



DHBW

Stuttgart

Java-TX Compiler in Java-TX

Julian Schmidt

<https://www.dhbw-stuttgart.de>

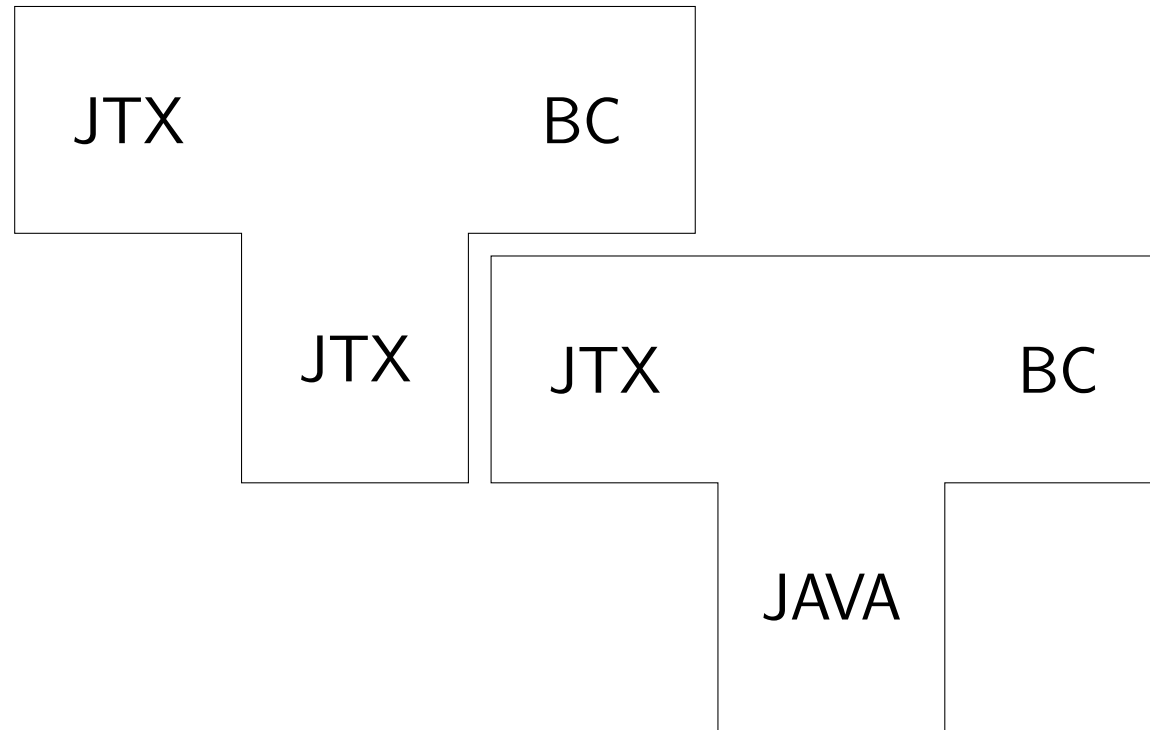
Agenda

- 1 Motivation
- 2 Aufbau der Umgebung
- 3 Probleme
 - 1 Überschreibung von Methoden
 - 2 Lambda Ausdrücke
- 4 Bugs

Motivation

- Welche Features fehlen noch in Java-TX?
- Welche Bugs gibt es?
- Wie performant ist Java-TX für größere Projekte?
- Vorteile/Nachteile zu Java in der Praxis

Motivation



Aufbau der Umgebung

Erster Versuch mit make:

```
1  # Use find to locate all .java and .jav files recursively
2  JAVASOURCES := $(shell find $(SRCDIR) -name '*.java')
3  JAVSOURCES := $(shell find $(SRCDIR) -name '*.jav')
4
5  # Convert .java/.jav files to .class files with the same directory structure
6  JAVACLASSES := $(patsubst $(SRCDIR)/%.java,$(DESTDIR)/%.class,$(JAVASOURCES))
7  JAVCLASSES := $(patsubst $(SRCDIR)/%.jav,$(DESTDIR)/%.class,$(JAVSOURCES))
8
9  # Rule for compiling .jav files
10 $(DESTDIR)/%.class: $(SRCDIR)/%.jav
11 java -jar $(JTX) -d "$(dir $@)" -cp "$(SRCDIR):$(DESTDIR):target/dependencies/" $<
12 # Rule for compiling .java files
13 $(DESTDIR)/%.class: $(SRCDIR)/%.java
14 $(JC) -nowarn -d $(DESTDIR) -cp "$(SRCDIR):$(DESTDIR):target/dependencies/*" $(JFLAGS) $<
```

Probleme:

- 1 javac compiliert und trackt Änderungen der Abhängigkeiten automatisch
- 2 javac ist sehr langsam wenn für jede Datei einzeln aufgerufen

```
1 javac src/main/java/de/dhbwstuttgart/  
  typedeployment/TypeInsert.java  
2 javac src/main/java/de/dhbwstuttgart/  
  typedeployment/TypeInsertPlacer.java  
3 ...  
4 javac src/main/java/Main.java
```

~ 5min

```
1 javac src/main/java/de/dhbwstuttgart/  
  typedeployment/TypeInsert.java src/main/java/  
  de/dhbwstuttgart/typedeployment/  
  TypeInsertPlacer.java ... src/main/java/Main.  
  java
```

~ 2sec

Bash Script

Gegeben: Quellverzeichnis, Zielverzeichnis

- 1 Suche rekursiv alle .java und .jav Dateien im Quellverzeichnis und speichere sie jeweils in einer Liste
- 2 Überprüfe für jede Quelldatei, ob die zugehörige .class Datei im Zielverzeichnis existiert und ob die Zieldatei neuer als die Quelldatei ist
 - Wenn ja, gehe weiter zur nächsten Datei
 - Wenn nein, füge die Quelldatei zur Liste der zu kompilierenden Dateien hinzu
- 3 Rufe den Java-TX Compiler mit allen Dateien in der jav-Liste als Argumente auf

```
java -jar $JAVATX_COMPILER_PATH -d $DESTDIR -cp "$SRCDIR:$DESTDIR:target/dependencies/" "${JAV_CHANGED[@]}"
```
- 4 Rufe den javac Compiler mit allen Dateien in der java-Liste als Argumente auf

```
javac -d $DESTDIR -cp "$SRCDIR:$DESTDIR:target/dependencies/*" $JAVAC_FLAGS "${JAVA_CHANGED[@]}"
```

Bugs

