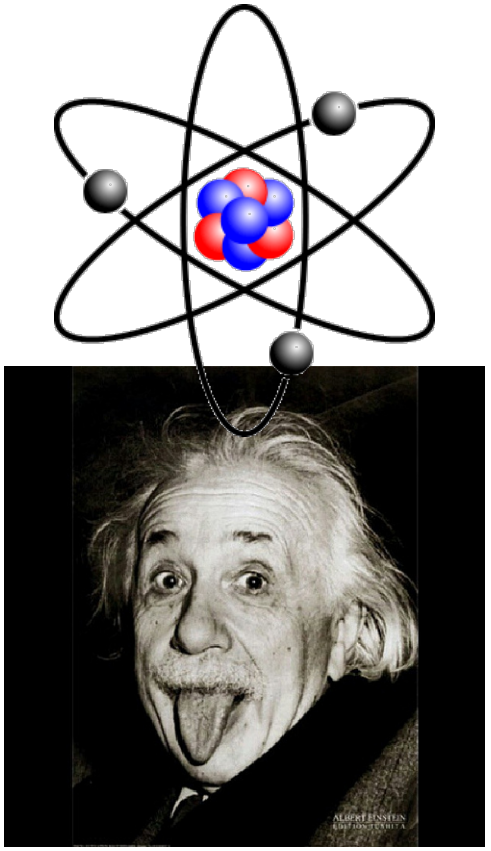
Teaching with NetBeans

NetBeans Day 2015
Munich

Karsten Sitterberg

Introducing – Myself

Physicist



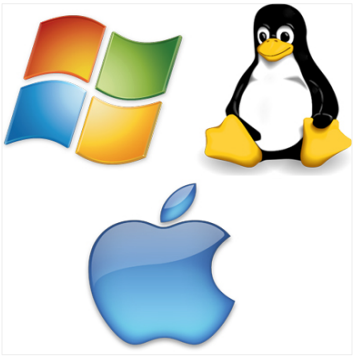
Freelance

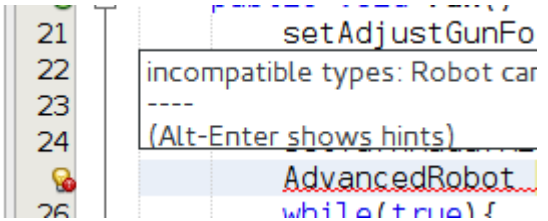
Certified since 2009
Currently
SCJA, OCJP, OCJD

Why Netbeans for teaching?

Thats Why!

- Free/Open source
- Available on all major platforms
- Fast error recognition
- Simple: Install, start, get to work
- Everything included

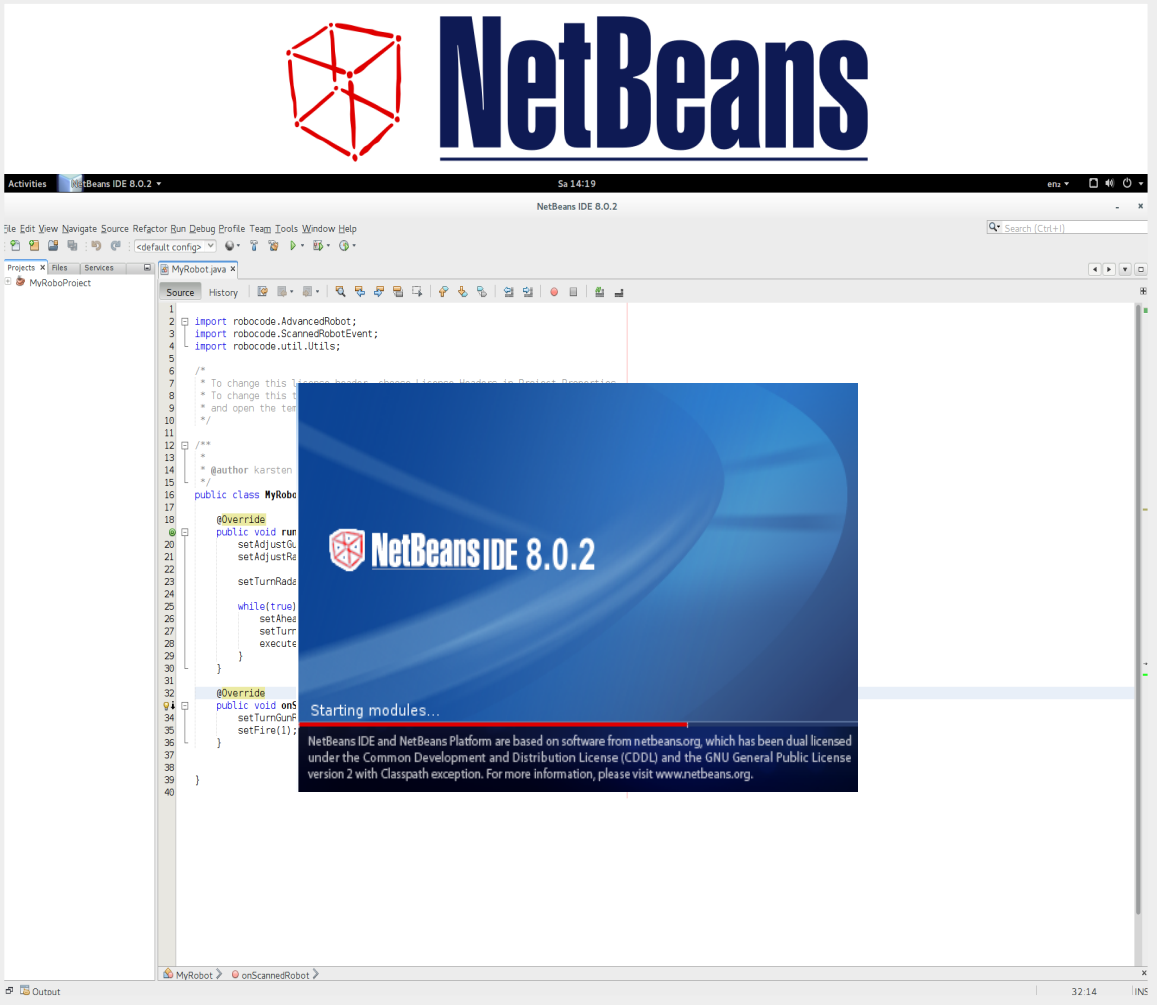




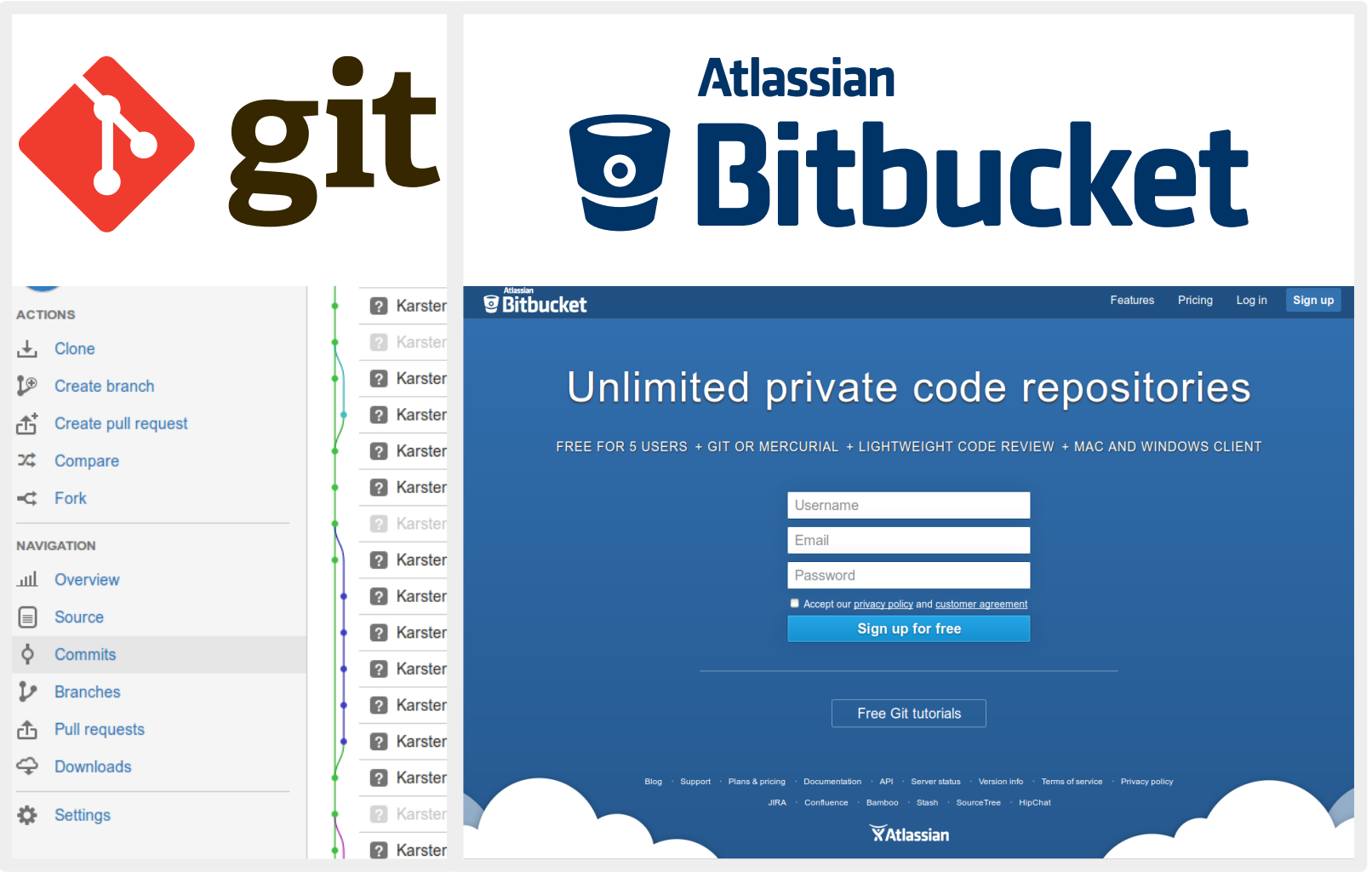
Teaching

The full stack

Tooling – Development

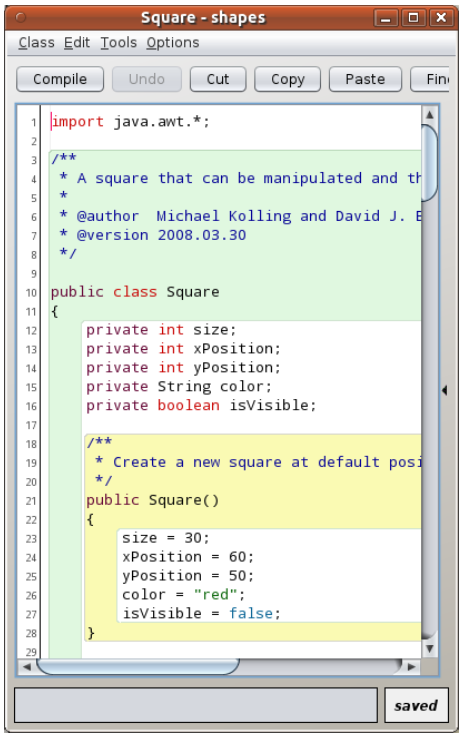


Tooling – Version Control



Tooling – Alternatives

BlueJ



Greenfoot

University of Kent

Different People

New Developer

They need

- Fast progress
- Key concepts (OO, Inheritance)
- Enterprise environment

→ NetBeans

- Fast development cycle
  (Error checking, Maven, …)

Bundled sample projects

Enthusiasts

They have a goal

- Small hacks
- Programming minecraft

→ NetBeans

- Assistance
  (Code completion, Javadoc, …)

Robocode

Code Katas

Example: (Code) Kata

- Originally: Karate moves
  → Begin Simple
  → Get more Complex
  → Repeated until perfection

- Adapted for Programming
  → Code Kata

Example: (Code) Kata

Katas tailored to purpose

- Learn programming
- Learn language
- Learn features, i.e. Java8

Example: (Code) Kata

FizzBuzz

Objective: Learn Programming

- Write a program that prints the integers from 1 to 100.
- For multiples of 3 print "Fizz" instead
- For multiples of 5 print "Buzz".
- For multiples of both print "FizzBuzz".

Demo: (Code) Kata

FizzBuzz

FizzBuzzConcat.java

```
1  package codekata;
2
3  public class FizzBuzzConcat {
4
5      public static void main(String[] args) {
6          for (int i = 1; i <= 100; i++) {
7              String output = "";
8              if (i % 3 == 0) {
9                  output += "Fizz";
10             }
11             if (i % 5 == 0) {
12                 output += "Buzz";
13             }
14             if (output.equals("")) {
15                 output += i;
16             }
17             System.out.println(output);
18         }
19     }
20 }
21
```

FizzBuzzStreams.java

```
1  package codekata;
2
3  import static java.util.stream.IntStream.rangeClosed;
4
5  public class FizzBuzzStreams {
6
7      public static void main(String... arguments) {
8          rangeClosed(1, 100)
9                  .mapToObj(i -> {
10                     if (i % (3 * 5) == 0) {
11                         return "FizzBuzz";
12                     } else if (i % 3 == 0) {
13                         return "Fizz";
14                     } else if (i % 5 == 0) {
15                         return "Buzz";
16                     } else {
17                         return Integer.toString(i);
18                     }
19                 })
20                 .forEach(System.out::println);
```

FizzBuzzSimple.java

```
1  package codekata;
2
3  public class FizzBuzzSimple {
4
5      public static void main(String[] args) {
6          for (int i = 1; i <= 100; i++) {
7              if (i % 15 == 0) {
8                  System.out.println("FizzBuzz");
9              } else if (i % 3 == 0) {
10                 System.out.println("Fizz");
11             } else if (i % 5 == 0) {
12                 System.out.println("Buzz");
13             } else {
14                 System.out.println(i);
15             }
16         }
17     }
18
19 }
20
```

Example: (Code) Kata

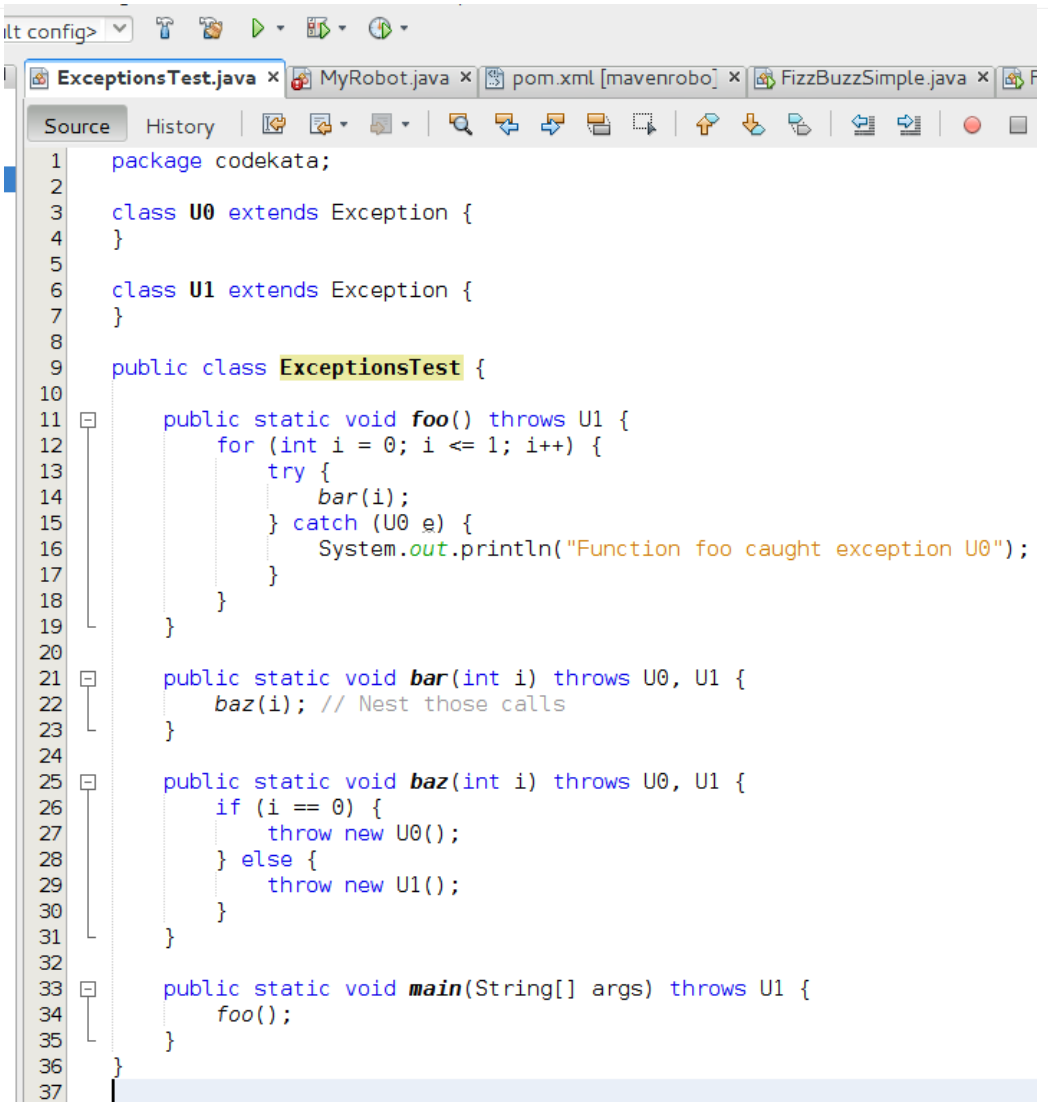
ExceptionsTest

Objective: Learn Language

- Create exceptions, throw and catch them from several nested calls
  - Create two exception classes, U0 and U1
  - Have function foo call function bar twice
  - Have function bar call function baz
  - Function baz should throw exception U0 on its first call, exception U1 on its second.
  - Function foo should catch only exception U0, not U1.
- Show/describe what happens when the program is run.

Demo: (Code) Kata

ExceptionsTest




```java
package codekata;

class U0 extends Exception {
}

class U1 extends Exception {
}

public class ExceptionsTest {

    public static void foo() throws U1 {
        for (int i = 0; i <= 1; i++) {
            try {
                bar(i);
            } catch (U0 e) {
                System.out.println("Function foo caught exception U0");
            }
        }
    }

    public static void bar(int i) throws U0, U1 {
        baz(i); // Nest those calls
    }

    public static void baz(int i) throws U0, U1 {
        if (i == 0) {
            throw new U0();
        } else {
            throw new U1();
        }
    }

    public static void main(String[] args) throws U1 {
        foo();
    }
}
```

Learning by playing

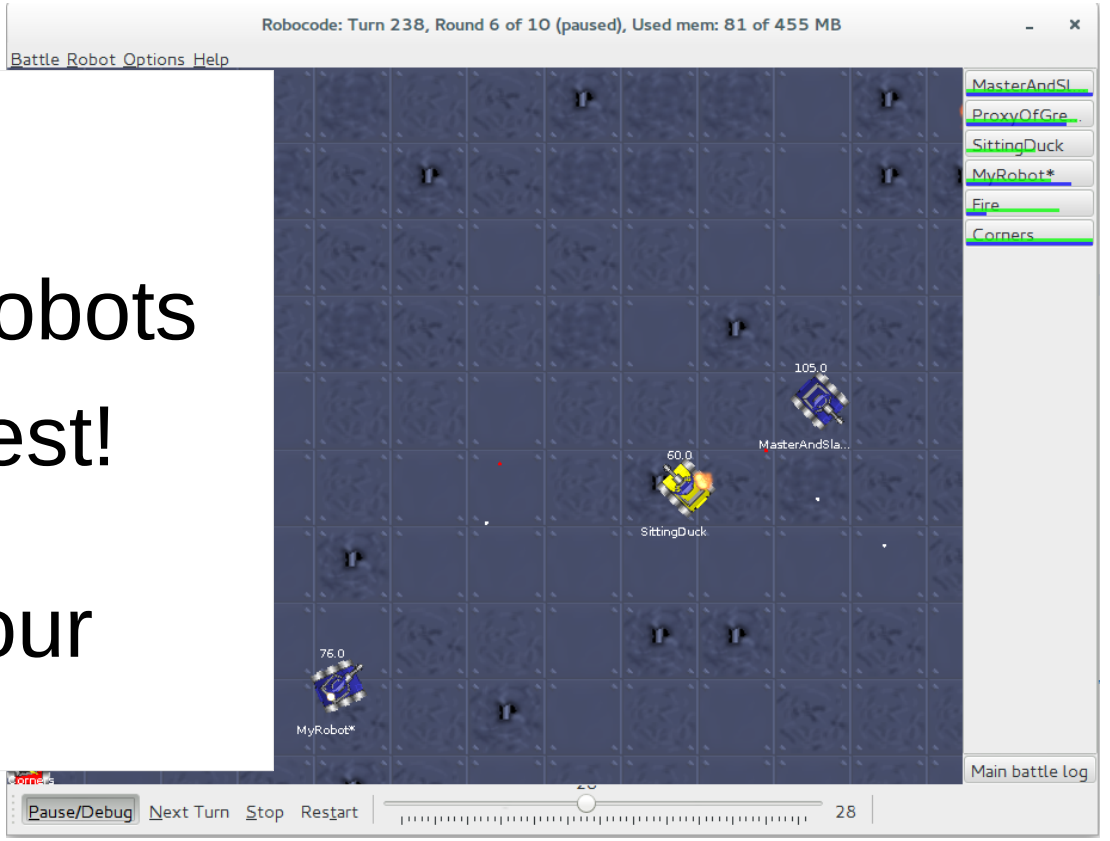
- Greenfoot
- Robocode
- Minecraft

Example: Robocode

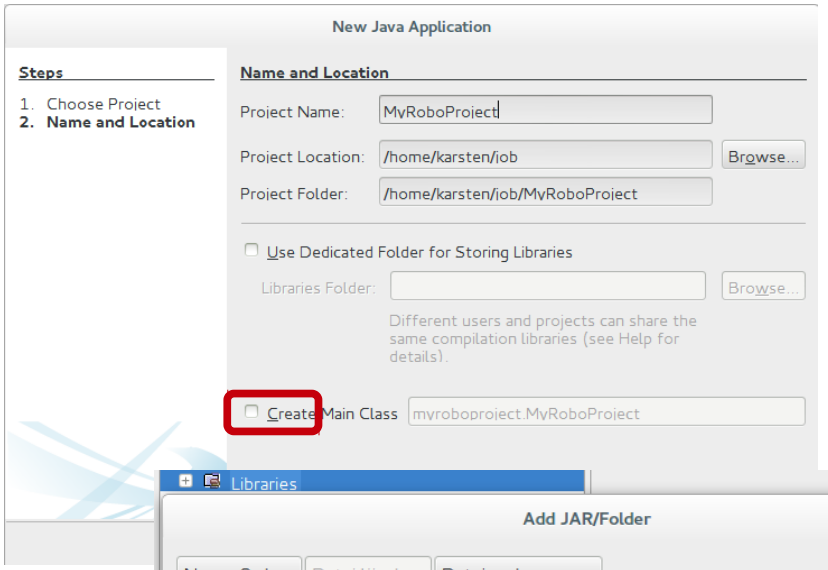
Example: Robocode

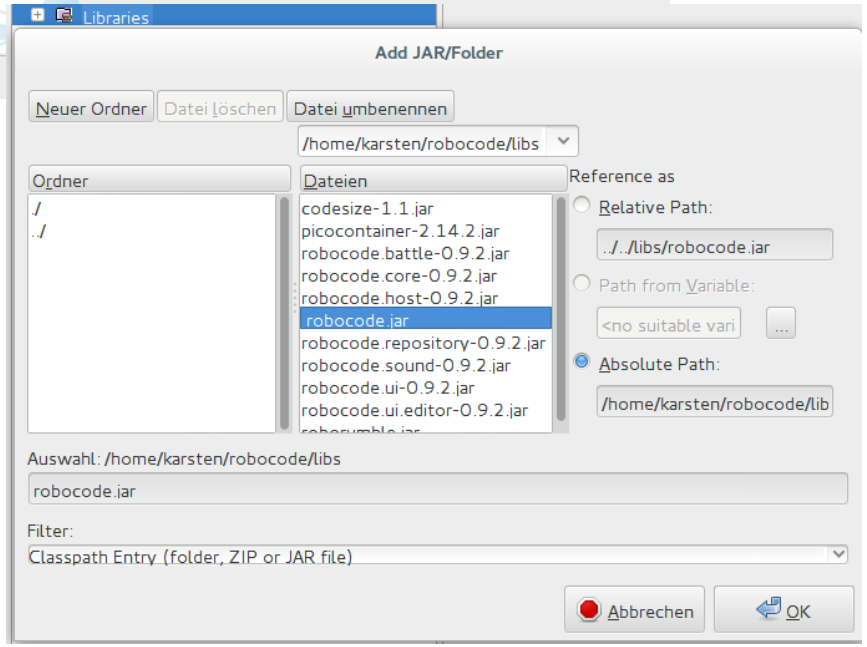
- Robot competition
- Program your own robots

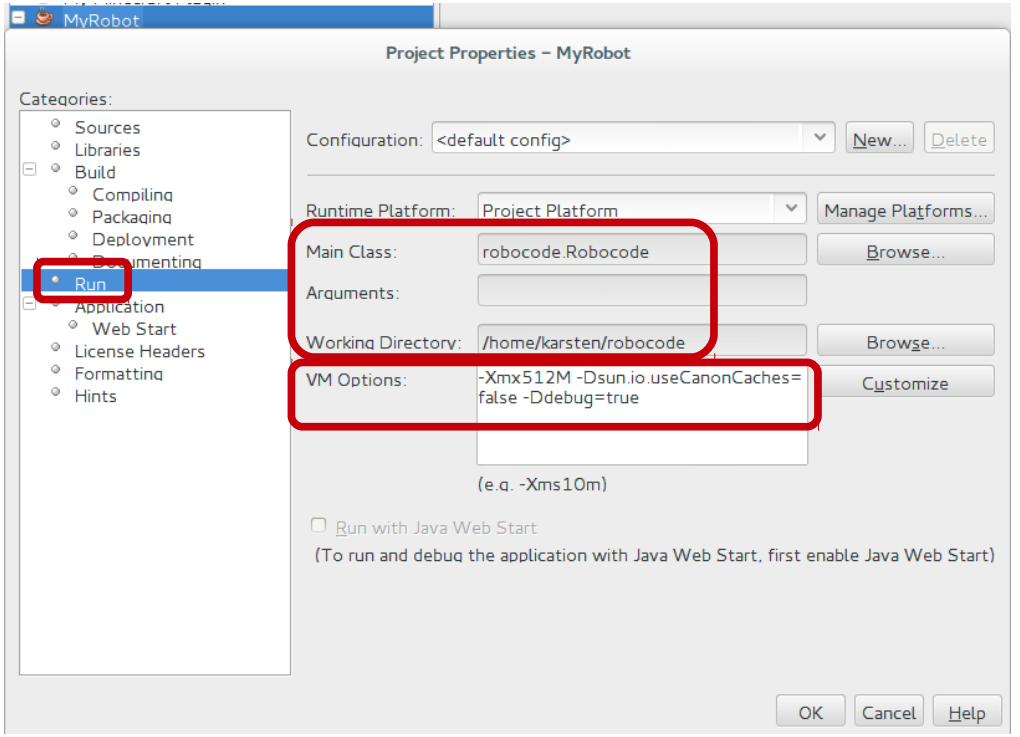
  → Survival of the fittest!
- Share robots with your friends



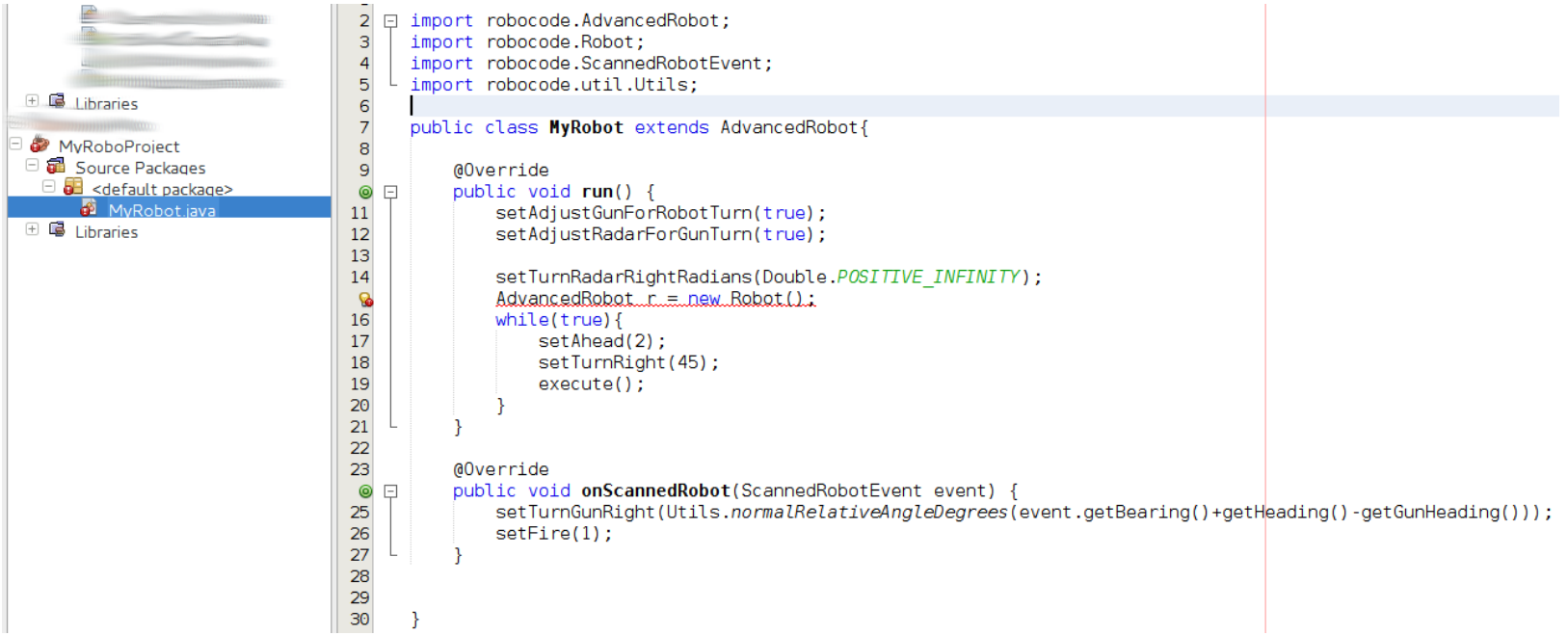
Demo: Robocode

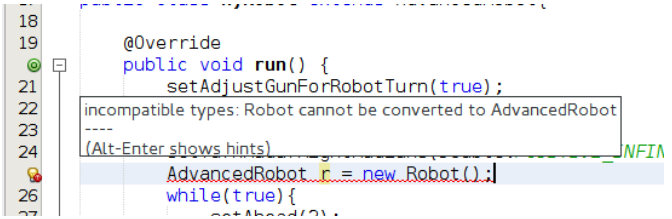




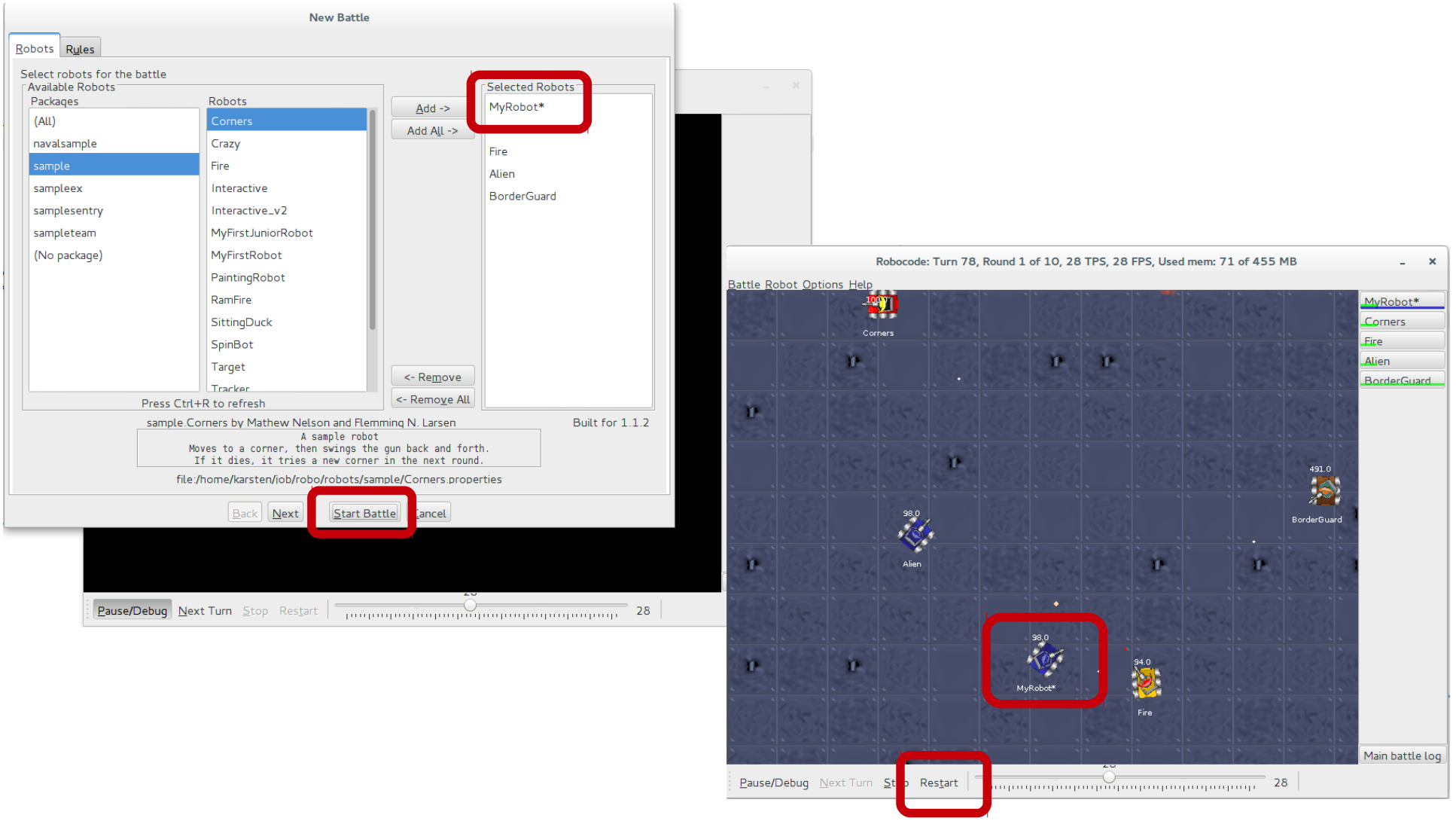


Demo: Robocode





Demo: Robocode

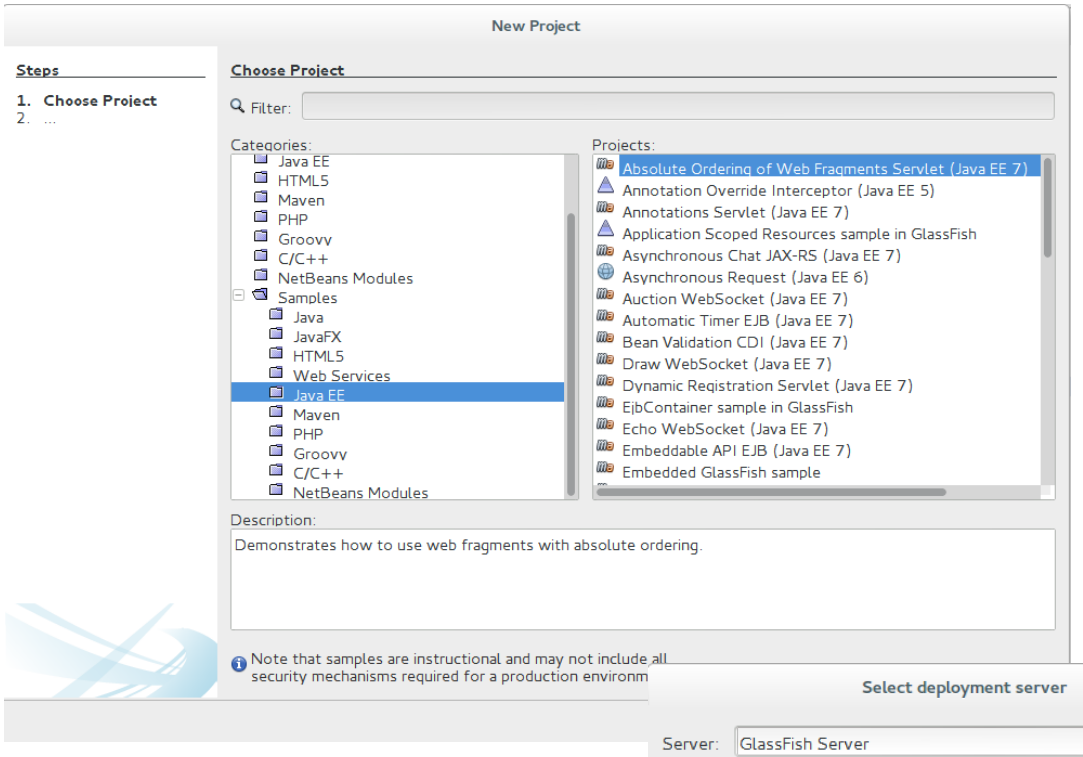


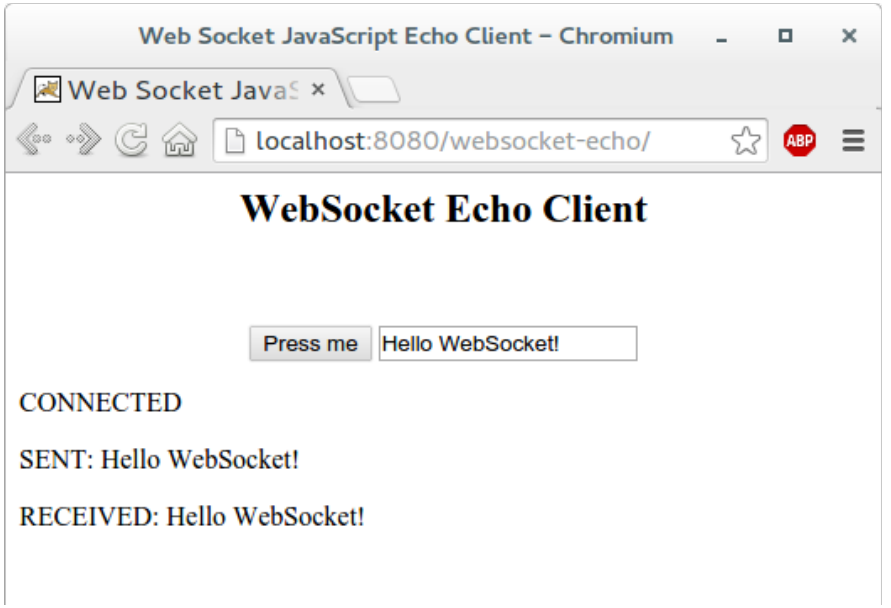
Other training material

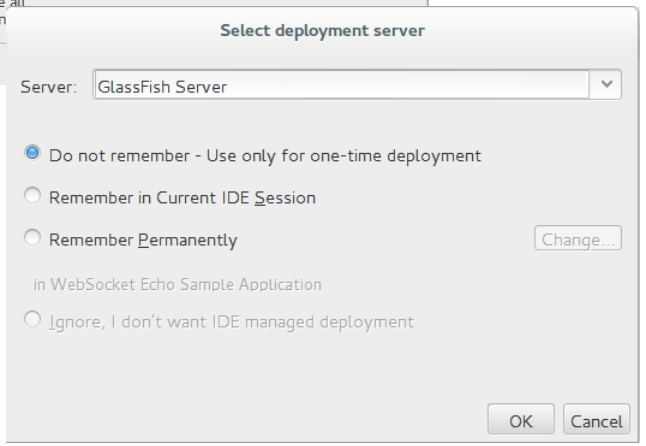
- Sample projects on Bitbucket/GitHub
- Oracle learning trails
- … and Netbeans brings own bundled projects

  + Appserver for JavaEE

Demo: Netbeans samples

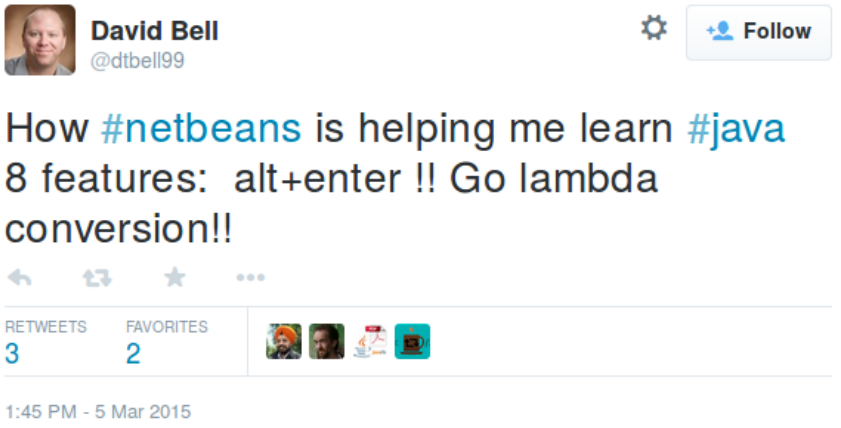


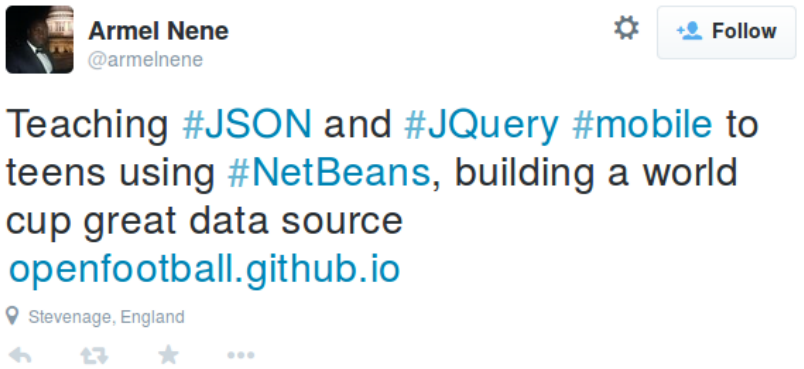




Testimonials/Feedback

'not just java'
→ JavaScript/PHP, HTML/CSS, Databases, C/C++, ...





Python would be good for teaching

Recommendations

Books:

- Head First series (i.e. Java, Programming)

Online resources:

- Netbeans.org (+trials)
- Rosettacode.com
- Greenfoot.org
- Robocode.sourceforge.net

Follow @netbeans on twitter!

Join local community/JUG

Thank you!

Questions?