

# Statická analýza kódu - za kód bez chyb

**Václav Pech**

Senior Software Developer

JetBrains, Inc.



## Něco o mě

- Václav Pech
  - Zhusta chybující programátor 8 let
  - Vděčný uživatel nástrojů pro analýzu kódu 3 roky
- JetBrains
  - Dodavatel nástrojů pro vývojáře
    - IntelliJ IDEA, TeamCity, ReSharper, a další



JavaMagazin



## Něco o prezentaci

- 1. Chyby v kódu
  - 2. On-demand analýza
  - 3. On-the-fly analýza
  - 4. Pro starší a pokročilé
- 
- Otázky a odpovědi
  - **POZOR!: Během prezentace bude ukazován a editován kód**



# Příklad 1



```
public void parse(final String text) {  
    if ((text == null) | (text.length()==0)) return;  
}
```

Neznámý programátor

## Příklad 2



```
public class InvalidSetter {
    private String employeeName;

    public void setEmployeeName(final String anEmployeeName) {
        if ((anEmployeeName != null) && (anEmployeeName.length() > 0))
            this.employeeName = anEmployeeName;
    }
}
```

Václav Pech, 2004

## Příklad 3



```
public boolean equals( Object object )
{
    if ( this == object )
    {
        return true;
    }
    return getId().equals( ( PluginDescriptor ) object ).getId() );
}
```

Maven 2.0.x SNAPSHOT

## Příklad 4



```
private void paintRed(final Rectangle clip, final Graphics g) {  
    g.setColor(Color.red);  
    if (clip!=null) return;  
    g.fillRect(clip.x, clip.y, clip.width, clip.height);  
}
```

- Nejmenovaný programátor u JetBrains

## Příklad 5



```
public void displayCustomer(Customer customer) {  
    if (customer==null)  
        logger.trace("Customer not set yet");  
    customer=createNewCustomer();  
    showCustomerDetails(customer);  
}
```

Václav Pech, 2005

## Situace

- Všichni vývojáři občas dělají chyby
- Chyby jsou i ve známých a velmi rozšířených knihovnách a nástrojích
- Většina chyb jsou triviality



## Poznej svého nepřítele

- Bad practice
- Correctness
- Internationalization
- Malicious code vulnerability
- Multithreaded correctness
- Performance
- Code style violations
- Dodgy



(Bill Pugh, FindBugs)

# Bad practice

```
try
{
    child.getContainerRealm().importFrom( "plexus.core", "org.codehaus.plexus.util.xml.Xpp3Dom" );
    child.getContainerRealm().importFrom( "plexus.core", "org.codehaus.plexus.util.xml.pull" );
}
catch ( NoSuchRealmException e )
{
    // won't happen
}
```

# Correctness



```
final BigDecimal summary=BigDecimal.ZERO;  
  
public void addToSummary(final BigDecimal value) {  
    summary.add(value);  
}
```

# Internationalization

```
final JLabel title = new JLabel("First name:");
```

```
final JLabel price = new JLabel(totalPrice.toString());
```

```
if (countryName1.compareTo(countryName2) < 0) {
```

## Malicious code vulnerability

```
public static final String[] ALL_EVENTS = {  
    PHASE_EXECUTION,  
    MOJO_EXECUTION,  
    PROJECT_EXECUTION,  
    REACTOR_EXECUTION  
};
```

# Multithreaded correctness



```
public class SharedCount {  
    private static long count=0;  
  
    public synchronized void increment() {  
        count++;  
    }  
}
```

# Performance

```
componentPrintString.append(" " + name);
```

```
private HashMap components = new HashMap();
```

```
if (components.size() > 0) {
```

# Code style violations

```
public void setActiveByDefault(boolean activeByDefault)
```

```
if (((property!=null))) {
```

```
if (property!=null) return;
```

```
int a, b, c;
```

## Dodgy

```
public final class Dodgy {  
    protected int count;  
    private String value="Never used";  
}
```

## Nástroje pro detekci chyb v kódu

- Debugger
- Profiler
- Unit tests
- Code revisions
  - Formal revisions
  - Pair programming
- Static code analysis

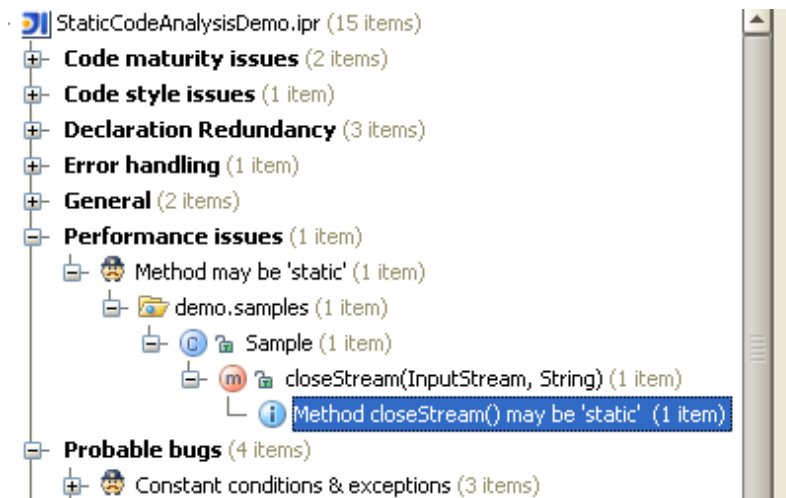


## Statická analýza kódu

- Detekuje chyby v kódu staticky na základě hledání chybových vzorů
  - Abstract Syntax Tree
- Analyzuje zdrojový kód, nebo bytecode
- Integruje s IDEčky, Antem, Mavenem či CI servery

# On-demand analýza

- Generuje reporty o nalezených chybách

**Name:**

public method void **closeStream**(InputStream in, String value)

**Location:**

class [Sample](#) (demo.samples)

**Problem synopsis:**

Method [closeStream\(\)](#) may be 'static' at line [29](#)

**Problem resolution:**

[Make static](#)

**Suppress:**

[Suppress for method](#)

## On-the-fly analýza

- Upozorňuje na chyby přímo v editoru

```
if (in == null)
```

```
    try {
```

```
        in.close();
```

```
    }
```

```
Method invocation 'in.close()' may produce 'java.lang.NullPointerException'.
```

## Pokročilé vlastnosti



- Konfigurace hledaných chyb a jejich závažnosti pomocí profilů
- Možnost potlačení (suppress) hlášení
- Exportování reportů pro jejich off-line prohlížení a mining

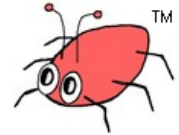
## Příklady dostupných nástrojů

- IntelliJ IDEA
- TPTP plugin for Eclipse
- FindBugs
- Checkstyle
- PMD
- AppPerfect Code Analyzer
- Fortify – focused on security

## IntelliJ IDEA

- Analýza integrovaná do IDE
  - Přes 700 hledaných chybových vzorů
  - Custom bug patterns
  - Profile management
  - Suppression pomocí anotací
- 
- Analýza závislostí, DSM
  - Detekce duplicit v kódu

## FindBugs



- Open source – Bill Pugh, Univ. Of Maryland
- Integruje do Eclipse a NetBeans
- Custom bug patterny v Javě
- Propracovaná správa reportů a historie chyb
- Analyzuje byte-code
- Přes 300 hledaných chybových vzorů
  
- Nemá quick-fixy
- Pouze pro Java kód

## TPTP



- Open source
- Integrovan do Eclipse
- Více než 100 patternů (včetně JDT)
- Custom bug patterny
  
- Pouze pro Java kód

## CheckStyle



- Open source
- Integruje se všemi zmiňovanými IDE
- Custom bug patterny v Javě
- Okolo 100 patternů
- Zaměřen na problémy stylu
  - Java and EJB 2.x
- Detekce duplicit v kódu
  
- Nemá quick fixy
- Pouze pro Java kód



## PMD

- Open source
- Integruje se všemi zmiňovanými IDE
- Custom bug patterny v Javě a XPath
- Okolo 200 patternů v Javě, JSP, JSF
- Suppression pomocí anotací
- Detekce duplicit v kódu
  
- Nemá quick fixy

## Příklad 6



```
public double requestCurrentRate(String fromCurrency, String toCurrency) {  
    if ((fromCurrency == null) && (toCurrency == null)) {  
        return Double.NaN;  
    }  
    double answer= 0.0;  
    if (fromCurrency.equals("USD") && toCurrency.equals("EUR")) {  
        answer= usdRate;  
    } else if (fromCurrency.equals("EUR") && toCurrency.equals("USD")) {  
        answer= 1.0 /usdRate;  
    }  
    return answer;  
}
```

## Další formáty

- Java in JSPs
- JavaScript
- CSS, HTML, XML
- GWT
- Struts
- Java EE code
- Java ME code

```
1. prodname span {  
    display: block;  
    margin: -3px 0 0 0;  
    padding: 0;  
    font-size: 0.8em;  
    font-weight: normal;  
    font-size: 70%;  
}
```

Property font-size is overwritten

```
var selectedQuantityIndex = volumeQuantities.length - 1;  
if ((quantity == null) || (isNaN(quantity))) {  
    Comparison quantity == null may cause unexpected type coercion
```

## Custom bug patterns

- Možnost definovat vlastní vzory pro detekci porušení specifických pravidel daného projektu či domény
- Příklady:
  - Volání určité metody
  - Vytvoření určité třídy
  - další
- Někdy je vhodné definovat rovněž quick fixy

## Inspection annotations

- @Nullable, @NotNull
- @Nls, @NonNls
- @PropertyKey
- @Pattern, @Language
  
- @Nonnegative, @Signed
- @Tainted, @Untainted
- @ThreadSafe, @GuardedBy
- JSR 305 and 308
  - Expert group zahrnuje rovněž vývojáře FindBugs and IntelliJ IDEA

## Inspection annotations – Příklad 1

```
@Nullable String myName;
```

```
@Nullable private String getName() {  
    return myName;  
}
```

```
public void test() {  
    String name = getName();  
    if (name.length() > 0) {  
        }  
}
```

Method invocation 'name.length()' may produce 'java.lang.NullPointerException' [more...](#) (Ctrl+F1)

## Inspection annotations – Příklad 2

```
abstract class Tool {  
    @Pattern("[a-zA-Z_0-9]+")  
    abstract String getId();  
}
```





```
class MyTool extends Tool {  
    String getId() {  
        return "ID (Not Valid)";  
    }  
}
```

Expression 'ID (Not Valid)' doesn't match pattern: [a-zA-Z\_0-9]+ [more...](#) (Ctrl+F1)

## Inspection annotations – Příklad 3

```
String getMessage(  
    @PropertyKey(resourceBundle="demo.bundle")  
    String key) {...}
```

```
void test() {  
    String s1 = getMessage("no.such.key");  
    String s2 = getMessage("key");  
}
```

	key=My String		bundle
	key2 =Other String		bundle

## Inspection annotations – Příklad 4

```
@Language("CSS")  
String cssDemo = "h2 { font-style: italic; }";
```

```
@Language("RegExp")  
String regexpDemo = "\\p{Alpha}(abc)\\1\\2";
```

Unresolved backreference

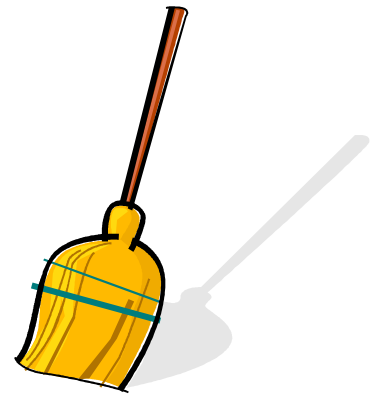
```
@Language("HTML")  
String xmlDemo = "<head><title>Test</title></hed>";
```

## Pust'te se do hledání chyb

- IntelliJ IDEA
- Eclipse
  - TPTP plugin
  - FindBugs plugin
- NetBeans
  - SQE plugin
    - FindBugs, PMD, CheckStyle, Lint4J

## Shrnutí

- Statická analýza kódu zamete s chybami
  - On-demand
  - On-the fly
- Výborný doplněk dalších metod
- Dostupná pro všechna IDE



A large, stylized yellow question mark graphic that serves as a background for the title text.

# Otázky