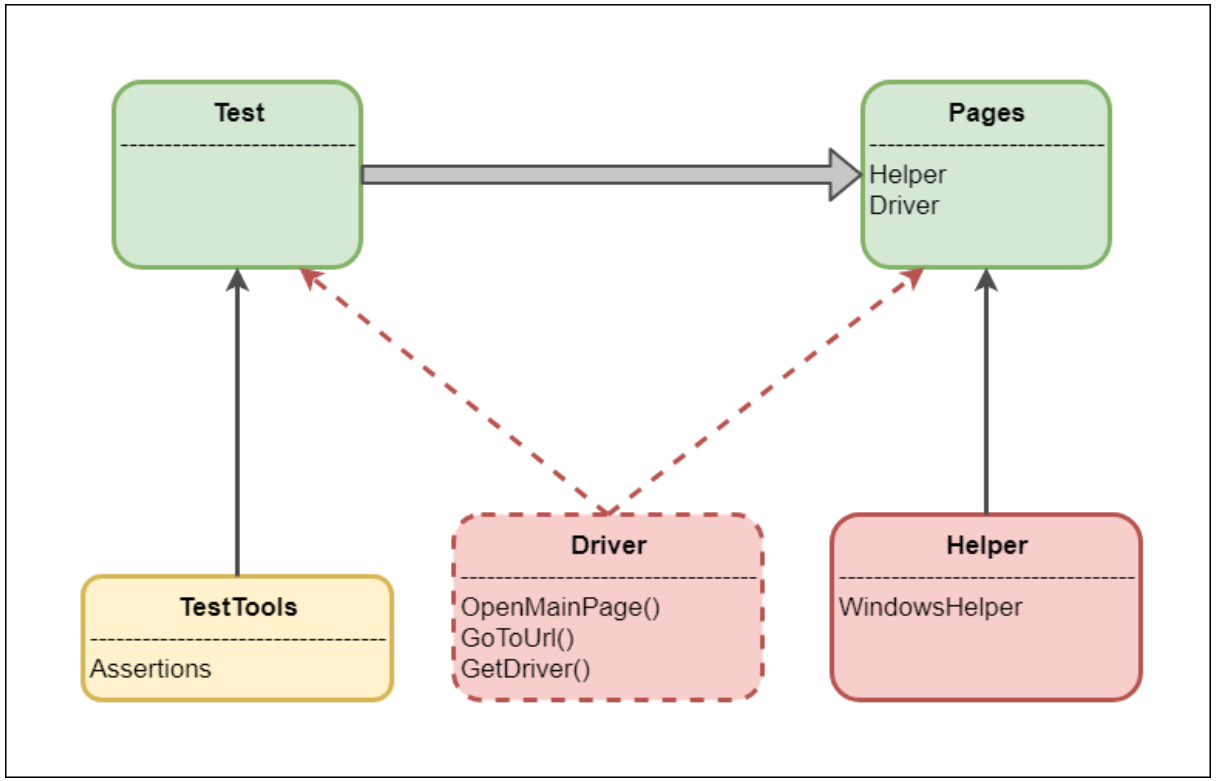
SELENIUM

ELŻBIETA SĄDEL

GRZEGORZ WITEK

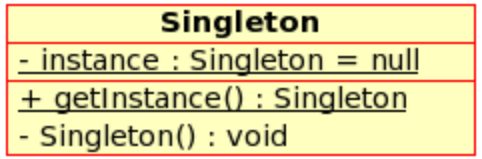
DRIVER CLASS

- Ukrycie metod selenium driver
- Obudowanie metod przez dodatkowe elementy (np. logger)
- Konfiguracja w jednym miejscu – odseparowanie testów od konfiguracji Selenium
- Przyspieszenie startowania projektu
- Ułatwienie pisania testów



DRIVER CLASS

Singleton – kreacyjny wzorzec projektowy, którego celem jest ograniczenie możliwości tworzenia obiektów danej klasy do jednej instancji oraz zapewnienie globalnego dostępu do stworzonego obiektu. Niektórzy programiści uznają go za antywzorzec, ponieważ łamie zasady projektowania obiektowego.



DRIVER CLASS

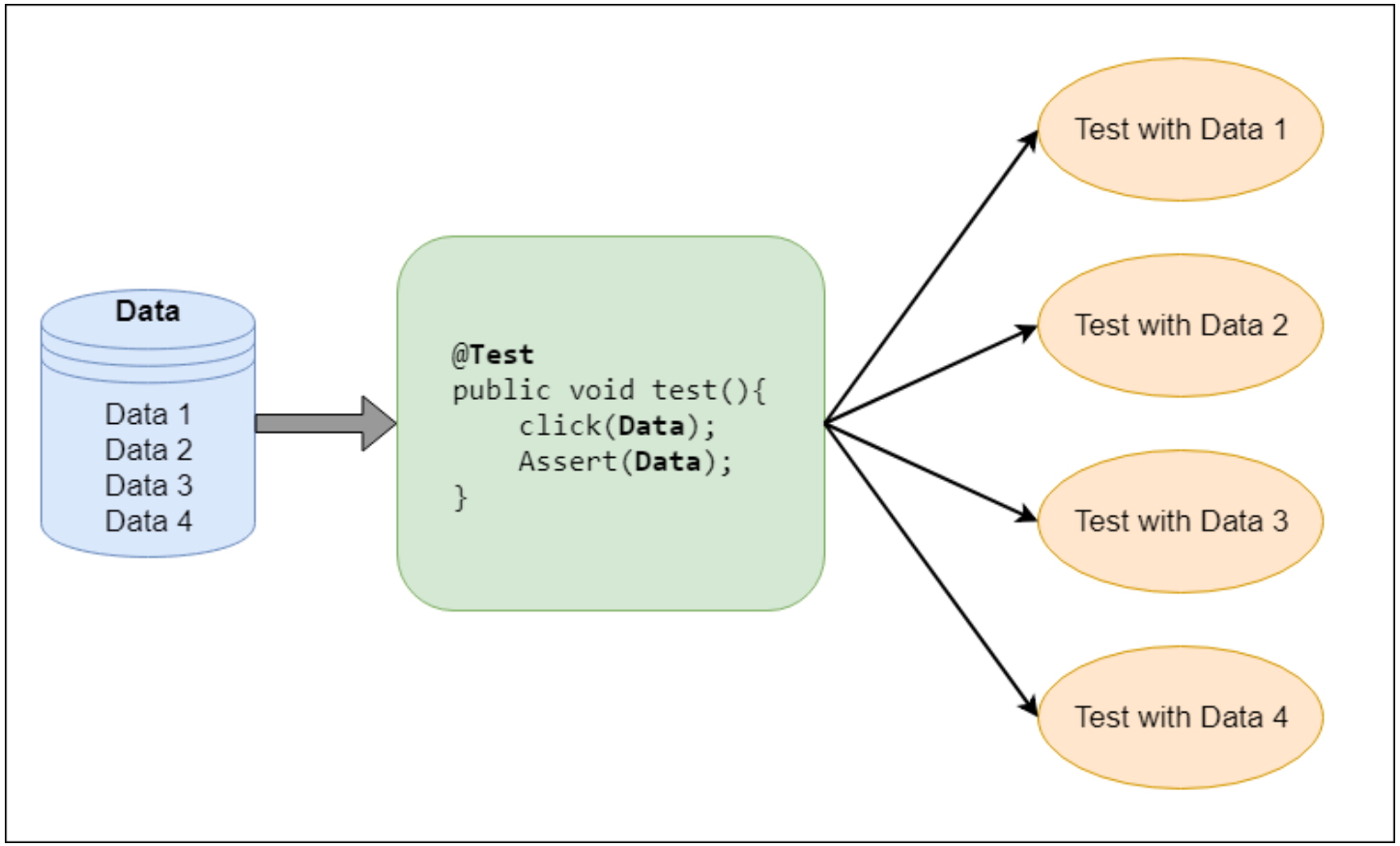
```
private static WebDriver driver;

public static WebDriver getDriver() {
    if (null == driver) {
        System.setProperty("webdriver.chrome.driver", "src/main/resources/chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait( l: 10, TimeUnit.SECONDS);
    }
    return driver;
}

public static void quit() {
    if(null != driver) {
        getDriver().quit();
    }
    driver = null;
}
```

DDT

Data-Driven Testing – koncepcja wykorzystywana w automatycznym testowaniu oprogramowania, polegająca na używaniu zbioru danych testowych jako zmienne zasilające jeden, ten sam automatyczny test w celu przetestowania większej ilości przypadków testowych.



DDT

```
@RunWith(value = Parameterized.class)
public class APLinkTest extends APBaseTest{

    @Parameter(0)
    public String targetPage;
    @Parameter(1)
    public String targetPageTitle;

    @Parameters(name = "{index}: Open social link for {0} and expect title with {1}")
    public static Collection<Object[]> data() {
        return Arrays.asList(new Object[][]{
                {"facebook", "facebook"},
                {"twitter", "twitter"},
                {"youtube", "youtube"},
                {"google-plus", "google+"}
        });
    }
```

INTERFEJS TAKESSCREENSHOT

Interfejs służący do wykonywania screenshotów.
Dokumentacja: https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/TakesScreenshot.html

Example usage:
File screenshotFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);
String screenshotBase64 = ((TakesScreenshot) driver).getScreenshotAs(OutputType.BASE64);

```
File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
String path = "c:\\SeleniumTemp\\screenshot.png";
FileUtils.copyFile(scrFile, new File(path));
```

INTERFEJS TAKESSCREENSHOT

```
private WebElement waitForElementAndReturn(By by, int timeoutInSeconds) {
    WebDriverWait wait = new WebDriverWait(driver, timeoutInSeconds);
    try{
        wait.until(ExpectedConditions.visibilityOf(driver.findElement(by)));
    }
    catch (TimeoutException e){
        File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
        Date date = new Date();
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss");
        String dateString = dateFormat.format(date);
        try {
            String path = "c:\\SeleniumTemp\\" + dateString + ".png";
            FileUtils.copyFile(scrFile, new File(path));
            System.out.println("Element with selector " + by + " was not found after " + timeoutInSeconds + "
        seconds.
            " + "Screenshot available at: " + path);
        } catch (IOException e1) {
            e1.printStackTrace();
        }
    }
    return wait.until(ExpectedConditions.visibilityOf(driver.findElement(by)));
}
```

@RULE

Pozwala na dodanie lub zmianę zachowania każdej metody testowej w klasie testowej.

```
public static class HasTempFolder {
      @Rule
      public final TemporaryFolder folder = new TemporaryFolder();

      @Test
      public void testUsingTempFolder() throws IOException {
        File createdFile = folder.newFile("myfile.txt");
        File createdFolder = folder.newFolder("subfolder");
        // ...
      }
    }
```

@RULE

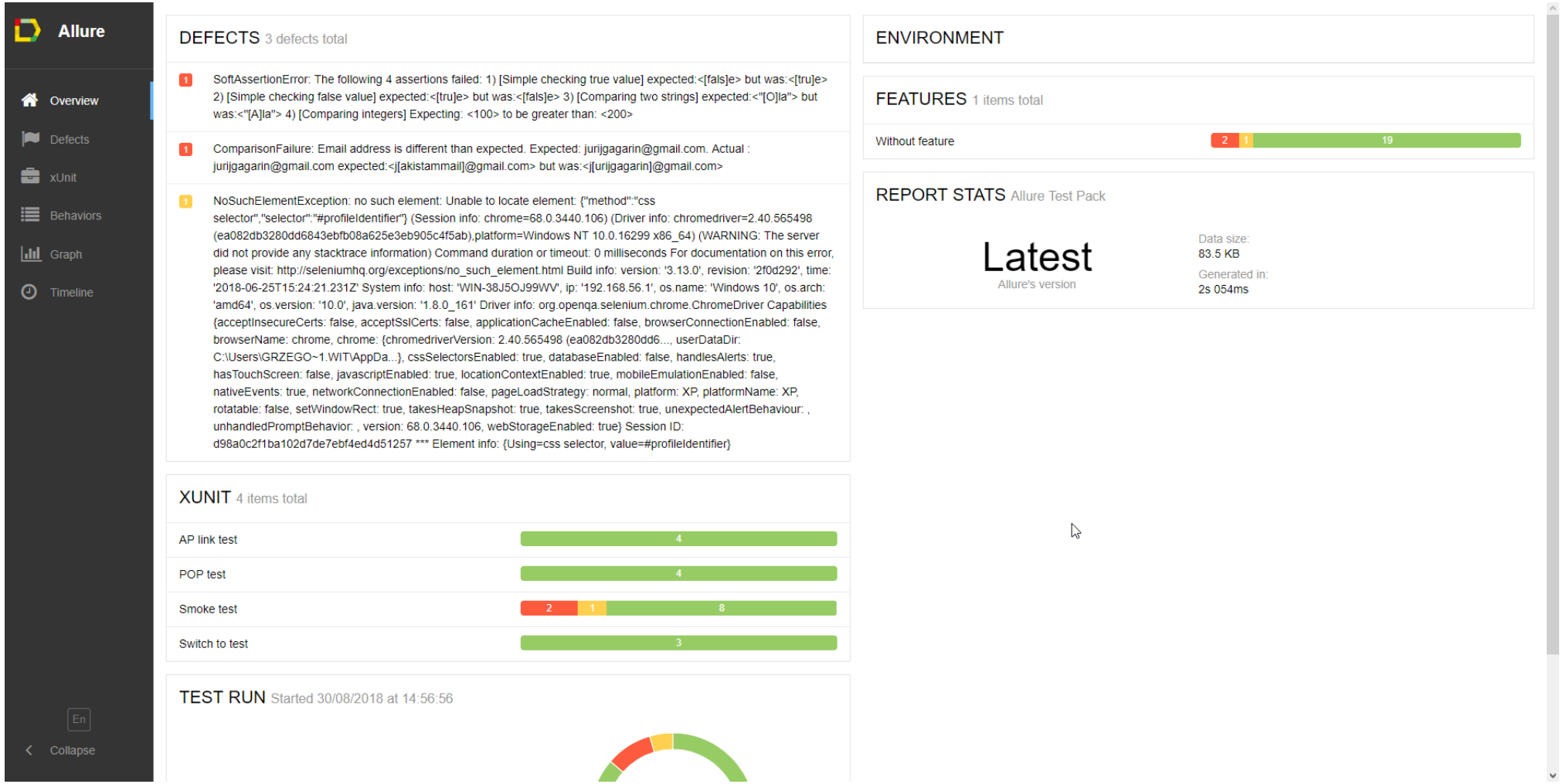
Dokumentacja:

https://github.com/junit-team/junit4/wiki/Rules#testname-rule

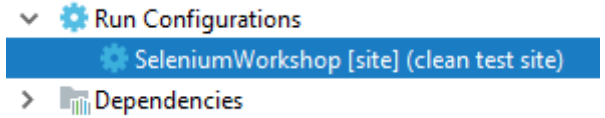
Examples of predefined rules:

- The **ErrorCollector** Rule allows execution of a test to continue after the first problem is found (for example, to collect all the incorrect rows in a table, and report them all at once).
- The **TestName** Rule makes the current test name available inside test method.
- The **Timeout** Rule applies the same timeout to all test methods in a class.

RAPORTY


Allure
Overview
Defects
xUnit
Behaviors
Graph
Timeline
En
Collapse
DEFECTS 3 defects total
1 SoftAssertionError: The following 4 assertions failed: 1) [Simple checking true value] expected:<[fals]e> but was:<[tru]e> 2) [Simple checking false value] expected:<[tru]e> but was:<[fals]e> 3) [Comparing two strings] expected:<"[O]la"> but was:<"[A]la"> 4) [Comparing integers] Expecting: <100> to be greater than: <200>
1 ComparisonFailure: Email address is different than expected. Expected: jurijgagarin@gmail.com. Actual : jurijgagarin@gmail.com expected:<j[akistammail]@gmail.com> but was:<j[urijgagarin]@gmail.com>
1 NoSuchElementException: no such element: Unable to locate element: {"method":"css selector","selector":"#profileIdentifier"} (Session info: chrome=68.0.3440.106) (Driver info: chromedriver=2.40.565498 (ea082db3280dd6843ebfb08a625e3eb905c4f5ab),platform=Windows NT 10.0.16299 x86_64) (WARNING: The server did not provide any stacktrace information) Command duration or timeout: 0 milliseconds For documentation on this error, please visit: http://seleniumhq.org/exceptions/no_such_element.html Build info: version: '3.13.0', revision: '2f0d292', time: '2018-06-25T15:24:21.231Z' System info: host: 'WIN-38J5OJ99WV', ip: '192.168.56.1', os.name: 'Windows 10', os.arch: 'amd64', os.version: '10.0', java.version: '1.8.0_161' Driver info: org.openqa.selenium.chrome.ChromeDriver Capabilities {acceptInsecureCerts: false, acceptSslCerts: false, applicationCacheEnabled: false, browserConnectionEnabled: false, browserName: chrome, chrome: {chromedriverVersion: 2.40.565498 (ea082db3280dd6..., userDataDir: C:\Users\GRZEGO~1.WIT\AppDa...}, cssSelectorsEnabled: true, databaseEnabled: false, handlesAlerts: true, hasTouchScreen: false, javascriptEnabled: true, locationContextEnabled: true, mobileEmulationEnabled: false, nativeEvents: true, networkConnectionEnabled: false, pageLoadStrategy: normal, platform: XP, platformName: XP, rotatable: false, setWindowRect: true, takesHeapSnapshot: true, takesScreenshot: true, unexpectedAlertBehaviour: , unhandledPromptBehavior: , version: 68.0.3440.106, webStorageEnabled: true} Session ID: d98a0c2f1ba102d7de7ebf4ed4d51257 *** Element info: {Using=css selector, value=#profileIdentifier}
XUNIT 4 items total
AP link test 4
POP test 4
Smoke test 2 1 8
Switch to test 3
TEST RUN Started 30/08/2018 at 14:56:56
ENVIRONMENT
FEATURES 1 items total
Without feature 2 1 19
REPORT STATS Allure Test Pack
Latest
Allure's version
Data size:
83.5 KB
Generated in:
2s 054ms

RAPORTY




```
<reporting>
    <excludeDefaults>true</excludeDefaults>
    <plugins>
        <plugin>
            <groupId>ru.yandex.qatools.allure</groupId>
            <artifactId>allure-maven-plugin</artifactId>
            <version>2.6</version>
            <configuration>
                <reportDirectory>${basedir}/target/allure-report</reportDirectory>
            </configuration>
        </plugin>
    </plugins>
</reporting>
```

RAPORTY + RULE

```
@Override
protected void failed(Throwable e, Description description) {
    takePageSource();
    makeScreenshot();
}

@Override
protected void finished(Description description) { Driver.quit(); }

@Attachment("Screenshot on failure")
private byte[] makeScreenshot() {
    return ((TakesScreenshot) Driver.getDriver()).getScreenshotAs(OutputType.BYTES);
}

@Attachment("Page Source")
private String takePageSource() { return (Driver.getDriver()).getPageSource(); }
```

RULE

```java
public class MasterTest {
    @Rule
    public TestRules ruleExample = new TestRules();
}
```

```java
public class SmokeTest extends MasterTest {

    String emailAddress = "jurijgagarin@gmail.com";
    String gmailURL = "https://www.gmail.com";

    @Test
    public void openPageTest(){
        Driver.openPage( url: "http://automationpractice.com");
        Assert.assertEquals( message: "Wrong page title",
                expected: "My Store", Driver.getTitle());
    }
```

DRIVERMANAGER

```xml
<dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>2.2.4</version>
</dependency>
```

```java
public static WebDriver getDriver() {
    if (null == driver) {
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait( l: 10, TimeUnit.SECONDS);
    }
    return driver;
}
```

DZIĘKUJEMY