `jminusminus`

LIR move instruction.

**NLIRMove(NBasicBlock, int, NLIRInstruction, NLIRInstruction)** - Constructor for class `jminusminus.NLIRMove`

Construct an **NLIRMove** instruction.

**NLIRMove(NBasicBlock, int, NRegister, NRegister)** - Constructor for class `jminusminus.NLIRMove`

Construct an **NLIRMove** instruction.

**NLIRNewArray** - Class in `jminusminus`

LIR instruction representing JVM array creation instructions.

**NLIRNewArray(NBasicBlock, int, int, int, String, String)** - Constructor for class `jminusminus.NLIRNewArray`

Construct an **NLIRNewArray** instruction.

**NLIRPhiFunction** - Class in `jminusminus`

LIR instruction representing phi functions.

**NLIRPhiFunction(NBasicBlock, int, String, String)** - Constructor for class `jminusminus.NLIRPhiFunction`

Construct an **NLIRPhiFunction** instruction.

**NLIRPutField** - Class in `jminusminus`

LIR instruction representing JVM (put) field instructions.

**NLIRPutField(NBasicBlock, int, int, String, String, String, String, NLIRInstruction)** - Constructor for class `jminusminus.NLIRPutField`

Construct an **NLIRPutField** instruction.

**NLIRReturn** - Class in `jminusminus`

LIR instruction representing a JVM return instruction.

**NLIRReturn(NBasicBlock, int, int, NPhysicalRegister)** - Constructor for class `jminusminus.NLIRReturn`

Construct an **NLIRReturn** instruction.

**NLIRStore** - Class in `jminusminus`

LIR instruction representing a store from a register to memory.

**NLIRStore(NBasicBlock, int, int, OffsetFrom, NRegister)** - Constructor for class `jminusminus.NLIRStore`

Construct an **NLIRStore** instruction.

**NLIRStringConstant** - Class in `jminusminus`

LIR instruction corresponding to the JVM instructions representing string constants.

**NLIRStringConstant(NBasicBlock, int, String)** - Constructor for class `jminusminus.NLIRStringConstant`

Construct an **NLIRStringConstant** instruction.

**NNaiveRegisterAllocator** - Class in `jminusminus`

Implements a naive register allocation method.

**NNaiveRegisterAllocator(NControlFlowGraph)** - Constructor for class `jminusminus>NNaiveRegisterAllocator`

Construct a **NNaiveRegisterAllocator**.

**NON\_ZERO\_DIGIT** - Static variable in interface `jminusminus.JavaCCParserConstants`

**NOP** - Static variable in class `jminusminus.CLConstants`

NOP instruction.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**NPhysicalRegister** - Class in `jminusminus`

An abstraction for a physical (SPIM) register.

**NPhysicalRegister(int, String)** - Constructor for class `jminusminus.NPhysicalRegister`

Construct an NPhysicalRegister.

**NRange** - Class in `jminusminus`

A liveness range (for an interval).

**NRange(int, int)** - Constructor for class `jminusminus.NRange`

Construct a liveness range extending from start to stop (positions in the code).

**NRegister** - Class in `jminusminus`

An abstraction for a (virtual or physical) register.

**NRegister(int, String)** - Constructor for class `jminusminus.NRegister`

Construct an NRegister.

**NRegisterAllocator** - Class in `jminusminus`

A register allocator maps virtual registers (from LIR code) to physical registers on the target machine.

**NRegisterAllocator(NControlFlowGraph)** - Constructor for class

`jminusminus.NRegisterAllocator`

Construct an NRegisterAllocator object given the control flow graph for method.

**NTuple** - Class in `jminusminus`

A tuple representation of a JVM instruction.

**NTuple(int, int, ArrayList<Short>)** - Constructor for class `jminusminus.NTuple`

Construct a tuple representing the JVM instruction with the given program counter, opcode, and operand list.

**NULL** - Static variable in interface `jminusminus.JavaCCParserConstants`

**NULLTYPE** - Static variable in class `jminusminus.Type`

The null void.

**numAnnotations** - Variable in class `jminusminus.CLParameterAnnotationInfo`

`parameter_annotations_table_entry.num_annotations` item.

**numAnnotations** - Variable in class `jminusminus.CLRuntimeInvisibleAnnotationsAttribute`

`RuntimeInvisibleAnnotations_attribute.num_annotations` item.

**numAnnotations** - Variable in class `jminusminus.CLRuntimeVisibleAnnotationsAttribute`

`RuntimeVisibleAnnotations_attribute.num_annotations` item.

**number** - Variable in class `jminusminus.NRegister`

Register number.

**number()** - Method in class `jminusminus.NRegister`

Return the number of this register.

**numberOfClasses** - Variable in class `jminusminus.CLInnerClassesAttribute`

`InnerClasses_attribute.number_of_classes` item.

**numberOfExceptions** - Variable in class `jminusminus.CLExceptionsAttribute`

`Exceptions_attribute.number_of_exceptions` item.

**numElementValuePairs** - Variable in class `jminusminus.CLAnnotation`

`annotation.num_element_value_pairs` item.

**numParameters** - Variable in class

`jminusminus.CLRuntimeInvisibleParameterAnnotationsAttribute`

`RuntimeInvisibleParameterAnnotations_attribute.num_parameters` item.

**numParameters** - Variable in class

`jminusminus.CLRuntimeVisibleParameterAnnotationsAttribute`

`RuntimeVisibleParameterAnnotations_attribute.num_parameters` item.

**numValues** - Variable in class `jminusminus.CLElementValue`

Add WeChat powcoder

<https://powcoder.com>

element\_value.array\_value.numValues item.

**NVirtualRegister** - Class in jminusminus

An abstraction for a virtual register.

**NVirtualRegister(int, String, String)** - Constructor for class jminusminus.NVirtualRegister  
Construct an NVirtualRegister.

## O

**OBJECT** - Static variable in class jminusminus.Type

The type java.lang.Object.

**offset** - Variable in class jminusminus.LocalContext

Next offset for a local variable.

**offset()** - Method in class jminusminus.LocalContext

The "next" offset.

**offset()** - Method in class jminusminus.LocalVariableDefn

The offset of this variable on the stack frame.

**offset** - Variable in class jminusminus.NControlFlowGraph

Stack offset counter..

**offset** - Variable in class jminusminus.NInterval

Offset.

**offsetFrom** - Variable in class jminusminus.NInterval

From offset.

**OffsetFrom** - Enum in jminusminus

The types of stack pointers.

**onFalseDestination** - Variable in class jminusminus.NHIRConditionalJump

Block to jump to on false.

**onFalseDestination** - Variable in class jminusminus.NLIRConditionalJump

Block to jump to on false.

**onTrueDestination** - Variable in class jminusminus.NHIRConditionalJump

Block to jump to on true.

**onTrueDestination** - Variable in class jminusminus.NLIRConditionalJump

Block to jump to on true.

**opcode** - Variable in class jminusminus.CLInsInfo

Opcode for this instruction.

**opcode** - Variable in class jminusminus.CLInstruction

Opcode for this instruction.

**opcode()** - Method in class jminusminus.CLInstruction

Return the opcode for this instruction.

**opcode** - Variable in class jminusminus.NHIRALoad

Opcode of the JVM instruction.

**opcode** - Variable in class jminusminus.NHIRArithmetic

Opcode for the arithmetic operator.

**opcode** - Variable in class jminusminus.NHIRASTore

Opcode of the JVM instruction.

**opcode** - Variable in class jminusminus.NHIRConditionalJump

Test expression opcode.

**opcode** - Variable in class jminusminus.NHIRGetField

Opcode of the JVM instruction.

**opcode** - Variable in class jminusminus.NHIRInvoke

Opcode of the JVM instruction.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**opcode** - Variable in class `jminusminus.NHIRNewArray`  
Opcode of the JVM instruction.

**opcode** - Variable in class `jminusminus.NHIRPutField`  
Opcode of the JVM instruction.

**opcode** - Variable in class `jminusminus.NHIRReturn`  
JVM opcode for the return instruction.

**opcode** - Variable in class `jminusminus.NLIRALoad`  
Opcode of the JVM instruction.

**opcode** - Variable in class `jminusminus.NLIRASTore`  
Opcode of the JVM instruction.

**opcode** - Variable in class `jminusminus.NLIRConditionalJump`  
Test expression opcode.

**opcode** - Variable in class `jminusminus.NLIRGetField`  
Opcode of the JVM instruction.

**opcode** - Variable in class `jminusminus.NLIRInvoke`  
Opcode of the JVM instruction.

**opcode** - Variable in class `jminusminus.NLIRNewArray`  
Opcode of the JVM instruction.

**opcode** - Variable in class `jminusminus.NLIRPutField`  
Opcode of the JVM instruction.

**opcode** - Variable in class `jminusminus.NLIRReturn`  
JVM opcode for the return instruction.

**opcode** - Variable in class `jminusminus.NTuple`  
Opcode of the instruction.

**open(String, int, int)** - Static method in class `SPIM`  
Wrapper for `SPIM.open()`.

**operandCount** - Variable in class `jminusminus.CLInsInfo`  
Number of operands for this instruction.

**operandCount** - Variable in class `jminusminus.CLInstruction`  
Number of operands for this instruction; determined statically for all instructions except `TABLESWITCH` and `LOOKUPSWITCH`.

**operandCount()** - Method in class `jminusminus.CLInstruction`  
Return the number of operands for this instruction.

**operands** - Variable in class `jminusminus.NTuple`  
Operands of the instructions.

**operator** - Variable in class `jminusminus.JBinaryExpression`  
The binary operator.

**optimize()** - Method in class `jminusminus.NControlFlowGraph`  
Carry out optimizations on the high-level instructions.

**orderBlocks()** - Method in class `jminusminus.NControlFlowGraph`  
Compute optimal ordering of the basic blocks in this cfg.

**outerClassInfoIndex** - Variable in class `jminusminus.CLInnerClassInfo`  
`classes_table_entry.outer_class_info_index` item.

## P

**PACKAGE** - Static variable in interface `jminusminus.JavaCCParserConstants`

**packageName()** - Method in class `jminusminus.JCompilationUnit`  
The package in which this compilation unit is defined.

**packageName()** - Method in class `jminusminus.Type`



Return the type's package name.

**parameterAnnotations** - Variable in class

jminusminus.CLRuntimeInvisibleParameterAnnotationsAttribute

RuntimeInvisibleParameterAnnotations\_attribute.

**parameterAnnotations** - Variable in class

jminusminus.CLRuntimeVisibleParameterAnnotationsAttribute

RuntimeVisibleParameterAnnotations\_attribute.

**params** - Variable in class jminusminus.JMethodDeclaration

The formal parameters.

**parent** - Variable in class jminusminus.NInterval

Parent of this interval.

**ParseException** - Exception in jminusminus

This exception is thrown when parse errors are encountered.

**ParseException(Token, int[][], String[])** - Constructor for exception

jminusminus.ParseException

This constructor is used by the method "generateParseException" in the generated parser.

**ParseException()** - Constructor for exception jminusminus.ParseException

The following constructors are for use by you for whatever purpose you can think of.

**ParseException(String)** - Constructor for exception jminusminus.ParseException

**Parser** - Class in jminusminus

A recursive descent parser that, given a lexical analyzer (a LookaheadScanner), parses a Java compilation unit (program file), taking tokens from the LookaheadScanner, and produces an abstract syntax tree (AST) for it.

**Parser(LookaheadScanner)** - Constructor for class jminusminus.Parser

Construct a parser from the given lexical analyzer.

**partialCodegen(Context, CLEmitter)** - Method in class jminusminus.JAST

Generate a partial class for this type, reflecting only the member information required to do analysis.

**partialCodegen(Context, CLEmitter)** - Method in class jminusminus.JConstructorDeclaration

Add this constructor declaration to the partial class.

**partialCodegen(Context, CLEmitter)** - Method in class jminusminus.JMethodDeclaration

Add this method declaration to the partial class.

**pc()** - Method in class jminusminus.CLEmitter

Return the pc (location counter).

**pc** - Variable in class jminusminus.CLInstruction

Location counter; index of this instruction within the code array of a method.

**pc()** - Method in class jminusminus.CLInstruction

Return the pc for this instruction.

**pc** - Variable in class jminusminus.NTuple

Program counter of the instruction.

**PLUS** - Static variable in interface jminusminus.JavaCCParserConstants

**PLUS\_ASSIGN** - Static variable in interface jminusminus.JavaCCParserConstants

**pop()** - Method in class jminusminus.CLBranchStack

Pop and return an element from the stack.

**POP** - Static variable in class jminusminus.CLConstants

POP instruction.

**POP2** - Static variable in class jminusminus.CLConstants

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

POP2 instruction.

**preAnalyze(Context)** - Method in class jminusminus.JClassDeclaration

Pre-analyze the members of this declaration in the parent context.

**preAnalyze()** - Method in class jminusminus.JCompilationUnit

Construct a context for the compilation unit, initializing it with imported types.

**preAnalyze(Context, CLEmitter)** - Method in class jminusminus.JConstructorDeclaration

Declare this constructor in the parent (class) context.

**preAnalyze(Context, CLEmitter)** - Method in class jminusminus.JFieldDeclaration

Declare fields in the parent's (partial) class.

**preAnalyze(Context, CLEmitter)** - Method in interface jminusminus.JMember

Declare the member name(s) in the specified (class) context.

**preAnalyze(Context, CLEmitter)** - Method in class jminusminus.JMethodDeclaration

Declare this method in the parent (class) context.

**preAnalyze(Context)** - Method in interface jminusminus.JTypeDecl

Pre-analyze the members of this declaration in the parent context.

**predecessors** - Variable in class jminusminus.NBasicBlock

List of predecessor blocks.

**pRegister** - Variable in class jminusminus.NInterval

The NPhysicalRegister assigned to this interval.

**pRegisters** - Variable in class jminusminus.NControlFlowGraph

Physical registers allocated for this cfg by the HIR to LIR conversion algorithm.

**PrettyPrinter** - Class in jminusminus

A utility class that allows pretty (indented) printing to STDOUT.

**PrettyPrinter()** - Constructor for class jminusminus.PrettyPrinter

Construct a PrettyPrinter with an indentation width of 2.

**PrettyPrinter(int)** - Constructor for class jminusminus.PrettyPrinter

Construct a PrettyPrinter given the indentation width.

**prevCharIsCR** - Variable in class jminusminus.SimpleCharStream

**prevCharIsLF** - Variable in class jminusminus.SimpleCharStream

**previousToken()** - Method in class jminusminus.LookaheadScanner

The previously scanned token.

**print(String)** - Method in class jminusminus.PrettyPrinter

Print the specified string to STDOUT.

**printChar(char)** - Static method in class spim.SPIM

Wrapper for SPIM.printChar().

**printDouble(double)** - Static method in class spim.SPIM

Wrapper for SPIM.printDouble().

**printf(String, Object...)** - Method in class jminusminus.PrettyPrinter

Print args to STDOUT according to the specified format.

**printFloat(float)** - Static method in class spim.SPIM

Wrapper for SPIM.printFloat().

**println(int)** - Static method in class spim.SPIM

Wrapper for SPIM.println().

**println()** - Method in class jminusminus.PrettyPrinter

Print an empty line to STDOUT.

**println(String)** - Method in class jminusminus.PrettyPrinter

Print the specified string (followed by a newline) to STDOUT.

**printString(String)** - Static method in class spim.SPIM

Assignment Project Exam Help

<https://powercoder.com>

Add WeChat powercoder

Wrapper for SPIM.readString().

**PRIVATE** - Static variable in interface jminusminus.JavaCCParserConstants

**PROTECTED** - Static variable in interface jminusminus.JavaCCParserConstants

**PUBLIC** - Static variable in interface jminusminus.JavaCCParserConstants

**push(CLInstruction, int)** - Method in class jminusminus.CLBranchStack

Push the specified information into the stack as a CLBranchTarget instance if the target has not been visited yet.

**PUTFIELD** - Static variable in class jminusminus.CLConstants

PUTFIELD instruction.

**PUTSTATIC** - Static variable in class jminusminus.CLConstants

PUTSTATIC instruction.

## R

**RA** - Static variable in class jminusminus.NPhysicalRegister

Return address (used by function call).

**rangeOverlaps(NRange)** - Method in class jminusminus.NRange

Does this liveness range overlap with another?

**ranges** - Variable in class jminusminus.NInterval

All live ranges for this virtual register.

**RBRACK** - Static variable in interface jminusminus.JavaCCParserConstants

**RCURLY** - Static variable in interface jminusminus.JavaCCParserConstants

**read(int, int)** - Static method in class spim.SPIM

Wrapper for SPIM.read().

**readChar()** - Method in class jminusminus.SimpleCharStream

**readChar()** - Static method in class spim.SPIM

Wrapper for SPIM.readChar().

**readDouble()** - Static method in class spim.SPIM

Wrapper for SPIM.readDouble().

**readFloat()** - Static method in class spim.SPIM

Wrapper for SPIM.readFloat().

**readInt()** - Static method in class spim.SPIM

Wrapper for SPIM.readInt().

**reads** - Variable in class jminusminus.NLIRInstruction

Registers that store the inputs (if any) of this instruction.

**readString(int)** - Static method in class spim.SPIM

Wrapper for SPIM.readString().

**readUnsignedInt()** - Method in class jminusminus.CLInputStream

Read four input bytes and return a long value in the range 0 through 4294967295.

**reclassify(Context)** - Method in class jminusminus.AmbiguousName

Reclassify the name according to the rules in the Java Language Specification.

**recordPosition()** - Method in class jminusminus.LookaheadScanner

Record the current position in the input, so that we can start looking ahead in the input (and later return to this position).

**ref** - Variable in class jminusminus.NBasicBlock

Ref count of this block.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**regId** - Static variable in class `jminusminus.NControlFlowGraph`

Virtual register identifier.

**regInfo** - Static variable in class `jminusminus.NPhysicalRegister`

Maps register number to the register's representation.

**registers** - Variable in class `jminusminus.NControlFlowGraph`

Registers allocated for this cfg by the HIR to LIR conversion algorithm.

**Relnit(InputStream)** - Method in class `jminusminus.JavaCCParser`

**Relnit(InputStream, String)** - Method in class `jminusminus.JavaCCParser`

**Relnit(Reader)** - Method in class `jminusminus.JavaCCParser`

**Relnit(JavaCCParserTokenManager)** - Method in class `jminusminus.JavaCCParser`

**Relnit(SimpleCharStream)** - Method in class `jminusminus.JavaCCParserTokenManager`

**Relnit(SimpleCharStream, int)** - Method in class `jminusminus.JavaCCParserTokenManager`

**Relnit(Reader, int, int, int)** - Method in class `jminusminus.SimpleCharStream`

**Relnit(Reader, int, int)** - Method in class `jminusminus.SimpleCharStream`

**Relnit(Reader)** - Method in class `jminusminus.SimpleCharStream`

**Relnit(InputStream, String, int, int, int)** - Method in class `jminusminus.SimpleCharStream`

**Relnit(InputStream, int, int, int)** - Method in class `jminusminus.SimpleCharStream`

**Relnit(InputStream, String)** - Method in class `jminusminus.SimpleCharStream`

**Relnit(InputStream)** - Method in class `jminusminus.SimpleCharStream`

**Relnit(InputStream, String, int, int)** - Method in class `jminusminus.SimpleCharStream`

**Relnit(InputStream, int, int)** - Method in class `jminusminus.SimpleCharStream`

**removeUnreachableBlocks()** - Method in class `jminusminus.NControlFlowGraph`

Remove blocks that cannot be reached from the begin block (B0).

**renumberLirInstructions()** - Method in class `jminusminus.NControlFlowGraph`

Assign new ids to the LIR instructions in this cfg.

**reportSemanticError(int, String, Object...)** - Method in class `jminusminus.JCompilationUnit`

Report a semantic error.

**resolve(Context)** - Method in class `jminusminus.ArrayTypeName`

Resolve this type in the given context.

**resolve(Context)** - Method in class `jminusminus.Type`

Resolve this type in the given context.

**resolve(Context)** - Method in class `jminusminus.TypeName`

Resolve this type in the given context.

**resolveLabels(Hashtable<String, Integer>)** - Method in class `jminusminus.CLEnvironment`

Resolve the jump labels to the corresponding pc values using the given label to pc mapping.

**resolveLabels(Hashtable<String, Integer>)** - Method in class `jminusminus.CLFlowControlInstruction`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Resolve the jump labels to the corresponding offset values using the given label to pc mapping.

**resolvePhiFunctions()** - Method in class `jminusminus.NControlFlowGraph`

Resolve the phi functions in this cfg, i.e., for each  $x = \text{phi}(x_1, x_2, \dots, x_n)$  generate an (LIR) move  $x_i, x$  instruction at the end of the predecessor  $i$  of the block defining the phi function; if the instruction there is a branch, add the instruction prior to the branch.

**RET** - Static variable in class `jminusminus.CLConstants`

RET instruction.

**RETURN** - Static variable in class `jminusminus.CLConstants`

RETURN instruction.

**RETURN** - Static variable in interface `jminusminus.JavaCCParserConstants`

**returnToPosition()** - Method in class `jminusminus.LookaheadScanner`

Return to the previously recorded position in the input stream of tokens.

**returnType** - Variable in class `jminusminus.JMethodDeclaration`

Return type.

**returnType()** - Method in class `jminusminus.Method`

Return the method's return type.

**rhs** - Variable in class `jminusminus.JBinaryExpression`

The rhs operand.

**rhs** - Variable in class `jminusminus.NHIRArithmetic`

Rhs HIR id.

**rhs** - Variable in class `jminusminus.NHIRConditionalJump`

Rhs HIR id.

**RPAREN** - Static variable in interface `jminusminus.JavaCCParserConstants`

Assignment Project Exam Help

<https://powcoder.com>

S

Add WeChat powcoder

**S0** - Static variable in class `jminusminus.NPhysicalRegister`

Temporary (preserved across call).

**S1** - Static variable in class `jminusminus.NPhysicalRegister`

Temporary (preserved across call).

**S2** - Static variable in class `jminusminus.NPhysicalRegister`

Temporary (preserved across call).

**S3** - Static variable in class `jminusminus.NPhysicalRegister`

Temporary (preserved across call).

**S4** - Static variable in class `jminusminus.NPhysicalRegister`

Temporary (preserved across call).

**S5** - Static variable in class `jminusminus.NPhysicalRegister`

Temporary (preserved across call).

**S6** - Static variable in class `jminusminus.NPhysicalRegister`

Temporary (preserved across call).

**S7** - Static variable in class `jminusminus.NPhysicalRegister`

Temporary (preserved across call).

**SALOAD** - Static variable in class `jminusminus.CLConstants`

SALOAD instruction.

**SASTORE** - Static variable in class `jminusminus.CLConstants`

SASTORE instruction.

**Scanner** - Class in `jminusminus`



A lexical analyzer for j--, that has no backtracking mechanism.

**Scanner(String)** - Constructor for class jminusminus.Scanner

Construct a Scanner object.

**SEMI** - Static variable in interface jminusminus.JavaCCParserConstants

**setArgumentCount(int)** - Method in class jminusminus.CLMethodInstruction

Set the number of arguments for the method for INVOKEINTERFACE instruction.

**setClassBytes(byte[])** - Method in class jminusminus.ByteClassLoader

Set the bytes representing the class.

**setClassRep(Class<?>)** - Method in class jminusminus.Type

This setter is used by JCompilationUnit.preAnalyze() to set the classRep to the specified partial class, computed during pre-analysis.

**setDebugStream(PrintStream)** - Method in class jminusminus.JavaCCParserTokenManager

**setInitializer(JExpression)** - Method in class jminusminus.JVariableDeclarator

Set the variable initializer.

**setTabSize(int)** - Method in class jminusminus.SimpleCharStream

**setType(Type)** - Method in class jminusminus.JFormalParameter

Set the type to the specified type.

**setType(Type)** - Method in class jminusminus.JVariableDeclarator

Set the declarator's type.

**signatureFor(String, Type)** - Static method in class jminusminus.Type

A helper for constructing method signatures for reporting unfound methods and constructors.

**signatureIndex** - Variable in class jminusminus.CLocalVariableTypeInfo  
local\_variable\_type\_table\_entry.descriptor\_index item.

**signatureIndex** - Variable in class jminusminus.CLSignatureAttribute

Signature\_attribute\_signature\_index item

**SimpleCharStream** - Class in jminusminus

An implementation of interface CharStream, where the stream is assumed to contain only ASCII characters (without unicode processing).

**SimpleCharStream(Reader, int, int, int)** - Constructor for class jminusminus.SimpleCharStream

**SimpleCharStream(Reader, int, int)** - Constructor for class jminusminus.SimpleCharStream

**SimpleCharStream(Reader)** - Constructor for class jminusminus.SimpleCharStream

**SimpleCharStream(InputStream, String, int, int, int)** - Constructor for class jminusminus.SimpleCharStream

**SimpleCharStream(InputStream, int, int, int)** - Constructor for class jminusminus.SimpleCharStream

**SimpleCharStream(InputStream, String, int, int)** - Constructor for class jminusminus.SimpleCharStream

**SimpleCharStream(InputStream, int, int)** - Constructor for class jminusminus.SimpleCharStream

**SimpleCharStream(InputStream, String)** - Constructor for class jminusminus.SimpleCharStream

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**SimpleCharStream(InputStream)** - Constructor for class jminusminus.SimpleCharStream

**simpleName()** - Method in class jminusminus.Type

Return the simple (unqualified) name for this Type.

**simpleName()** - Method in class jminusminus.TypeName

Return the simple name for this type.

**SINGLE\_LINE\_COMMENT** - Static variable in interface jminusminus.JavaCCParserConstants

**SIPUSH** - Static variable in class jminusminus.CLConstants

SIPUSH instruction.

**size()** - Method in class jminusminus.CLConstantPool

Return the size of the constant pool.

**sourceFileIndex** - Variable in class jminusminus.CLSourceFileAttribute

SourceFile\_attribute.sourcefile\_index item.

**SP** - Static variable in class jminusminus.NPhysicalRegister

Stack pointer.

**specialConstructor** - Variable in exception jminusminus.ParseException

This variable determines which constructor was used to create this object and thereby affects the semantics of the "getMessage" method (see below).

**specialToken** - Variable in class jminusminus.Token

This field is used to access special tokens that occur prior to this token, but after the immediately preceding regular (nonspecial) token.

**spill** - Variable in class jminusminus.NInterval

Whether or not to spill.

**spill()** - Method in class jminusminus.NInterval

Assigns an offset to this interval (if one hasn't been already assigned).

**spim** - package spim

This package contains the SPIM runtime (.s) files and their corresponding Java wrappers.

**SPIM** - Class in spim

This is a Java wrapper class for the SPIM runtime object SPIM.s.

**SPIM()** - Constructor for class spim.SPIM

**splitAt(int)** - Method in class jminusminus.NInterval

Split the current interval at the given index.

**splitRange(int, int)** - Method in class jminusminus.NRange

Mutates current range to be only as long as the split point and returns the remainder as a new range.

**stackDepth** - Variable in class jminusminus.CLBranchTarget

Depth of stack before the target instruction is executed.

**stackUnits** - Variable in class jminusminus.CLInsInfo

Words produced - words consumed from the operand stack by this instruction.

**stackUnits** - Variable in class jminusminus.CLInstruction

Stack units; words produced - words consumed from the operand stack by this instruction.

**stackUnits()** - Method in class jminusminus.CLInstruction

Return the stack units for this instruction.

**STAR** - Static variable in interface jminusminus.JavaCCParserConstants

**start** - Variable in class jminusminus.NRange

The range's start position.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**startLabel** - Variable in class jminusminus.CLEnception

The exception handler is active from this instruction in the code section of the current method being added to ...

**startPC** - Variable in class jminusminus.CLEnception

startLabel is resolved to this value.

**startPC** - Variable in class jminusminus.CLEnceptionInfo

exception\_table\_entry.start\_pc item.

**startPC** - Variable in class jminusminus.CLLineNumberInfo

line\_number\_table\_entry.start\_pc item.

**startPC** - Variable in class jminusminus.CLLocalVariableInfo

local\_variable\_table\_entry.start\_pc item.

**startPC** - Variable in class jminusminus.CLLocalVariableTypeInfo

local\_variable\_type\_table\_entry.start\_pc item.

**startsAtBlock()** - Method in class jminusminus.NInterval

Returns the basic block in which this interval's start position falls.

**statements()** - Method in class jminusminus.JBlock

Return the list of statements comprising the block.

**STATIC** - Static variable in interface jminusminus.JavaCCParserConstants

**STATIC\_LEXER\_ERROR** - Static variable in error jminusminus.TokenMgrError

An attempt was made to create a second instance of a static token manager.

**staticFlag** - Static variable in class jminusminus.SimpleCharStream

**stop** - Variable in class jminusminus.NRange

The range's stop position.

**STRING** - Static variable in class jminusminus.Type

The type java.lang.String.

**STRING\_LITERAL** - Static variable in interface jminusminus.JavaCCParserConstants

**stringIndex** - Variable in class jminusminus.CLConstantStringInfo

CONSTANT\_String\_info.string\_index item.

**sType** - Variable in class jminusminus.NHIRInstruction

Short type name for this instruction.

**successors** - Variable in class jminusminus.NBasicBlock

List of successor blocks.

**SUPER** - Static variable in interface jminusminus.JavaCCParserConstants

**superClass** - Variable in class jminusminus.CLFile

ClassFile.super\_class item.

**superClass()** - Method in class jminusminus.Type

Return the Type's super type (or null if there is none).

**superType()** - Method in class jminusminus.JClassDeclaration

Return the class' super class type.

**superType()** - Method in interface jminusminus.JTypeDecl

Return the super class' type.

**surroundingContext** - Variable in class jminusminus.Context

The surrounding context (scope).

**surroundingContext()** - Method in class jminusminus.Context

Return the surrounding context (scope) in the stack of contexts.

**SWAP** - Static variable in class jminusminus.CLConstants

SWAP instruction.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

## T

- T0** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T1** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T2** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T3** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T4** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T5** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T6** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T7** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T8** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- T9** - Static variable in class jminusminus.NPhysicalRegister  
Temporary (not preserved across call).
- TABLESWITCH** - Static variable in class jminusminus.CLConstants  
TABLESWITCH instruction.
- tabSize** - Variable in class jminusminus.SimpleCharStream
- tag** - Variable in class jminusminus.CLCPInfo  
cp\_info.tag item.
- tag** - Variable in class jminusminus.CLElementValue  
element\_value.tag item.
- target** - Variable in class jminusminus.CLBranchTarget  
Target instruction.
- target** - Variable in class jminusminus.JFieldSelection  
The target expression.
- target** - Variable in class jminusminus.NHIRGetField  
Target for the field.
- target** - Variable in class jminusminus.NHIRInvoke  
Target for the method.
- target** - Variable in class jminusminus.NHIRPutField  
Target for the field.
- target** - Variable in class jminusminus.NLIRGetField  
Target for the field.
- target** - Variable in class jminusminus.NLIRInvoke  
Target for the method.
- target** - Variable in class jminusminus.NLIRPutField  
Target for the field.
- THIS** - Static variable in interface jminusminus.JavaCCParserConstants

**thisClass** - Variable in class jminusminus.CLFile

ClassFile.this\_class item.

**thisType()** - Method in class jminusminus.JClassDeclaration

Return the type that this class declaration defines.

**thisType()** - Method in interface jminusminus.JTypeDecl

Return the type that this type declaration defines.

**toBytes()** - Method in class jminusminus.CLArithmeticInstruction

**toBytes()** - Method in class jminusminus.CLArrayInstruction

**toBytes()** - Method in class jminusminus.CLBitInstruction

**toBytes()** - Method in class jminusminus.CLComparisonInstruction

**toBytes()** - Method in class jminusminus.CLConversionInstruction

**toBytes()** - Method in class jminusminus.CLFieldInstruction

**toBytes()** - Method in class jminusminus.CLFlowControlInstruction

**toBytes()** - Method in class jminusminus.CLInstruction

Return the bytecode for this instruction.

**toBytes()** - Method in class jminusminus.CLPushStoreInstruction

**toBytes()** - Method in class jminusminus.CLMethodInstruction

**toBytes()** - Method in class jminusminus.CLMiscInstruction

**toBytes()** - Method in class jminusminus.CLObjectInstruction

**toBytes()** - Method in class jminusminus.CLStackInstruction

**toClass()** - Method in class jminusminus.CLEmitter

Return the class being constructed as a Java Class instance.

**toDescriptor()** - Method in class jminusminus.ArrayTypeName

Return the JVM descriptor for this type.

**toDescriptor()** - Method in class jminusminus.Constructor

Return the JVM descriptor for this constructor.

**toDescriptor()** - Method in class jminusminus.Method

Return the JVM descriptor for this method.

**toDescriptor()** - Method in class jminusminus.Type

The JVM descriptor for this type.

**toDescriptor()** - Method in class jminusminus.TypeName

Return the JVM descriptor for this type.

**token** - Variable in class jminusminus.JavaCCParser

**token()** - Method in class jminusminus.LookaheadScanner

The currently scanned token.

**Token** - Class in jminusminus

Describes the input token stream.

**Token()** - Constructor for class jminusminus.Token

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



**token\_source** - Variable in class jminusminus.JavaCCParser

**tokenBegin** - Variable in class jminusminus.SimpleCharStream

**tokenImage** - Static variable in interface jminusminus.JavaCCParserConstants

**tokenImage** - Variable in exception jminusminus.ParseException

This is a reference to the "tokenImage" array of the generated parser within which the parse error occurred.

**TokenInfo** - Class in jminusminus

A representation of tokens returned by the lexical analyzer method, getNextToken().

**TokenInfo(TokenKind, String, int)** - Constructor for class jminusminus.TokenInfo

Construct a TokenInfo from its kind, the semantic text forming the token, and its line number.

**TokenInfo(TokenKind, int)** - Constructor for class jminusminus.TokenInfo

Construct a TokenInfo from its kind, and its line number.

**TokenKind** - Enum in jminusminus

An enum of token kinds.

**TokenMgrError** - Error in jminusminus

**TokenMgrError()** - Constructor for error jminusminus.TokenMgrError

**TokenMgrError(String, int)** - Constructor for error jminusminus.TokenMgrError

**TokenMgrError(boolean, int, int, int, String, char, int)** - Constructor for error jminusminus.TokenMgrError

**tokenRep()** - Method in class jminusminus.TokenInfo

Return the token's string representation.

**toLir()** - Method in class jminusminus.NHIRALoad

**toLir()** - Method in class jminusminus.NHIRArithmetic

**toLir()** - Method in class jminusminus.NHIRASTore

**toLir()** - Method in class jminusminus.NHIRConditionalJump

**toLir()** - Method in class jminusminus.NHIRGetField

**toLir()** - Method in class jminusminus.NHIRGoto

**toLir()** - Method in class jminusminus.NHIRInstruction

Convert and return a low-level representation (LIR) of this HIR instruction.

**toLir()** - Method in class jminusminus.NHIRIntConstant

**toLir()** - Method in class jminusminus.NHIRInvoke

**toLir()** - Method in class jminusminus.NHIRLoadLocal

**toLir()** - Method in class jminusminus.NHIRNewArray

**toLir()** - Method in class jminusminus.NHIRPhiFunction

**toLir()** - Method in class jminusminus.NHIRPutField

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**toLir()** - Method in class jminusminus.NHIRReturn

**toLir()** - Method in class jminusminus.NHIRStringConstant

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRALoad

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRArithmetic

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRASTore

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRConditionalJump

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRGetField

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRGoto

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRInstruction

Translate this LIR instruction into SPIM and write it out to the specified output stream.

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRIntConstant

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRInvoke

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRLoad

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRMove

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRNewArray

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRPutField

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRReturn

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRStore

**toSpim(PrintWriter)** - Method in class jminusminus.NLIRStringConstant

**toString()** - Method in class jminusminus.ArrayTypeName

A string representation of the type in Java form.

**toString()** - Method in class jminusminus.Method

Return the Java representation for this method.

**toString()** - Method in class jminusminus.NBasicBlock

Return a string representation of this block.

**toString()** - Method in class jminusminus.NHIRALoad

**toString()** - Method in class jminusminus.NHIRArithmetic

**toString()** - Method in class jminusminus.NHIRASTore

**toString()** - Method in class jminusminus.NHIRConditionalJump

**toString()** - Method in class jminusminus.NHIRGetField

**toString()** - Method in class jminusminus.NHIRGoto

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**toString()** - Method in class jminusminus.NHIRInstruction

Return a string representation of this instruction.

**toString()** - Method in class jminusminus.NHIRIntConstant

**toString()** - Method in class jminusminus.NHIRInvoke

**toString()** - Method in class jminusminus.NHIRLoadLocal

**toString()** - Method in class jminusminus.NHIRLocal

**toString()** - Method in class jminusminus.NHIRNewArray

**toString()** - Method in class jminusminus.NHIRPhiFunction

**toString()** - Method in class jminusminus.NHIRPutField

**toString()** - Method in class jminusminus.NHIRReturn

**toString()** - Method in class jminusminus.NHIRStringConstant

**toString()** - Method in class jminusminus.NLIRALoad

**toString()** - Method in class jminusminus.NLIRArithmetic

**toString()** - Method in class jminusminus.NLIRStore

**toString()** - Method in class jminusminus.NLIRConditionalJump

**toString()** - Method in class jminusminus.NLIRGetField

**toString()** - Method in class jminusminus.NLIRGoto

**toString()** - Method in class jminusminus.NLIRInstruction

Return a string representation of this instruction.

**toString()** - Method in class jminusminus.NLIRIntConstant

**toString()** - Method in class jminusminus.NLIRInvoke

**toString()** - Method in class jminusminus.NLIRLoad

**toString()** - Method in class jminusminus.NLIRLoadLocal

**toString()** - Method in class jminusminus.NLIRMove

**toString()** - Method in class jminusminus.NLIRNewArray

**toString()** - Method in class jminusminus.NLIRPhiFunction

**toString()** - Method in class jminusminus.NLIRPutField

**toString()** - Method in class jminusminus.NLIRReturn

**toString()** - Method in class jminusminus.NLIRStore

**toString()** - Method in class jminusminus.NLIRStringConstant

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**toString()** - Method in class `jminusminus.NPhysicalRegister`

Return a string representation of this physical register.

**toString()** - Method in class `jminusminus.NRange`

The string representation of the range.

**toString()** - Method in class `jminusminus.NVirtualRegister`

Return a string representation of this virtual register.

**toString()** - Method in class `jminusminus.Token`

Returns the image.

**toString()** - Method in class `jminusminus.TokenInfo`

Return the token's image.

**toString()** - Method in enum `jminusminus.TokenKind`

Return the string representation of the token.

**toString()** - Method in class `jminusminus.Type`

A printable (j--) string representation of this type.

**toString()** - Method in class `jminusminus.TypeName`

Return the Java representation of this type.

**TRUE** - Static variable in interface `jminusminus.JavaCCParserConstants`

**tuples** - Variable in class `jminusminus.NBasicBlock`

List of tuples in this block.

**tuplesToHir()** - Method in class `jminusminus.NControlFlowGraph`

Convert tuples in each block to their high-level (HIR) representations.

**type()** - Method in class `jminusminus.Field`

Return the field's type.

**type()** - Method in interface `jminusminus.IDefn`

The (local variable, formal parameter, or local or imported name) definition's type.

**type** - Variable in class `jminusminus.JExpression`

Expression type

**type()** - Method in class `jminusminus.JExpression`

Return the expression type.

**type()** - Method in class `jminusminus.JFormalParameter`

Return the parameter's type.

**type()** - Method in class `jminusminus.JVariableDeclarator`

Return the variable type.

**type()** - Method in class `jminusminus.LocalVariableDefn`

The type for this variable.

**Type** - Class in `jminusminus`

For representing j-- types.

**Type()** - Constructor for class `jminusminus.Type`

This constructor is to keep the compiler happy.

**type()** - Method in class `jminusminus.TypeNameDefn`

The type for this definition.

**typeFor(Class<?>)** - Static method in class `jminusminus.Type`

Construct a Type representation for a type from its (Java) Class representation.

**typeIndex** - Variable in class `jminusminus.CLAnnotation`

annotation.type\_index item.

**TypeName** - Class in `jminusminus`

Any reference type that can be denoted as a (possibly qualified) identifier.

**TypeName(int, String)** - Constructor for class `jminusminus.TypeName`

Construct an TypeName given its line number, and string spelling out its fully qualified

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

name.

**TypeNameDefn** - Class in `jminusminus`

A definition of a type name.

**TypeNameDefn(Type)** - Constructor for class `jminusminus.TypeNameDefn`

Construct a type name definition given its type.

**typeNameIndex** - Variable in class `jminusminus.CLElementValue`

`element_value.enum_const_value.type_name_index` item.

## U

**UnBoxing** - Class in `jminusminus`

Unboxing requires invoking the appropriate conversion method from the (Java) API.

**UnBoxing(Type, Type, String)** - Constructor for class `jminusminus.UnBoxing`

Construct an UnBoxing converter.

**unescape(String)** - Static method in class `jminusminus.Util`

Unescape the escaped characters in the specified string and return the unescaped string.

**UNIT\_SIZE\_STACK** - Static variable in class `jminusminus.CLConstants`

Stack units for the instructions that set the operand stack to unit size.

**UpdateLineColumn(char)** - Method in class `jminusminus.SimpleCharStream`

**usePositions** - Variable in class `jminusminus.NInterval`

All use positions (in LIR) and their types for this virtual register

**Util** - Class in `jminusminus`

This class defines helper functions.

**Util()** - Constructor for class `jminusminus.Util`

Assignment Project Exam Help

<https://powcoder.com>

## V

Add WeChat powcoder

**V0** - Static variable in class `jminusminus.NPhysicalRegister`

Expression evaluation and results of a function.

**V1** - Static variable in class `jminusminus.NPhysicalRegister`

Expression evaluation and results of a function.

**value** - Variable in class `jminusminus.CLElementValuePair`

`element_value_pairs_table_entry.value` item.

**value** - Variable in class `jminusminus.NHIRASTore`

HIR id of the value to store.

**value** - Variable in class `jminusminus.NHIRIntConstant`

The constant int value.

**value** - Variable in class `jminusminus.NHIRPutField`

HIR id of the value of the field.

**value** - Variable in class `jminusminus.NHIRReturn`

Return value HIR id.

**value** - Variable in class `jminusminus.NHIRStringConstant`

The constant string value.

**value** - Variable in class `jminusminus.NLIRIntConstant`

The constant int value.

**value** - Variable in class `jminusminus.NLIRStringConstant`

The constant string value.

**valueOf(String)** - Static method in enum `jminusminus.CLConstants.Category`

Returns the enum constant of this type with the specified name.



**valueOf(String)** - Static method in enum jminusminus.[InstructionType](#)

Returns the enum constant of this type with the specified name.

**valueOf(String)** - Static method in enum jminusminus.[OffsetFrom](#)

Returns the enum constant of this type with the specified name.

**valueOf(String)** - Static method in enum jminusminus.[TokenKind](#)

Returns the enum constant of this type with the specified name.

**values()** - Static method in enum jminusminus.[CLConstants.Category](#)

Returns an array containing the constants of this enum type, in the order they are declared.

**values** - Variable in class jminusminus.[CLElementValue](#)

element\_value.array\_value.values item.

**values()** - Static method in enum jminusminus.[InstructionType](#)

Returns an array containing the constants of this enum type, in the order they are declared.

**values()** - Static method in enum jminusminus.[OffsetFrom](#)

Returns an array containing the constants of this enum type, in the order they are declared.

**values()** - Static method in enum jminusminus.[TokenKind](#)

Returns an array containing the constants of this enum type, in the order they are declared.

**visited** - Variable in class jminusminus.[NBasicBlock](#)

Has this block been visited?

**VOID** - Static variable in interface jminusminus.[JavaCCParserConstants](#)

**VOID** - Static variable in class jminusminus.[Type](#)

The void type.

**vRegId** - Variable in class jminusminus.[NInterval](#)

The virtual register id corresponding to the index in the array list of NIntervals used by register allocation.

## W

**WHILE** - Static variable in interface jminusminus.[JavaCCParserConstants](#)

**WIDE** - Static variable in class jminusminus.[CLConstants](#)

WIDE instruction.

**WidenReference** - Static variable in interface jminusminus.[Converter](#)

For widening conversion (no run-time code needed).

**write(CLOutputStream)** - Method in class jminusminus.[CLAnnotation](#)

Write the contents of this object to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.[CLAnnotationDefaultAttribute](#)

**write(CLOutputStream)** - Method in class jminusminus.[CLAttributeInfo](#)

Write the contents of this attribute to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.[CLCodeAttribute](#)

**write(CLOutputStream)** - Method in class jminusminus.[CLConstantClassInfo](#)

**write(CLOutputStream)** - Method in class jminusminus.[CLConstantDoubleInfo](#)

**write(CLOutputStream)** - Method in class jminusminus.[CLConstantFloatInfo](#)

**write(CLOutputStream)** - Method in class jminusminus.[CLConstantIntegerInfo](#)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**write(CLOutputStream)** - Method in class jminusminus.CLConstantLongInfo

**write(CLOutputStream)** - Method in class jminusminus.CLConstantMemberRefInfo

**write(CLOutputStream)** - Method in class jminusminus.CLConstantNameAndTypeInfo

**write(CLOutputStream)** - Method in class jminusminus.CLConstantPool

Write the contents of the constant\_pool to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLConstantStringInfo

**write(CLOutputStream)** - Method in class jminusminus.CLConstantUtf8Info

**write(CLOutputStream)** - Method in class jminusminus.CLConstantValueAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLCPInfo

Write the contents of this constant pool item to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLDeprecatedAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLElementValue

Write the contents of this object to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLElementValuePair

Write the contents of this object to the specified output stream.

**write()** - Method in class jminusminus.CLEmitter

Write out the class to the file system as a .class file if toFile is true.

**write(CLOutputStream)** - Method in class jminusminus.CLEnclosingMethodAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLExceptionInfo

Write the contents of this object to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLExceptionsAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLFile

Write the contents of this class to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLInnerClassesAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLInnerClassInfo

Write the contents of this object to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLLineNumberInfo

Write the contents of this object to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLLineNumberTableAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLLocalVariableInfo

Write the contents of this object to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLLocalVariableTableAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLLocalVariableTypeInfo

Write the content of this object to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLLocalVariableTypeTableAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLMemberInfo

Write the contents of this class member to the specified output stream.

**write(CLOutputStream)** - Method in class jminusminus.CLParameterAnnotationInfo

Write the contents of this object to the specified output stream.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**write(CLOutputStream)** - Method in class  
jminusminus.CLRuntimeInvisibleAnnotationsAttribute

**write(CLOutputStream)** - Method in class  
jminusminus.CLRuntimeInvisibleParameterAnnotationsAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLRuntimeVisibleAnnotationsAttribute

**write(CLOutputStream)** - Method in class  
jminusminus.CLRuntimeVisibleParameterAnnotationsAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLSignatureAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLSourceDebugExtensionAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLSourceFileAttribute

**write(CLOutputStream)** - Method in class jminusminus.CLSyntheticAttribute

**write()** - Method in class jminusminus.NEmitter  
Write out SPIM file(s) to the file system.

**write** - Variable in class jminusminus.NLIRInstruction  
Register that stores the result (if any) of this instruction.

**write(int, String, int)** - Static method in class spim.SPIM  
Wrapper for SPIM.write().

**writeHirToStdOut(PrettyPrinter)** - Method in class jminusminus.NBasicBlock  
Write the HIR instructions in this block to STDOUT.

**writeHirToStdOut(PrettyPrinter)** - Method in class jminusminus.NControlFlowGraph  
Write the hir instructions in this cfg to STDOUT.

**writeInt(long)** - Method in class jminusminus.CLOutputStream  
Write four bytes to the output stream to represent the value of the argument.

**writeIntervalsToStdOut(PrettyPrinter)** - Method in class jminusminus.NControlFlowGraph  
Write the intervals in this cfg to STDOUT.

**writeLirToStdOut(PrettyPrinter)** - Method in class jminusminus.NBasicBlock  
Write the LIR instructions in this block to STDOUT.

**writeLirToStdOut(PrettyPrinter)** - Method in class jminusminus.NControlFlowGraph  
Write the lir instructions in this cfg to STDOUT.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLAnnotation  
Write the content of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLAnnotationDefaultAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLAttributeInfo  
Write the contents of this attribute to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLCodeAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantClassInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantDoubleInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantFieldRefInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantFloatInfo

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantIntegerInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantInterfaceMethodInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantLongInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantMethodInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantNameAndTypeInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantPool

Write the contents of the constant pool to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantStringInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantUtf8Info

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLConstantValueAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLCPInfo

Write the content of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLDeprecatedAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLElementValue

Write the content of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLElementValuePair

Write the content of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLEnclosingMethodAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLExceptionInfo

Write the contents of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLExceptionsAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLFieldInfo

**writeToStdOut()** - Method in class jminusminus.CLFile

Write the contents of the class file to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLInnerClassesAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLInnerClassInfo

Write the contents of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLLineNumberInfo

Write the contents of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLLineNumberTableAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLLocalVariableInfo

Write the contents of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLLocalVariableTableAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLLocalVariableTypeInfo

Write the contents of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLLocalVariableTypeTableAttribute

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLMemberInfo

Write the contents of this class member to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLMethodInfo

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLParameterAnnotationInfo

Write the content of this object to STDOUT in a format similar to that of javap.

**writeToStdOut(PrettyPrinter)** - Method in class  
jminusminus.CLRuntimeInvisibleAnnotationsAttribute

**writeToStdOut(PrettyPrinter)** - Method in class  
jminusminus.CLRuntimeInvisibleParameterAnnotationsAttribute

**writeToStdOut(PrettyPrinter)** - Method in class  
jminusminus.CLRuntimeVisibleAnnotationsAttribute

**writeToStdOut(PrettyPrinter)** - Method in class  
jminusminus.CLRuntimeVisibleParameterAnnotationsAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLSignatureAttribute

**writeToStdOut(PrettyPrinter)** - Method in class  
jminusminus.CLSourceDebugExtensionAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLSourceFileAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CLSyntheticAttribute

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.CompilationUnitContext

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.Context

Write the contents of this context to STDOUT.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JArrayExpression

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JArrayInitializer

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JAST

Write the information pertaining to this AST to STDOUT.

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JBinaryExpression

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JBlock

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JCastOp

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JClassDeclaration

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JCompilationUnit

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JConstructorDeclaration

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JEmptyStatement

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JFieldDeclaration

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JFieldSelection

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JFormalParameter

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JIfStatement

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JInstanceOfOp

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JLiteralChar

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JLiteralFalse

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JLiteralInt

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JLiteralNull

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JLiteralString

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JLiteralTrue

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JMessageExpression

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JMethodDeclaration

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JNewArrayOp

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JNewOp

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JReturnStatement

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JStatementExpression

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JSuper

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JSuperConstruction

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JThis  
inheritDoc

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JThisConstruction

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JUnaryExpression

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JVariable

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JVariableDeclaration

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JVariableDeclarator

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JWhileStatement

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.JWildExpression

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.LocalContext

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.MethodContext

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.NInterval

Write the interval information to STDOUT.

**Assignment Project Exam Help**

<https://powcoder.com>

**Add WeChat powcoder**

**writeToStdOut(PrettyPrinter)** - Method in class jminusminus.NTuple

Write the information pertaining to this tuple to STDOUT.

**writeTuplesToStdOut(PrettyPrinter)** - Method in class jminusminus.NBasicBlock

Write the tuples in this block to STDOUT.

**writeTuplesToStdOut(PrettyPrinter)** - Method in class jminusminus.NControlFlowGraph

Write the tuples in this cfg to STDOUT.

## Z

**ZERO** - Static variable in class jminusminus.NPhysicalRegister

Constant 0.

A B C D E F G H I J K L M N O P R S T U V W Z

- Prev
- Next
- Frames
- No Frames
- All Classes
- Help

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder