

# ST/X Libjava

Marcel Hlopko  
Ing. Jan Vraný, Ph.D.  
Ing. Zdeněk Troníček, Ph.D.

January 24, 2012

# What is STX:LIBJAVA

A Java  
implementation for  
Smalltalk/X VM

# What does it mean?

System with ability:

# What does it mean?

System with ability:

- to run Java code

# What does it mean?

System with ability:

- to run Java code
- to call Java methods from Smalltalk

# What does it mean?

System with ability:

- to run Java code
- to call Java methods from Smalltalk
- to call Smalltalk methods from Java

# 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
- interoperability research

# How it works?

- loads Java `.class` files

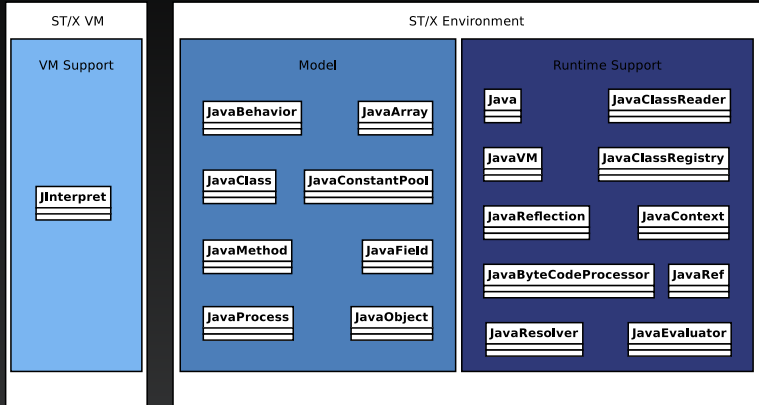
# How it works?

- 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





# Obstacles on the way



# 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

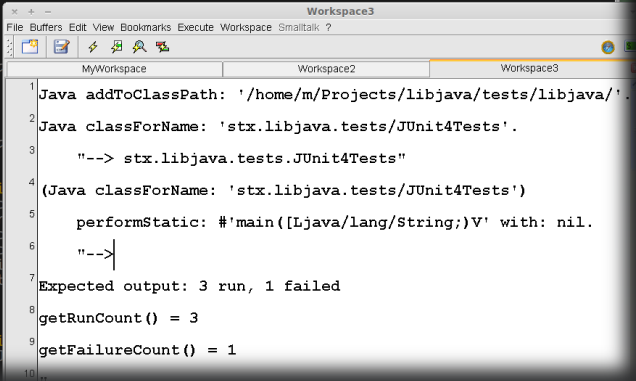
- Class Loaders
- Synchronization
- Exceptions
- finally



# Hands on - JUnit

# Hands on - JUnit

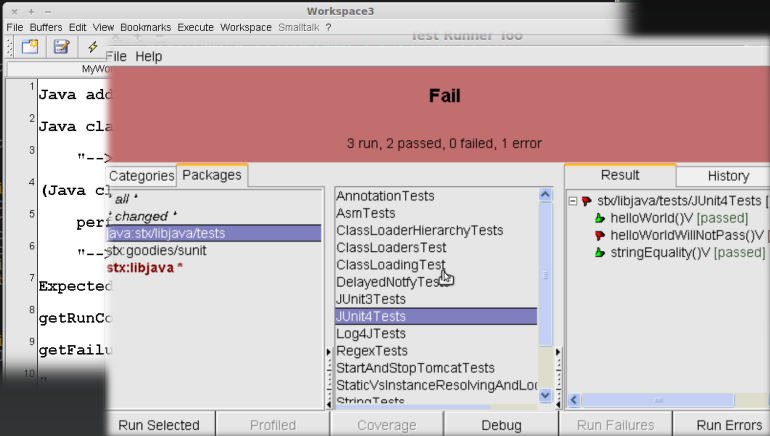
```
36
35 public class JUnit4Tests {
34
33     @Test
32     public void helloWorld() {
31         assertEquals("Is zero equal 5-5 test",0, (5-5));
30     }
29
28     @Test
27     public void assertEquals() {
26         assertEquals(0, (5-5));
25     }
24
23     @Test
22     public void assertEquals()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





# Hands on - Groovy



# Hands on - Groovy

[illegible]

# 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" and "& 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

