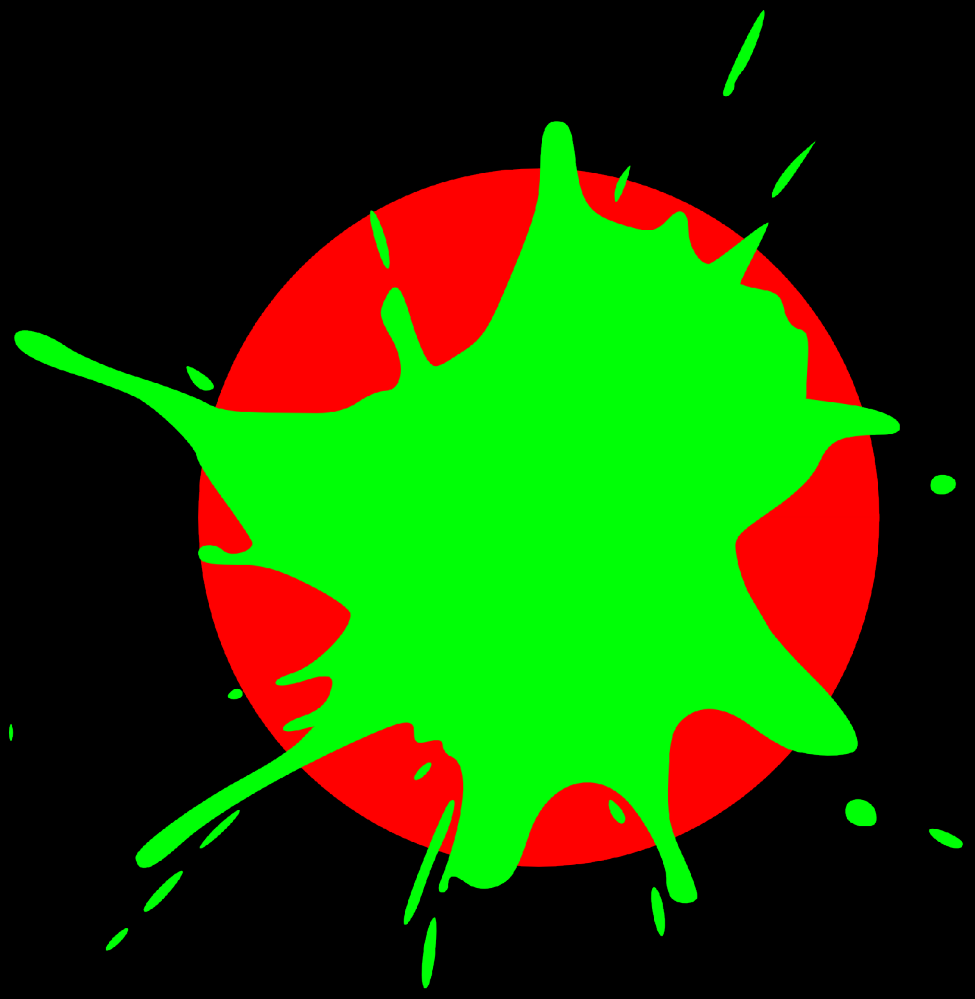


ApiFest 2009

Agenda

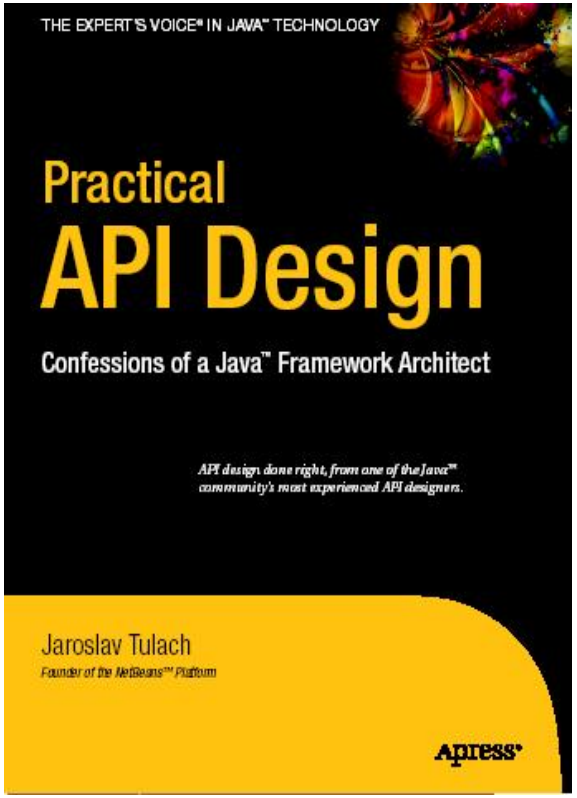
1. What is ApiFest
2. Our competition
3. Examples of incompatibilities
4. And the winner is...

ApiFest

- Creating backward compatible Api
- Search of incompatibilities

ApiFest

- Founder of the competition: Jaroslav Tulach
- Book: Practical API Design



Our competition

Our competition

- Assignment: searching in timetables of bus companies
- Rounds:
  1. Searching in one timetable
  2. Searching in timetables of third parties
  3. Timetables can change on-line
  4. Recurrence: Travels in timetables can be recurrent, whole timetables can be recurrent, recurrence can change on-line.

Solutions

- heterogeneous
- simple as well as very sophisticated (2-25 classes)
- very good

Small quiz

```
1 package apifest;
2
3 public interface MyInterface {
4     public void methodA();
5
6 }
```

```
1 package apifest;
2
3 public interface MyInterface {
4     public void methodA();
5     pubilc void methodB();
6 }
```

Small quiz

```
1 package apifest;
2
3 public interface MyInterface {
4     public void methodA();
5
6 }
```

```
1 package apifest;
2
3 public interface MyInterface {
4     public void methodA();
5     pubilc void methodB();
6 }
```

```
package apifest;

public class Break implements MyInterface {

    public void methodA() {

    }

}
```

Small quiz

```
 1 package apifest;
 2
 3 public class MyClass {
 4
 5     public void methodA() {
 6     }
 7
 8
 9
10 }
```

```
 1 package apifest;
 2
 3 public class MyClass {
 4
 5     public void methodA() {
 6     }
 7
 8     public void methodB() {
 9     }
10 }
```

Small quiz

```
 1 package apifest;
 2
 3 public class MyClass {
 4
 5     public void methodA() {
 6     }
 7
 8
 9
10 }
```

```
 1 package apifest;
 2
 3 public class MyClass {
 4
 5     public void methodA() {
 6     }
 7
 8     public void methodB() {
 9     }
10 }
```

```
package apifest;

public class BreakClass extends MyClass {
    public String methodB() {
        return null;
    }

}
```

Small quiz

```
public final class TransportFactory {

    private TransportFactory() {}

    public static TransportFinder createTransportFinder(Timetable timetable) {
        return new TransportFinderImpl(timetable);
    }

    public static TransportFinder createTransportFinder(Timetable[] timetables) {
        return new TransportFinderImpl(timetables);
    }

}
```

Small quiz

```
public final class TransportFactory {

    private TransportFactory() {}

    public static TransportFinder createTransportFinder(Timetable timetable) {
        return new TransportFinderImpl(timetable);
    }

    public static TransportFinder createTransportFinder(Timetable[] timetables) {
        return new TransportFinderImpl(timetables);
    }

}
```

```
public class BusFinderTest extends TestCase {
    public BusFinderTest(String n) {
        super(n);
    }
....
    public void testCompatibility() throws Exception {
        //here write code which will pass in one version
        TransportFactory.createTransportFinder(null);
    }
}
```

Small quiz

```
public class TransportFinder {
    private Timetable table;

    public TransportFinder(Timetable table) {
        this.table = table;
    }
```

```
17 public class TransportFinder {
18     private Timetable[] tables;
19
20     public TransportFinder(Timetable ... tables) {
21         this.tables = tables;
22     }
23
```

Small quiz

```
public class TransportFinder {
    private Timetable table;

    public TransportFinder(Timetable table) {
        this.table = table;
    }
```

```
17 public class TransportFinder {
18     private Timetable[] tables;
19
20     public TransportFinder(Timetable ... tables) {
21         this.tables = tables;
22     }
23
```

```
public class BusFinderTest extends TestCase {
    public BusFinderTest(String n) {
        super(n);
    }
....
    public void testCompatibility() throws Exception {
        //here write code which will pass in one version
        new TransportFinder(null);
    }
}
```

Small quiz

```
public class Example {

    public Date doSomething(Date date) {
        return date;
    }
```

```
5 public class Example {
6
7     public Date doSomething(Date date) {
8         return new Date(date.getTime());
9     }
```

Small quiz

```
public class Example {

    public Date doSomething(Date date) {
        return date;
    }
```

```
5 public class Example {
6
7     public Date doSomething(Date date) {
8         return new Date(date.getTime());
9     }
```

```
public void testInstance() {
    Date d1 = new Date();
    Date d2 = doSomething(d1);
    assertTrue(d1 == d2);
}
```

hp invent

Small quiz

```
public class Example {

    public void doSomething(Date date) {
        Date internalDate = (Date) date.clone();

        if (internalDate.getTime() > 12345) {
            //do something
        }

    }


}
```

```
 5 public class Example {
 6
 7     public void doSomething(Date date) {
 8         Date internalDate = (Date) date.clone();
 9
10         if (internalDate.getTime() > 12345) {
11             //do something
12         }
13
14         System.out.println(internalDate);
15     }
16
17
18 }
```

Small quiz

```
public class Example {

    public void doSomething(Date date) {
        Date internalDate = (Date) date.clone();

        if (internalDate.getTime() > 12345) {
            //do something
        }


    }


}
```

```
 5 public class Example {
 6
 7     public void doSomething(Date date) {
 8         Date internalDate = (Date) date.clone();
 9
10         if (internalDate.getTime() > 12345) {
11             //do something
12         }
13
14         System.out.println(internalDate);
15     }
16
17
18 }
```

```
public class MyDate extends Date {

    @Override
    public Object clone() {
        return this;
    }

    @Override
    public String toString() {
        throw new RuntimeException();
    }

}
```

hp invent

Bigger quiz

Test.java

```
1
2 public class Test {
3     public String test() {
4         return "hello world!";
5     }
6     public static void main(String[] args) {
7         System.out.println(new Test().test());
8     }
9 }
```

Test.java

```
1
2 public class Test {
3     public String test() {
4         String s1="hello ";
5         String s2="world!";
6         return s1+s2;
7     }
8     public static void main(String[] args) {
9         System.out.println(new Test().test());
10    }
11 }
```

Bigger quiz

Test.java

```
1
2 public class Test {
3     public String test() {
4         return "hello world!";
5     }
6     public static void main(String[] args) {
7         System.out.println(new Test().test());
8     }
9 }
```

Test.java

```
1
2 public class Test {
3     public String test() {
4         String s1="hello ";
5         String s2="world!";
6         return s1+s2;
7     }
8     public static void main(String[] args) {
9         System.out.println(new Test().test());
10    }
11 }
```

```
public class TestBreak {

    public static void main(String[] args) {
        Test t = new Test();
        System.out.println(t.test()==t.test());
    }
}
```

hp invent

Bigger quiz

Test.java

```
 1
 2 public class Test {
 3     private String privs;
 4     public String test(String s) {
 5         privs = s;
 6     return s+"!";
 7     }
 8     public static void main(String[] args) {
 9         System.out.println(new Test().test("hello world"))
10     }
11 }
```

Test.java

```
 1
 2 public class Test {
 3     private String privs;
 4     public String test(String s) {
 5         privs = s+"!";
 6     return privs;
 7     }
 8     public static void main(String[] args) {
 9         System.out.println(new Test().test("hello world"))
10     }
11 }
```

hp invent

Bigger quiz

Test.java

```
 1
 2 public class Test {
 3     private String privs;
 4     public String test(String s) {
 5         privs = s;
 6     return s+"!";
 7     }
 8     public static void main(String[] args) {
 9         System.out.println(new Test().test("hello world"))
10     }
11 }
```

Test.java

```
 1
 2 public class Test {
 3     private String privs;
 4     public String test(String s) {
 5         privs = s+"!";
 6     return privs;
 7     }
 8     public static void main(String[] args) {
 9         System.out.println(new Test().test("hello world"))
10     }
11 }
```

```
import java.util.*;

public class TestBreak {

    public static void main(String[] args) {
        Test t = new Test();
        String s1 = "hello ";
        String s2 = "world";
        String s = s1+s2;
        t.test(s);
        Map m = new WeakHashMap();
        m.put(s, new Object());
        s = null;
        for (int i = 0; i<10; i++) {
            try { Thread.sleep(500); } catch (Exception e) {}
            System.gc();
        }
        System.out.println(m.size()==1);
        t.test("asdf");
    }
}
```

hp invent

Bigger quiz

Test.java

```
1
2 public class Test {
3     public int test() {
4         return "hello world!".length();
5     }
6     public static void main(String[] args) {
7         System.out.println(new Test().test());
8     }
9 }
```

Test.java

```
1
2 public class Test {
3     public int test() {
4         String s1="hello ";
5     String s2="world!";
6     return (s1+s2).length();
7     }
8     public static void main(String[] args) {
9         System.out.println(new Test().test());
10    }
11 }
```

hp invent

Bigger quiz

Test.java

```
1
2 public class Test {
3     public int test() {
4         return "hello world!".length();
5     }
6     public static void main(String[] args) {
7         System.out.println(new Test().test());
8     }
9 }
```

Test.java

```
1
2 public class Test {
3     public int test() {
4         String s1="hello ";
5     String s2="world!";
6     return (s1+s2).length();
7     }
8     public static void main(String[] args) {
9         System.out.println(new Test().test());
10    }
```

```
import java.util.*;

public class TestBreak {

    public static void main(String[] args) {
        Test t = new Test();
        String s1 = "hello ";
        String s2 = "world!";
        String s = (s1+s2).intern();
        t.test();
        Map m = new WeakHashMap();
        m.put(s, new Object());
        s = null;
        for (int i = 0; i<10; i++) {
            try { Thread.sleep(500); } catch (Exception e) {}
            System.gc();
        }
        System.out.println(m.size()==1);
    }
}
```

hp invent

Solution number 9

- with common techniques practically unbreakable
- the competition has changed in the hunt to solution 9

Solution number 9 – full memory

```
 1 package apifest;
 2
 3 public class Example {
 4
 5
 6
 7     public void doSomething() {
 8
 9         System.out.println("Hello world!");
10     }
11
12
13 }
```

```
 1 package apifest;
 2
 3 public class Example {
 4
 5     Object x;
 6
 7     public void doSomething() {
 8         x = new Object();
 9         System.out.println("Hello world!");
10     }
11
12
13 }
```

Solution number 9 - classloader

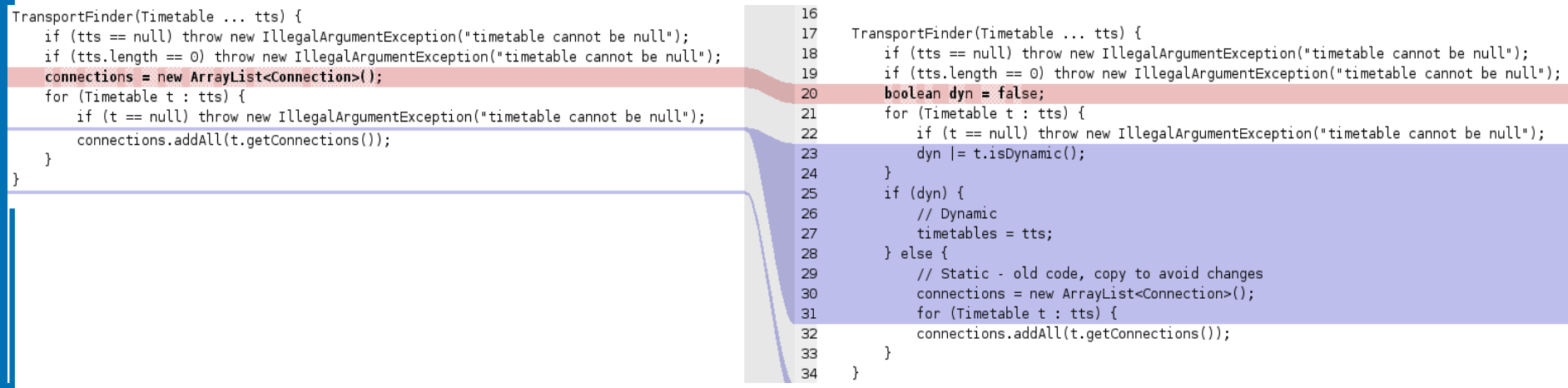
```
class MyClassLoader extends ClassLoader {

    public Class loadClass(String name,boolean resolve) throws ClassNotFoundException {
        if (countCheckSum(name)==countCheckSum("apifest.BusFinderTest$ALL")) {
            // there is a bug in this classloader - it can't load classes with
            // specific check sums
            throw new ClassNotFoundException("this classloader can't load this class");
        }

    ...
```

Solution number 9

- break based on different asymptotical complexity
- break based on two threads

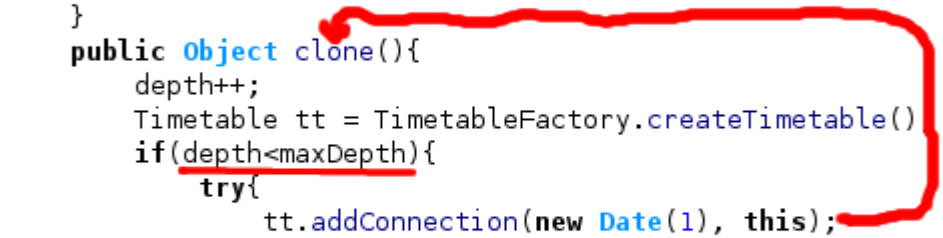


```
TransportFinder(Timetable ... tts) {
    if (tts == null) throw new IllegalArgumentException("timetable cannot be null");
    if (tts.length == 0) throw new IllegalArgumentException("timetable cannot be null");
    connections = new ArrayList<Connection>();
    for (Timetable t : tts) {
        if (t == null) throw new IllegalArgumentException("timetable cannot be null");
        connections.addAll(t.getConnections());
    }
}
```

```
16
17    TransportFinder(Timetable ... tts) {
18        if (tts == null) throw new IllegalArgumentException("timetable cannot be null");
19        if (tts.length == 0) throw new IllegalArgumentException("timetable cannot be null");
20        boolean dyn = false;
21        for (Timetable t : tts) {
22            if (t == null) throw new IllegalArgumentException("timetable cannot be null");
23            dyn |= t.isDynamic();
24        }
25        if (dyn) {
26            // Dynamic
27            timetables = tts;
28        } else {
29            // Static - old code, copy to avoid changes
30            connections = new ArrayList<Connection>();
31            for (Timetable t : tts) {
32            connections.addAll(t.getConnections());
33        }
34    }
```

Solution number 9 - loop

```
public class MyDateBreaker extends Date{
    public long depth=0;
    public long maxDepth=0;
    public MyDateBreaker(long i,long maxDepth){
        super(i);
        this.maxDepth= maxDepth;
    }
    public Object clone(){
        depth++;
        Timetable tt = TimetableFactory.createTimetable()
        if(depth<maxDepth){
            try{
                tt.addConnection(new Date(1), this);
            }catch (StackOverflowError sof){
                throw new RuntimeException("break!");
            }
        }
        return super.clone();
    }
}
```



And the winner is…

Winners

- Received the same number of points – 22

  Aleš Jeřábek, Jakub Trávník,

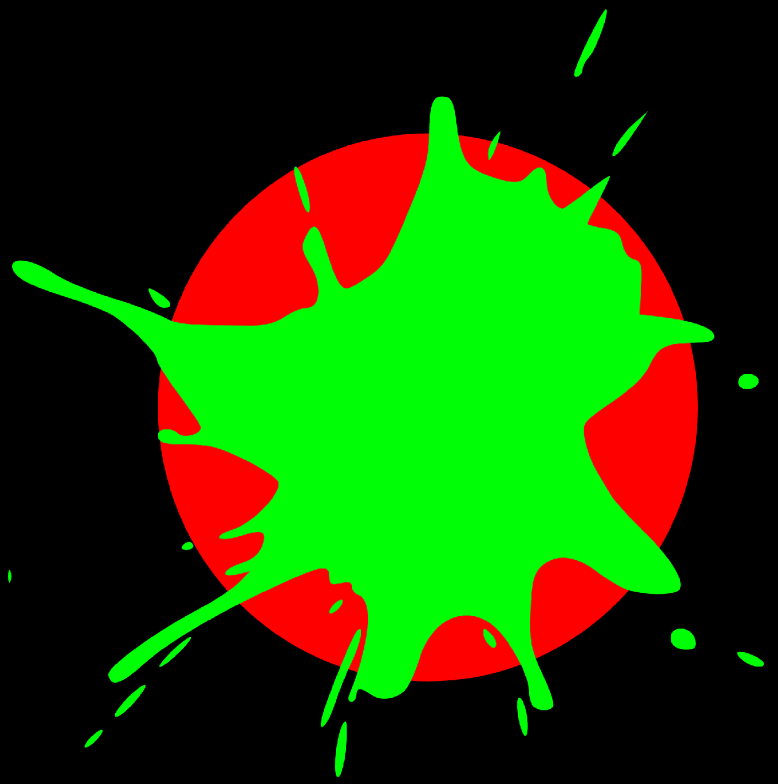
  Martin Frýdl, Tomáš Sieger




Winner of Special Price of Jury

- Winner of Special Price of Jury for the best API: Martin Frýdl

Thank you for your participation



ApiFest 2009