

# **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

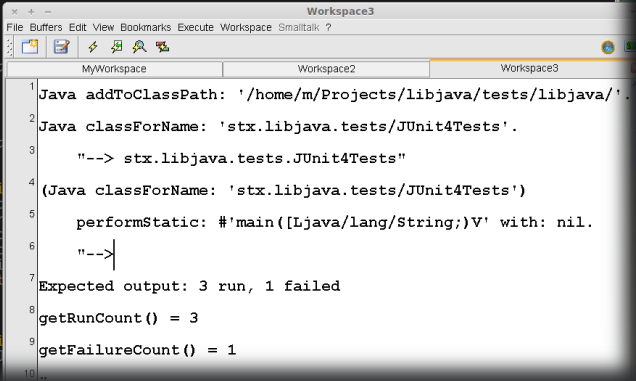
**and it's fun :)**

# 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

The screenshot shows an IDE with a Java file named `JUnit4Tests`. The code contains several JUnit test methods. A file explorer window is open, showing a directory structure with a red overlay. The explorer shows a tree view with 'Categories' and 'Packages' tabs. The 'Categories' tab is active, showing a list of categories. The 'Packages' tab is also visible. The 'Categories' list includes 'all', 'changed', 'java:stx/libjava/tests', 'stx:goodies/sunit', and 'stx:libjava'. The 'Packages' list includes 'An', 'As', 'Cl', 'Cl', 'Cl', 'De', 'JU', 'JU', 'Lo', 'Re', 'St'.

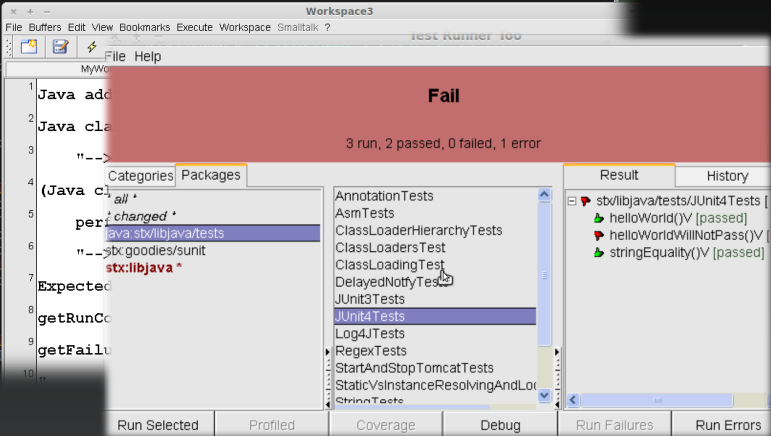
```

35 public class JUnit4Tests {
36
37     @Test
38     public void helloWorld() {
39         assertEquals("Is zero equal 5-5 test",0, (5-5));
40     }
41
42     @Test
43     public void assertEquals() {
44         assertEquals(0, (5-5));
45     }
46
47     @Test
48     public void assertEquals() {
49         assertEquals(0, (5-5));
50     }
51
52     public static void main(String[] args) {
53         System.out.println("Hello World");
54         Result r = System.out.println("Hello World");
55         System.out.println("Hello World");
56         for (Fail f : System.out.println("Hello World")) {
57             System.out.println(f);
58         }
59     }
60
61     public static void main(String[] args) {
62         return 0;
63     }
64 }

```

File Explorer Window:

- File Buffers Edit View Bookmarks Execute Workspace Smaltalk ?
- File Help
- MyWork
- 1 Java add
- 2 Java cla
- 3 "-->
- 4 (Java cl
- 5 all \*
- 6 changed \*
- 7 java:stx/libjava/tests
- 8 stx:goodies/sunit
- 9 stx:libjava \*
- 10 Expected
- 11 getRunCo
- 12 getFailu
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64



# 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?

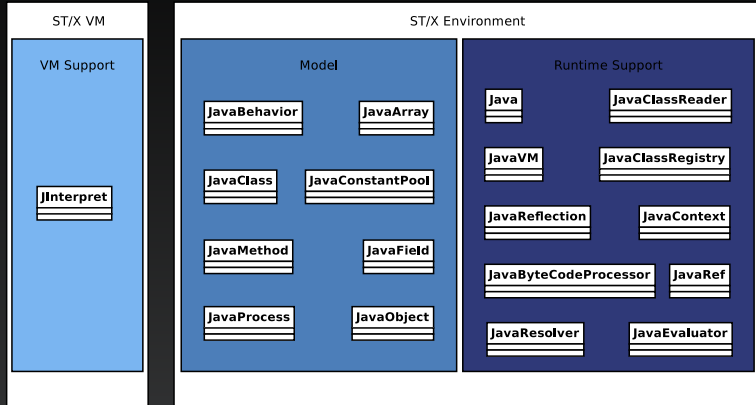
- loads Java `.class` files
- executes Java bytecodes (no translation)



# 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



The screenshot shows an IDE with a Java 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 IDE's output console shows the following output:

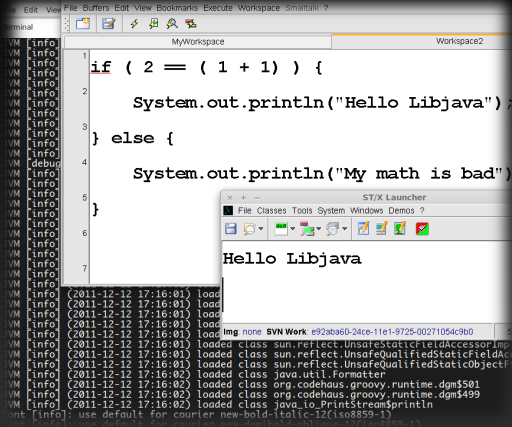
```

Hello Libjava

```

The console also shows a list of loaded classes, including `java.lang.Object`, `java.lang.String`, `java.util.Formatter`, `org.codehaus.groovy.runtime.dgm$501`, `org.codehaus.groovy.runtime.dgm$499`, and `java.io.PrintStream`.

# Hands on - Groovy



The screenshot shows an IDE with a Groovy script in the editor. The script contains a conditional statement that prints "Hello Libjava" if 2 equals 1+1, and "My math is bad" otherwise. The console output shows "Hello Libjava". The background of the IDE window is filled with a repeating pattern of the word "VM" and "info" followed by timestamps and class loading information.

```
1 if ( 2 == ( 1 + 1 ) ) {  
2     System.out.println("Hello Libjava");  
3 } else {  
4     System.out.println("My math is bad")  
5 }  
6  
7  
Hello Libjava  
(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  
ont [info]: 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 a yellow background containing links: "Administration", "Status", "Tomcat Manager", "Documentation", and "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", "\$CATALINA\_HOME/webapps/ROOT/", and "where \"\$CATALINA\_HOME\" is the root of the Tom". The text is partially cut off on the right side.

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

