

Fragen, Antworten und Kommentare zur aktuellen Vorlesung

Bitte nehmen Sie an der anonymen Lehrevaluation unter <https://forms.gle/5Xj4V8FmbYJjAmzB7> bis zum 11.6 teil. Die von mir kommentierten Ergebnisse stehen im letzten Fragen&Antworten-Dokument des Semesters.

Hinweis: Informationen zum Stand der Software-Qualität können diesen regelmäßig durchgeführten Befragungen entnommen werden: <https://www.softwaretest-umfrage.de/>

Frage: Im VL-Video, klingt es so, als ob Sie von Capture&Replay abraten.

Antwort: „Abraten“ stimmt nicht ganz. Wenn man die entstehenden Skripte sehr einfach weiterverarbeiten, also anpassen und neue Skripte daraus erstellen kann, stellt der Ansatz eine sinnvolle Möglichkeit da. Da ich mich noch an Zeiten (Ende 1990er) erinnere, in denen Capture&Replay als DIE ultimative Testlösung verkauft wurde, habe ich da immer einen unangenehmen Geschmack im Mund. Aber, das over-hypen von Techniken mit sinnvollem Hintergrund ist ja ein typisches Verhalten nicht nur in der IT-Branche (z. B. auch Low-/No-Code oder auch KI).

Frage: In der VL wird eine Klasse GUIExtension genutzt, wo finde ich die?

Antwort: Sorry, fehlte in der Lernnotiz. Die Klasse wird für die Praktikumsaufgaben aber nicht benötigt.

```
package assertj5;

// Quelle: https://stackoverflow.com/questions/70631626/assertj-swing-and-junit-5-support
import org.assertj.swing.junit.runner.FailureScreenshotTaker;
import org.assertj.swing.junit.runner.ImageFolderCreator;
import org.junit.jupiter.api.extension.Extension;
import org.junit.jupiter.api.extension.ExtensionContext;
import org.junit.jupiter.api.extension.InvocationInterceptor;
import org.junit.jupiter.api.extension.ReflectiveInvocationContext;

import java.lang.reflect.Method;

import static org.assertj.swing.annotation.GUIExtensionFinder.isGUIExtension;
import static org.assertj.swing.junit.runner.Formatter.testNameFrom;

/**
 * Understands a JUnit 5 extension that takes a screenshot of a failed
 * GUI test.
 * The JUnit 4 runner is available in
 * {@link org.assertj.swing.junit.runner.GUIExtensionRunner}.
 * @author William Bakker
 */
public class GUIExtension implements Extension, InvocationInterceptor {
    private final FailureScreenshotTaker screenshotTaker;
```

```

public GUIExtension() {
    screenshotTaker = new FailureScreenshotTaker(
        new ImageFolderCreator().createImageFolder());
}

@Override
public void interceptTestMethod(
    Invocation<Void> invocation,
    ReflectiveInvocationContext<Method> invocationContext,
    ExtensionContext extensionContext)
    throws Throwable {
    try {
        invocation.proceed();
    } catch (Throwable t) {
        takeScreenshot(invocationContext.getExecutable());
        throw t;
    }
}

private void takeScreenshot(Method method) {
    final Class<?> testClass = method.getDeclaringClass();
    if (!(isGUIExtension(testClass, method)))
        return;
    screenshotTaker.saveScreenshot(testNameFrom(testClass, method));
}
}

```