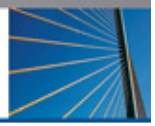
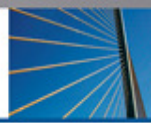
Calling Java

JNIPort for VisualWorks



Agenda

- JNIPort
- Java Native Interface – Invocation Interface
- JNIPort Implementation
  - The Low Level Interface
  - The Twilight Zone
  - Ghost classes
- Tools
- Plans



JNIPort

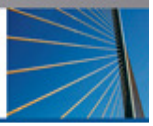
Use any Java classes with any Java VM in Smalltalk

Free open source class library

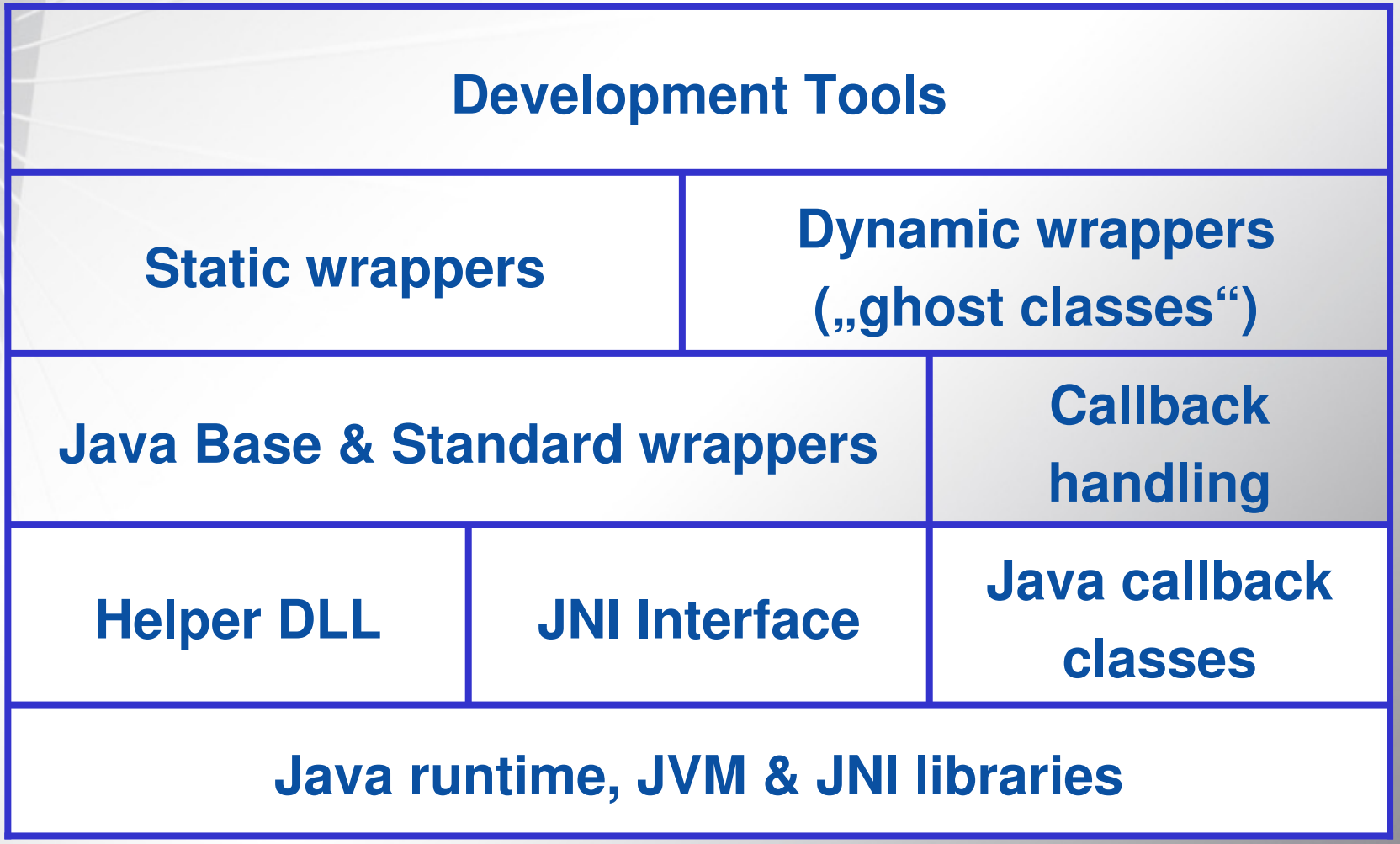
Developed by Chris Uppal for Dolphin Smalltalk

Ported to VisualWorks in 2006/2007

```
| jvm class |
jvm := JVM current.
class := jvm findClass: #'java.lang.System'.
class currentTimeMillis_null. "--> 1045217556089"
```

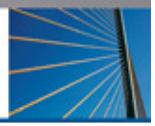



JNIPort components



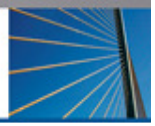
JNI – Invocation Interface

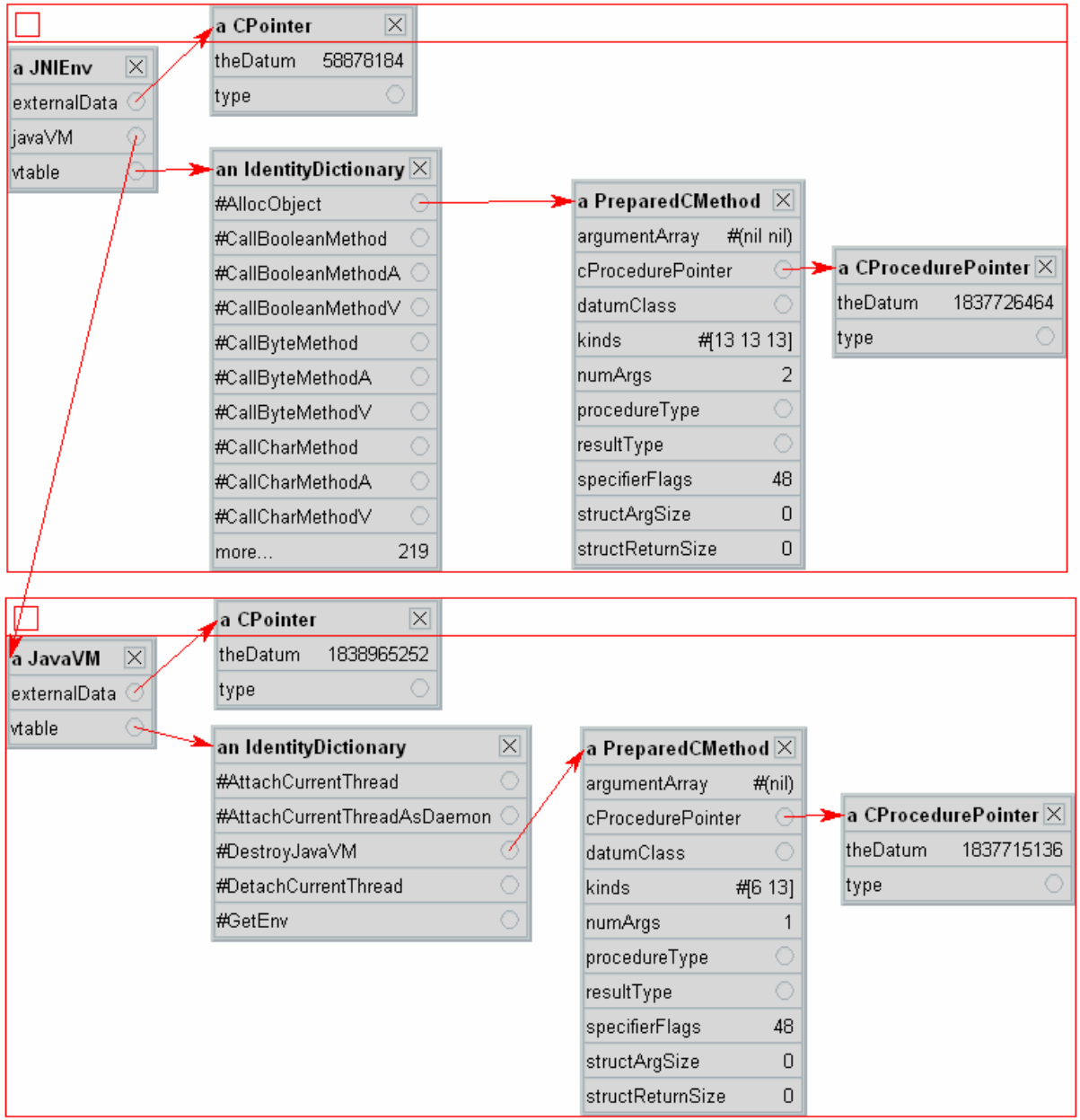
- The Java VM is a library, not an executable
- Function tables (vtable): JavaVM, JNIEnv
- JavaVM – Start / stop the Java VM
- JNIEnv – Access classes and instances
  - Look up Java classes
  - Send messages to Java classes and instances
  - Access Java objects using reference objects
    - Local references – valid inside a thread
    - Global references – valid in all threads



Lowest Level API

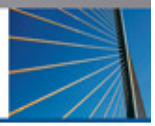
```
jniEnv :=  JNILibrary new createFirstJNIEnv: JavaVMInitArgs new.

math := jniEnv FindClass_name: 'java/lang/Math'
                    onException: [:error | "error handling"].

absID := jniEnv GetStaticMethodID_class: math
                    name: 'abs' sig: '(D)D'
                    onException: [:error | "error handling"].

arguments := JNIValueArray fromArray: #(-321.2d) types: #(jdouble).

result := jniEnv CallStaticDoubleMethodA_class: math
                    methodID: absID args: arguments
                    onException: [:error | "error handling"].

env javaVM DestroyJavaVM.
```

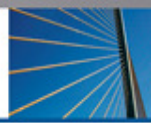

a JNIEnv
externalData
javaVM
vtable
a CPointer
theDatum 58878184
type
an IdentityDictionary
#AllocObject
#CallBooleanMethod
#CallBooleanMethodA
#CallBooleanMethodV
#CallByteMethod
#CallByteMethodA
#CallByteMethodV
#CallCharMethod
#CallCharMethodA
#CallCharMethodV
more... 219
a PreparedCMethod
argumentArray #(nil nil)
cProcedurePointer
datumClass
kinds #[13 13 13]
numArgs 2
procedureType
resultType
specifierFlags 48
structArgSize 0
structReturnSize 0
a CProcedurePointer
theDatum 1837726464
type
a JavaVM
externalData
vtable
a CPointer
theDatum 1838965252
type
an IdentityDictionary
#AttachCurrentThread
#AttachCurrentThreadAsDaemon
#DestroyJavaVM
#DetachCurrentThread
#GetEnv
a PreparedCMethod
argumentArray #(nil)
cProcedurePointer
datumClass
kinds #[6 13]
numArgs 1
procedureType
resultType
specifierFlags 48
structArgSize 0
structReturnSize 0
a CProcedurePointer
theDatum 1837715136
type

The Twilight Zone

- References are automatically encapsulated by Smalltalk objects
- Statics, Java Instances
- Messages are sent to Statics and Instances
  - No need to talk to the JNIEnv
  - But still low level
- Automatic Life Cycle Management (GC)
- Access to Java Reflection

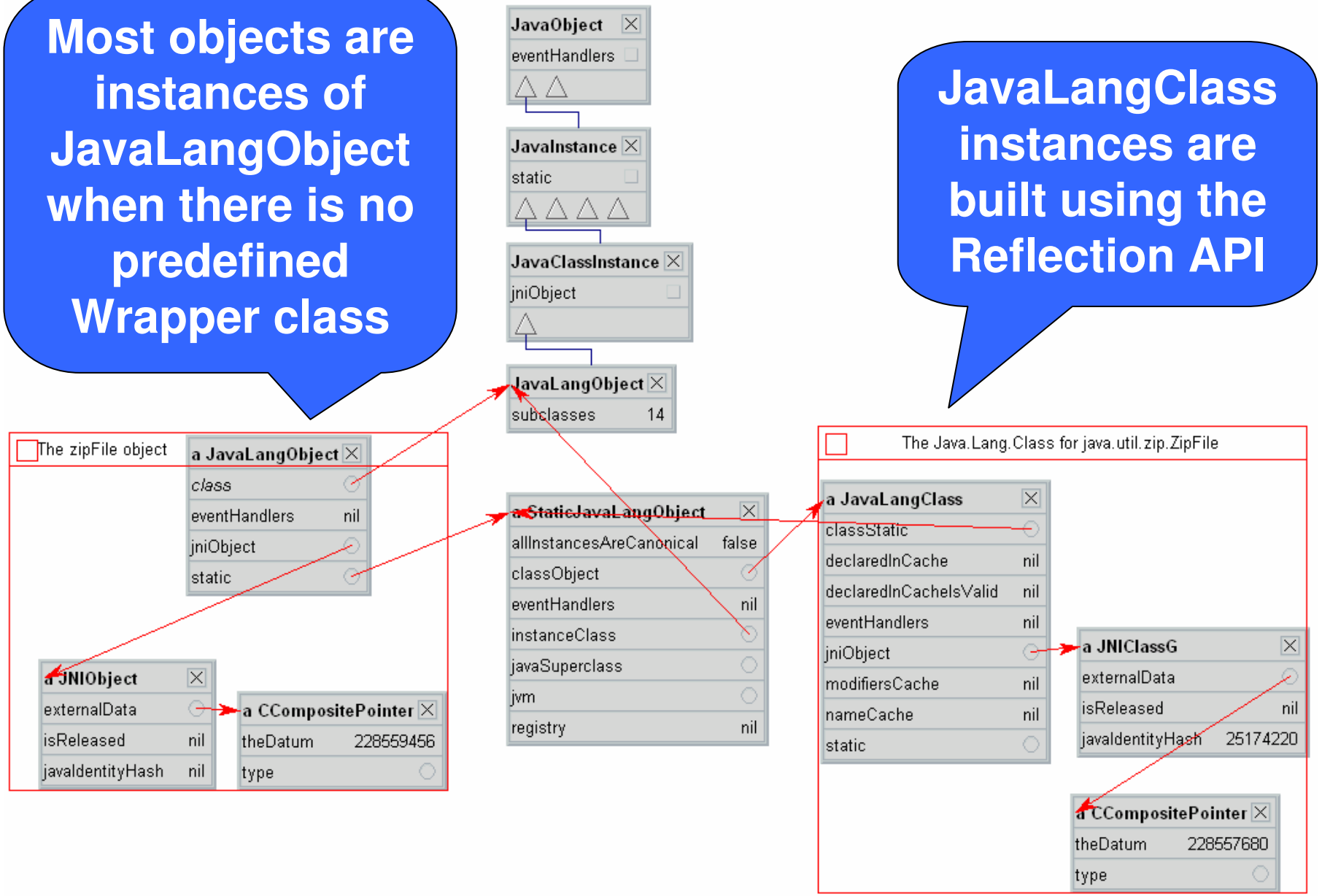


```
jvm := JVM current.
zipfileClass := jvm findClass: #'java.util.zip.ZipFile'.
args := (JNIValueArray new: 1).
args objectAt: 1 put: (‘file.zip‘ asJavaString: jvm).
zipfile := zipfileClass
    callConstructorSignature: '(Ljava/lang/String;)V'
    withArguments: args.
zipfile callIntMethod: 'size'.          "--> 6"
entries := zipfile
    callObjectMethod: 'entries'
    signature: '()Ljava/util/Enumeration;'.
```

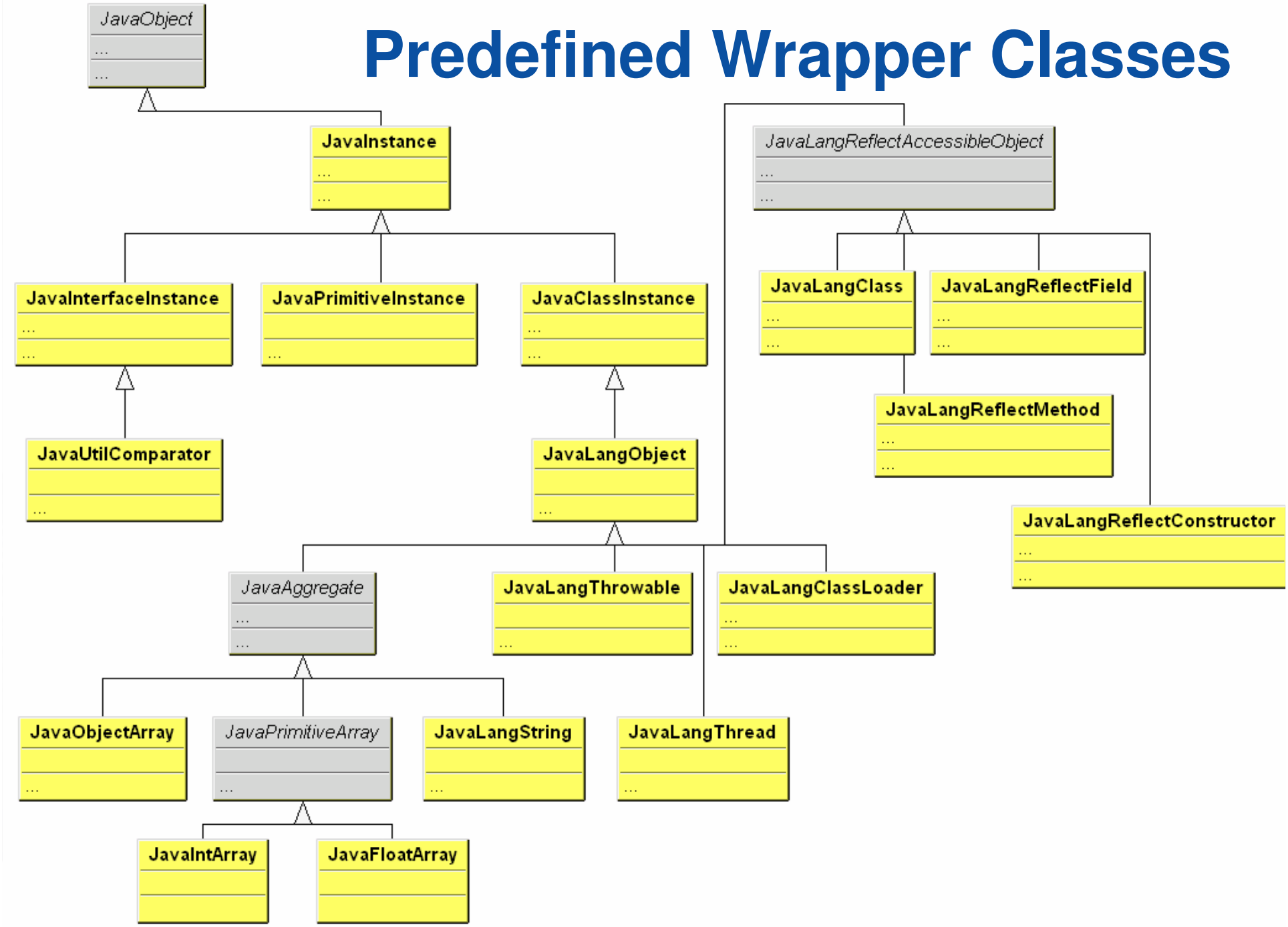



Most objects are instances of JavaLangObject when there is no predefined Wrapper class

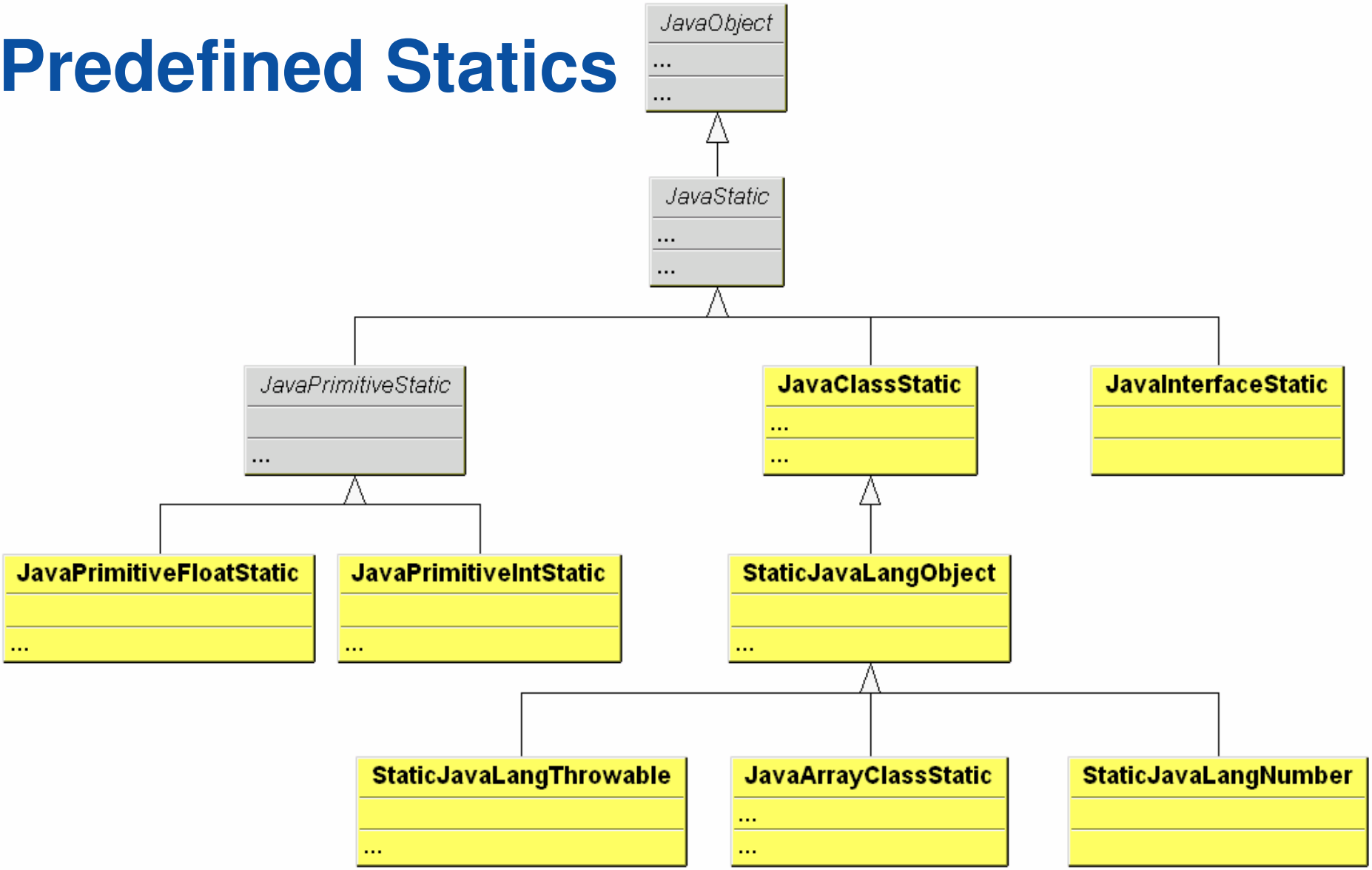
JavaLangClass instances are built using the Reflection API



Predefined Wrapper Classes



Predefined Statics



Ghost classes

Use Java objects just like Smalltalk objects

Dynamically generated…

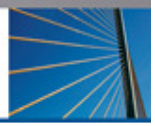
- …wrapper classes
- …wrapper methods

Code generation triggered by creating the first reference to an instance of a Java class

~~Ghost classes disappear when they have no instances~~

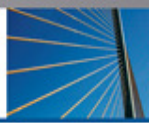
Ghost methods

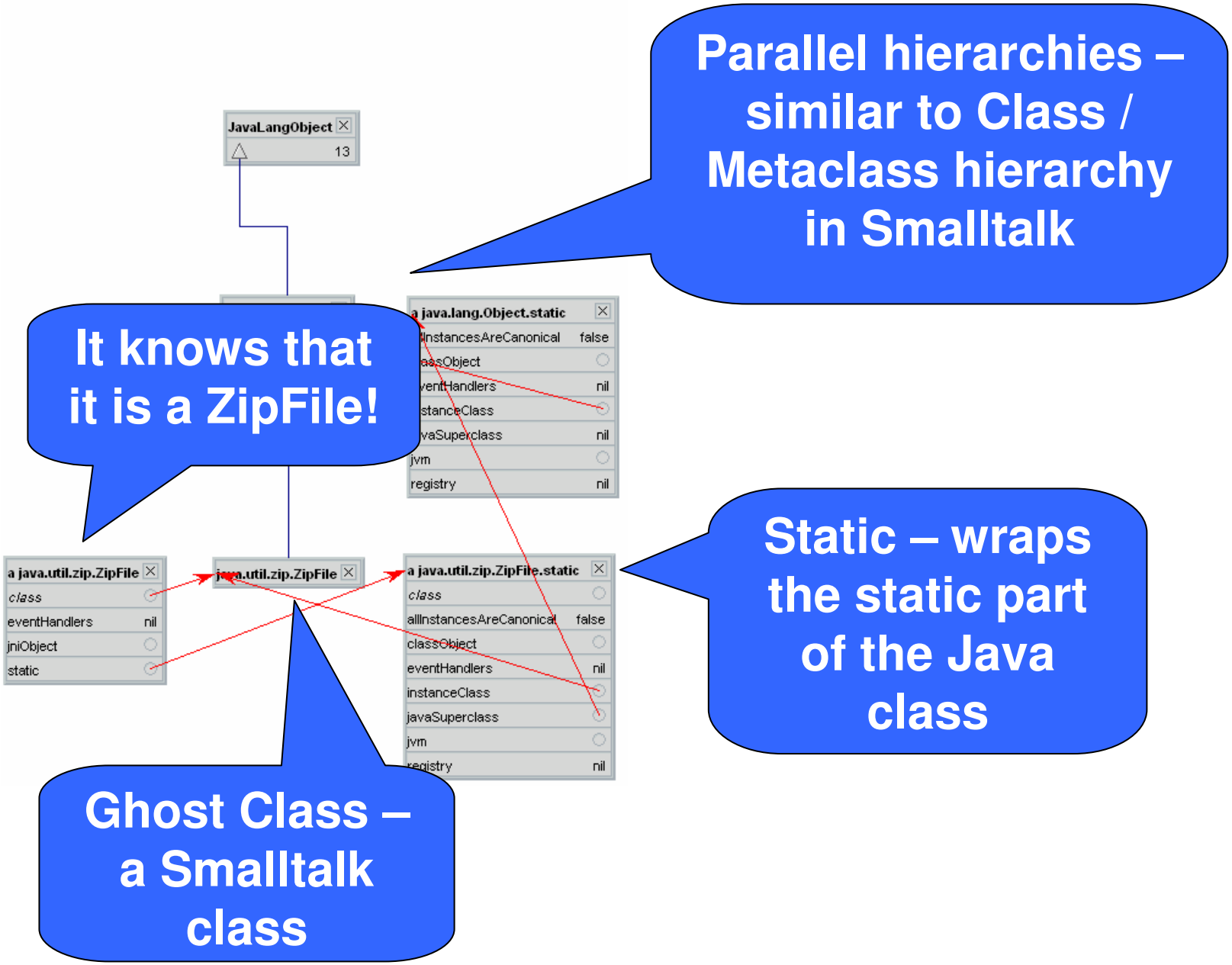
- Source code generated using information from Reflection API, querying a Java class' protocol
- Context specific information embedded into CompiledCode as „literals“
- Source code is discarded
  - Can be kept in image (JNIPort configuration option)
- Augmented tools to handle ghost methods

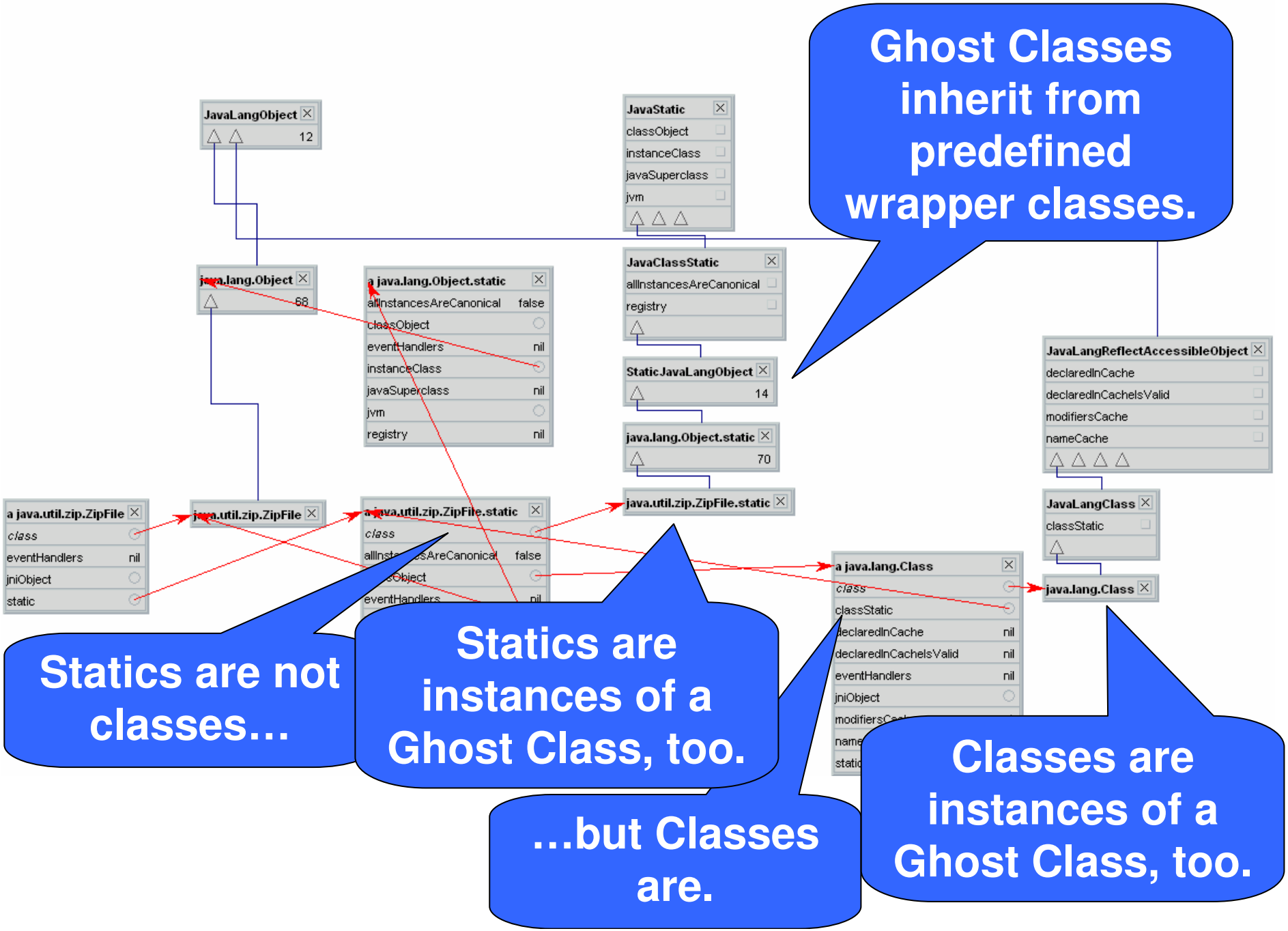


```
jvm := JVM current.

zipfileClass := jvm findClass: #'java.util.zip.ZipFile'.

zipfile := zipfileClass new_String: 'MyZipFile.zip'.

zipfile size_null.

entries := zipfile entries_null.

entries asAnEnumeration
    do: [:each | Transcript cr; print: each].
```

No need to implement these methods – they are generated on the fly.



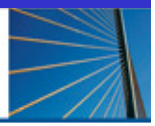

Parallel hierarchies – similar to Class / Metaclass hierarchy in Smalltalk
It knows that it is a ZipFile!
Static – wraps the static part of the Java class
Ghost Class – a Smalltalk class
JavaLangObject
13
a java.lang.Object.static
allInstancesAreCanonical false
classObject
eventHandlers nil
instanceClass
javaSuperclass nil
jvm
registry nil
a java.util.zip.ZipFile
class
eventHandlers nil
jniObject
static
java.util.zip.ZipFile
a java.util.zip.ZipFile.static
class
allInstancesAreCanonical false
classObject
eventHandlers nil
instanceClass
javaSuperclass
jvm
registry nil


JavaLangObject
12
JavaStatic
classObject
instanceClass
javaSuperclass
jvm
Ghost Classes inherit from predefined wrapper classes.
java.lang.Object
68
a java.lang.Object.static
allInstancesAreCanonical false
classObject
eventHandlers nil
instanceClass
javaSuperclass nil
jvm
registry nil
JavaClassStatic
allInstancesAreCanonical
registry
StaticJavaLangObject
14
java.lang.Object.static
70
JavaLangReflectAccessibleObject
declaredInCache
declaredInCacheIsValid
modifiersCache
nameCache
a java.util.zip.ZipFile
class
eventHandlers nil
jniObject
static
java.util.zip.ZipFile
java.util.zip.ZipFile.static
class
eventHandlers
java.util.zip.ZipFile.static
JavaLangClass
classStatic
a java.lang.Class
class
classStatic
declaredInCache nil
declaredInCacheIsValid nil
eventHandlers nil
jniObject
java.lang.Class
Statics are not classes…
Statics are instances of a Ghost Class, too.
…but Classes are.
Classes are instances of a Ghost Class, too.

Don‘t worry – you don‘t have to know that.

Just write your code.

JNIPort does the rest.



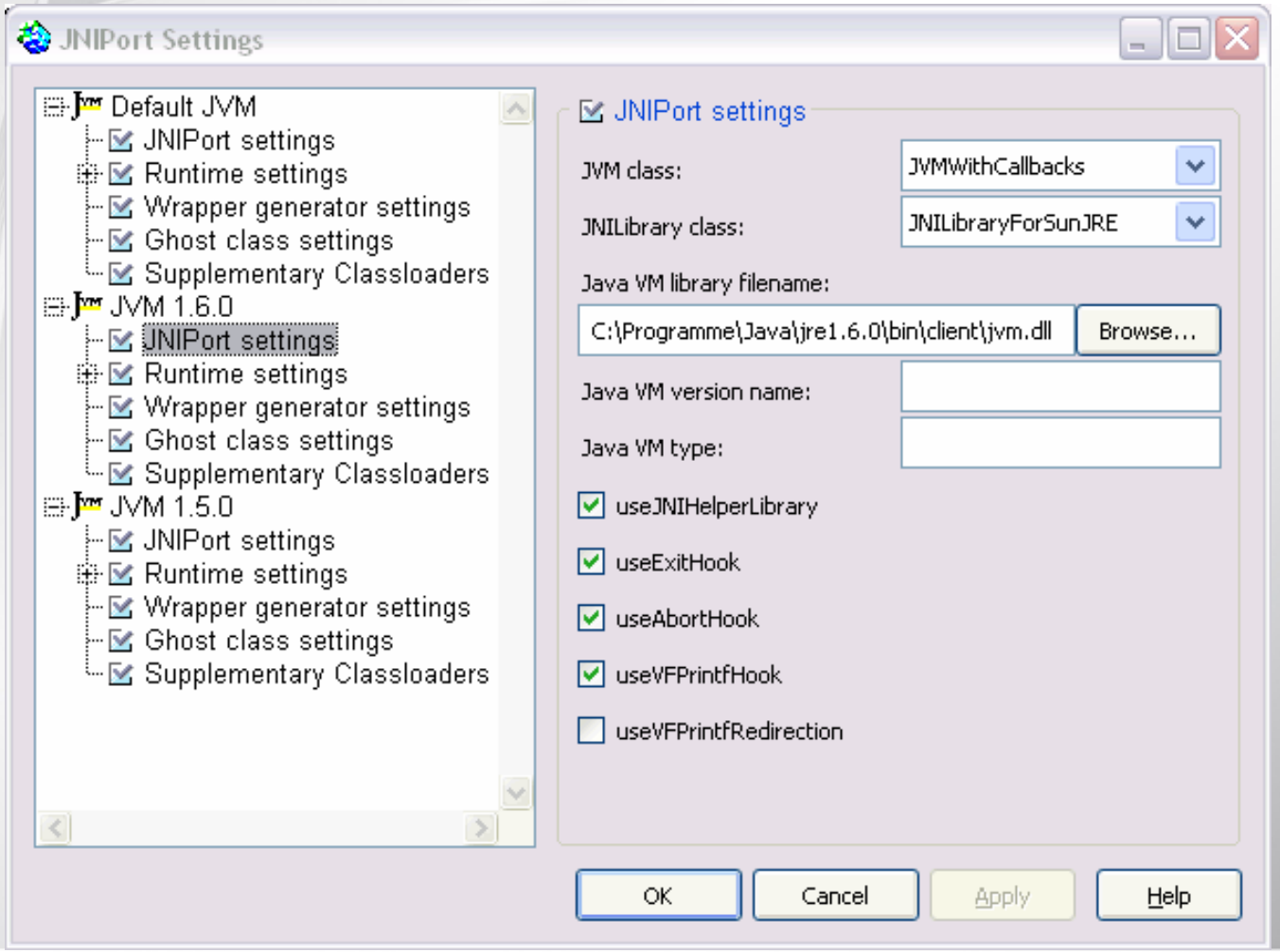
Tools

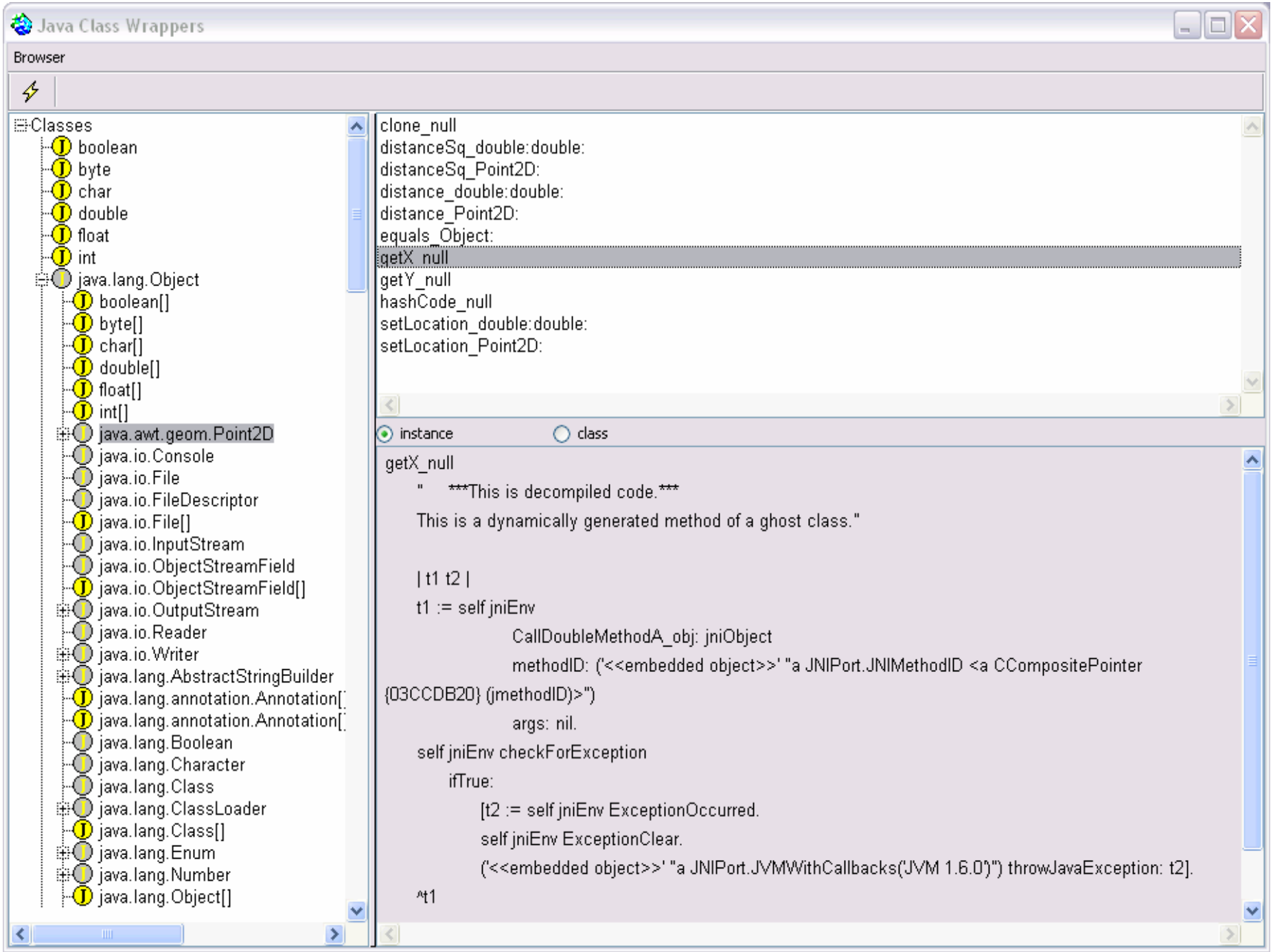
Settings Tool

Class Wrapper Browser

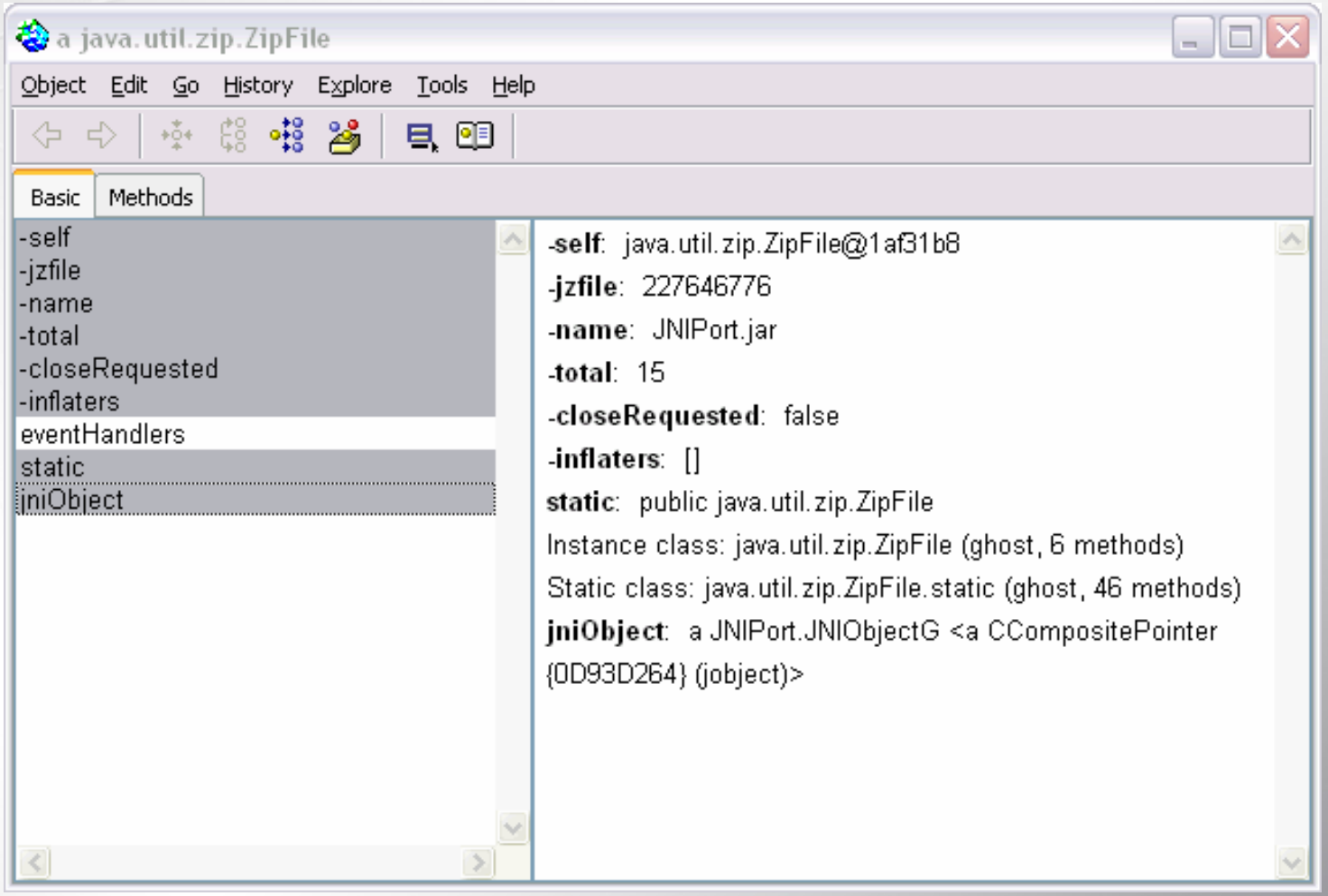
Inspector

Decompiler




Java Class Wrappers
Browser
Classes
boolean
byte
char
double
float
int
java.lang.Object
boolean[]
byte[]
char[]
double[]
float[]
int[]
java.awt.geom.Point2D
java.io.Console
java.io.File
java.io.FileDescriptor
java.io.File[]
java.io.InputStream
java.io.ObjectStreamField
java.io.ObjectStreamField[]
java.io.OutputStream
java.io.Reader
java.io.Writer
java.lang.AbstractStringBuilder
java.lang.annotation.Annotation[
java.lang.annotation.Annotation[
java.lang.Boolean
java.lang.Character
java.lang.Class
java.lang.ClassLoader
java.lang.Class[]
java.lang.Enum
java.lang.Number
java.lang.Object[]
clone_null
distanceSq_double:double:
distanceSq_Point2D:
distance_double:double:
distance_Point2D:
equals_Object:
getX_null
getY_null
hashCode_null
setLocation_double:double:
setLocation_Point2D:
instance
class
getX_null
" ***This is decompiled code.***
This is a dynamically generated method of a ghost class."
| t1 t2 |
t1 := self jniEnv
CallDoubleMethodA_obj: jniObject
methodID: ('<<embedded object>>' "a JNIPort.JNIMethodID <a CCompositePointer
{03CCDB20} (jmethodID)>")
args: nil.
self jniEnv checkForException
ifTrue:
[t2 := self jniEnv ExceptionOccurred.
self jniEnv ExceptionClear.
('<<embedded object>>' "a JNIPort.JVMWithCallbacks('JVM 1.6.0')") throwJavaException: t2].
^t1

Inspector


a java.util.zip.ZipFile
Object Edit Go History Explore Tools Help
Basic Methods
-self
-jzfile
-name
-total
-closeRequested
-inflaters
eventHandlers
static
jniObject
-self: java.util.zip.ZipFile@1af31b8
-jzfile: 227646776
-name: JNIPort.jar
-total: 15
-closeRequested: false
-inflaters: []
static: public java.util.zip.ZipFile
Instance class: java.util.zip.ZipFile (ghost, 6 methods)
Static class: java.util.zip.ZipFile.static (ghost, 46 methods)
jniObject: a JNIPort.JNIObjectG <a CCompositePointer {0D93D264} (jobject)>

Plans

Tools for generating static wrapper classes

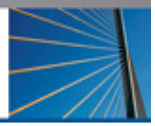
Linux, MacOS X

Java packages as Smalltalk Namespaces:

```
java.lang.System currentTimeMillis_null.
```

instead of

```
class := JVM current findClass:
   #'java.lang.System'.
class currentTimeMillis_null.
```

Resources

- Cincom Public Repository
  - Registry (version 16 or later)
  - FastCMethodPointers (version 1.1 or later)
  - Weaklings (version 12 or later)
  - JNIPort Prerequisites
  - JNIPort
  - JNIPort Tools
  - JNIPort Tests
- Chris Uppal's web site: http://www.metagnostic.org
  - Documentation:
    http://www.metagnostic.org/DolphinSmalltalk/JNIPort.html
  - Extras directory in
    http://www.metagnostic.org/DolphinSmalltalk/JNIPort-Complete.zip