

When a Virtual Machine is not Complex Enough

Claus Gittinger

Marcel Hlopko

Jan Kurš

Jan Vraný

What is STX:LIBJAVA

A Java
implementation for
Smalltalk/X

Smalltalk/X

What is Smalltalk/X??

- Fast smalltalk implementation

Smalltalk/X

What is Smalltalk/X??

- Fast smalltalk implementation
- VM written in C, including JIT

Smalltalk/X

What is Smalltalk/X??

- Fast smalltalk implementation
- VM written in C, including JIT
- Support for Ruby, Javascript, XQuery, Pascal

Why yet another language?

- different approach

Why yet another language?

- different approach
- possibility to reuse existing code

Why yet another language?

- different approach
- possibility to reuse existing code
- interop research

Why yet another language?

- different approach
- possibility to reuse existing code
- interop research

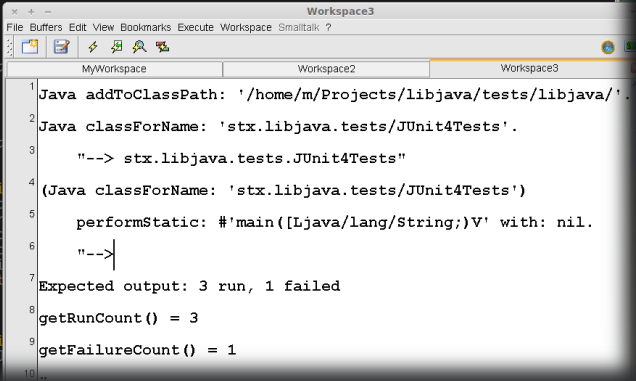
and it's fun :)

Hands on - JUnit

Hands on - JUnit

```
36
35 public class JUnit4Tests {
34
33     @Test
32     public void helloWorld() {
31         assertEquals("Is zero equal 5-5 test",0, (5-5));
30     }
29
28     @Test
27     public void assertEquals() {
26         assertEquals(0, (5-5));
25     }
24
23     @Test
22     public void assertEquals() {
21         assertEquals(0, (5-5));
20     }
19
18     public static void main(String[] args) {
17         System.out.println("JUnit4Tests");
16         Result r = new Result();
15         System.out.println(r);
14         System.out.println(r);
13         for (Failure f : r.getFailures()) {
12             System.out.println(f.getMessage());
11         }
10     }
9
8     public static void main(String[] args) {
7         return 0;
6     }

```

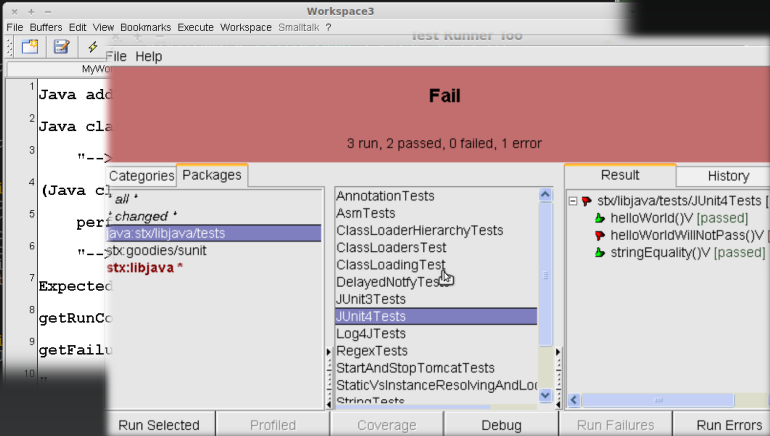


```
Workspace3
File Buffers Edit View Bookmarks Execute Workspace Smalltalk ?
MyWorkspace Workspace2 Workspace3
1 Java addToClassPath: '/home/m/Projects/libjava/tests/libjava/'.
2 Java classForName: 'stx.libjava.tests/JUnit4Tests'.
3     "--> stx.libjava.tests.JUnit4Tests"
4 (Java classForName: 'stx.libjava.tests/JUnit4Tests')
5     performStatic: #'main([Ljava/lang/String;)V' with: nil.
6     "-->
7 Expected output: 3 run, 1 failed
8 getRunCount() = 3
9 getFailureCount() = 1
10

```

Hands on - JUnit

```
36
35 public class JUnit4Tests {
34
33     @Test
32     public void helloWorld() {
31         assertEquals("Is zero equal 5-5 test",0, (5-5));
30     }
29
28     @Test
27     public void assertEquals() {
26         assertEquals(0, (5-5));
25     }
24
23     @Test
22     public void assertEquals() {
21         assertEquals(0, (5-5));
20     }
19
18     public static void main(String[] args) {
17         System.out.println("JUnit4Tests");
16         Result r = new Result();
15         System.out.println(r);
14         System.out.println(r);
13         for (Fail f : r.getFailures()) {
12             System.out.println(f);
11         }
10     }
9
8     public static void main(String[] args) {
7         return 0;
6     }
}
```



Hands on - JUnit

That's it??



Hands on - JUnit

That's it??

Not quite yet



How it works?

- loads Java `.class` files

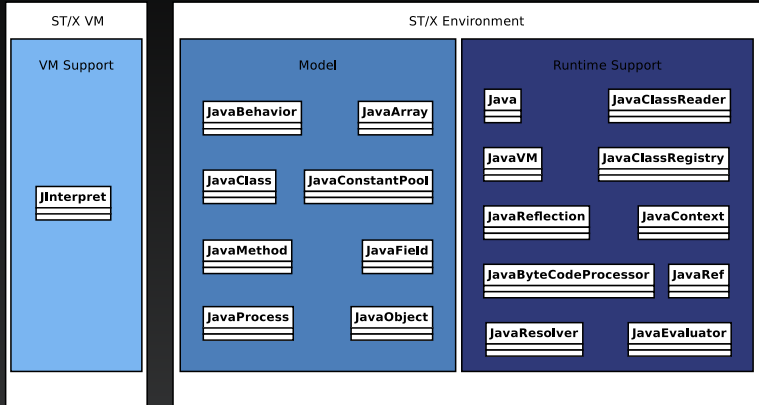
How it works?

- loads Java `.class` files
- executes Java bytecodes (no translation)

How it works?

- loads Java `.class` files
- executes Java bytecodes (no translation)
- no difference between Java and Smalltalk objects

High level overview



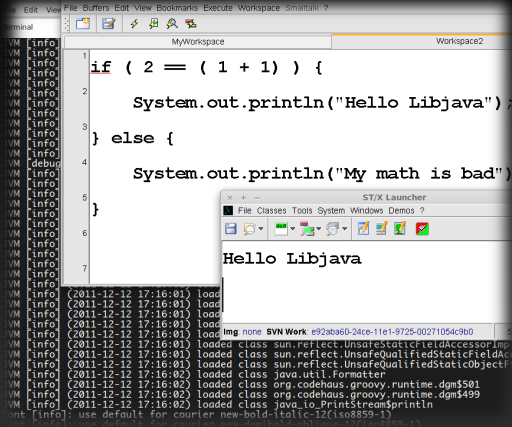
Hands on - Groovy



Hands on - Groovy

[illegible]

Hands on - Groovy



The screenshot shows an IDE with a Groovy script in a file named 'MyWorkspace'. The code is as follows:

```
1 if ( 2 == ( 1 + 1 ) ) {  
2     System.out.println("Hello Libjava");  
3 } else {  
4     System.out.println("My math is bad")  
5 }  
6  
7
```

The output window shows the result of the execution:

```
Hello Libjava
```

The console window shows the following output:

```
(2011-12-12 17:16:01) load  
(2011-12-12 17:16:01) load  
(2011-12-12 17:16:01) load  
(2011-12-12 17:16:01) load  
(2011-12-12 17:16:01) load  
(2011-12-12 17:16:01) loaded class sun.reflect.UnsafeStaticFieldAccessorImpl  
(2011-12-12 17:16:01) loaded class sun.reflect.UnsafeQualifiedStaticFieldAcc  
(2011-12-12 17:16:01) loaded class sun.reflect.UnsafeQualifiedStaticObjectF  
(2011-12-12 17:16:02) loaded class java.util.Formatter  
(2011-12-12 17:16:02) loaded class org.codehaus.groovy.runtime.dgm$501  
(2011-12-12 17:16:02) loaded class org.codehaus.groovy.runtime.dgm$499  
(2011-12-12 17:16:02) loaded class java.io.PrintStream$Println  
(2011-12-12 17:16:02) use default for courier new-bold-italic-12(iso8859-1)
```



Ok, better, but still..

Obstacles on the way

- Class Loaders



Obstacles on the way

- Class Loaders
- Synchronization



Obstacles on the way

- Class Loaders
- Synchronization
- Exceptions



Obstacles on the way

- Class Loaders
- Synchronization
- Exceptions
- finally



Hands on - Tomcat



Hands on - Tomcat



The screenshot shows a web browser window with the title "Apache Tomcat". The address bar displays "swing.fit.cvut.cz:8085". The browser's bookmark bar contains several folders: "ril", "VM Spec", "to_watch", "st", "ebee", "guitar", "ww", "lisp", "cs", and "katas". The main content area of the page features the Apache Tomcat logo (a yellow cat) on the left, followed by the text "Apache Tomcat & stX:libJava" in the center. Below this, there is a sidebar on the left with two sections: "Administration" containing links for "Status" and "Tomcat Manager", and "Documentation" containing a link for "Release Notes". The main text area on the right contains the following text: "If you're seeing this page via a web browser, it", "As you may have guessed by now, this is the defa", a code snippet "\$CATALINA_HOME/webapps/ROOT/", and a paragraph starting with "where \"\$CATALINA_HOME\" is the root of the Tom" and "don't think you should be, then you're either a user" and "administrator who hasn't got his/her setup quite rig".

Apache Tomcat

stX:libJava

Administration

- [Status](#)
- [Tomcat Manager](#)

Documentation

- [Release Notes](#)

If you're seeing this page via a web browser, it

As you may have guessed by now, this is the defa

`$CATALINA_HOME/webapps/ROOT/`

where "\$CATALINA_HOME" is the root of the Tom

don't think you should be, then you're either a user

administrator who hasn't got his/her setup quite rig

Future work

- JIT



Future work

- JIT
- Integration



Future work

- JIT
- Integration
- Incremental compiler



Future work

- JIT
- Integration
- Incremental compiler

but mostly

interoperability



Questions

Q & A

