



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Факультет компьютерных наук
Департамент программной инженерии

ОБЕСПЕЧЕНИЕ КАЧЕСТВА И ТЕСТИРОВАНИЕ

Семинар 9: Автоматизированное
тестирование пользовательского
интерфейса. Настольные приложения.
AssertJ.

Москва, 2020



ИНСТРУМЕНТЫ АВТОТЕСТИРОВАНИЯ

- Симулируют действия пользователя на GUI и проверяют состояние приложения
- Тесты на популярных языках программирования (Python, Java, C#, Visual Basic и т.д.)
- Тесты на предметно-ориентированных графических языках (keyword-driven testing)
- Коммерческие инструменты и инструменты с открытым исходным кодом
- Существующие инструменты могут сильно отличаться по своим возможностям



ПРОБЛЕМЫ ИНСТРУМЕНТОВ АВТОТЕСТИРОВАНИЯ

- Трудоемкость разработки тестов
- Проблемы с распознаванием элементов UI и доступа к их состоянию
- Сложности поддержки тестов: поломки при изменениях в UI
- Сложности переноса тестов в новое окружение



КОММЕРЧЕСКИЕ ИНСТРУМЕНТЫ

- Ranorex
- TestComplete
- Micro Focus Unified Functional Testing
- Silk Test



БЕСПЛАТНЫЕ ИНСТРУМЕНТЫ

- Autolt и Actonia
- **AssertJ Swing**
- IntelliJ UI Test Robot
- и многие другие



ASSERTJ SWING

- Библиотека с открытым исходным кодом для тестирования Java Swing приложений
- Интегрируется с JUnit и TestNG
- Позволяет запускать приложения, распознавать стандартные элементы UI, симулировать действия пользователя с ними и проверять их состояние
- Расширяемая библиотека



ASSERTJ SWING – ЗАПУСК ПРОГРАММЫ

```
final Robot robot = BasicRobot.robotWithNewAwtHierarchy();  
ApplicationLauncher.application(SimpleApp.class).start();  
window = WindowFinder.findFrame("SimpleApp").withTimeout(10000).using(robot);
```



ASSERTJ SWING – ДЕЙСТВИЯ С ЭЛЕМЕНТАМИ UI

```
window.textBox("textToCopy").enterText("Some random text");
```

```
window.button("copyButton").click();
```

```
window.label("copiedText").requireText("Some random text");
```



ASSERTJ SWING – РАСПОЗНАНИЕ ЭЛЕМЕНТОВ UI

// by name

```
ButtonFixture button = window.button("login");
```

// by type

```
ButtonFixture button = window.button();
```

// custom search criteria (the button's text)

```
ButtonFixture button = window.button(new GenericTypeMatcher<JButton>(JButton.class) {  
    @Override  
    protected boolean isMatching(JButton button) {  
        return "Login".equals(button.getText());  
    }  
});
```



ASSERTJ SWING – ПОИСК ЭЛЕМЕНТОВ UI В ИЕРАРХИИ

```
// new LoginFrame();
```

```
ComponentFinder finder = BasicComponentFinder.finderWithNewAwtHierarchy();
```

```
finder.findByName("login", true); // will fail finding component of login frame
```

```
// new MainFrame();
```

```
finder.findByName("pw", true); // will work finding label of main frame
```



ЛИТЕРАТУРА

1. AssertJ Swing - <http://joel-costigliola.github.io/assertj/assertj-swing.html>
2. Инструменты тестирования графического интерфейса - https://en.wikipedia.org/wiki/Comparison_of_GUI_testing_tools



HOMEWORK 7

- Используя фреймворк AssertJ Swing, разработать GUI-тесты для приложения Calculator на JUnit или TestNG.
- Тесты должны обеспечивать полное покрытие кода (JaCoCo) и демонстрировать ошибку в реализации.
- Оценить тестируемость приложения. Кратко описать проблемы, которые возникли при разработке тестов и которые могут возникнуть в будущем из-за изменений в GUI приложения. Кратко 2-5 предложений.
- Изменять код приложения Calculator для улучшения тестируемости не разрешается.

СПАСИБО! ВОПРОСЫ?



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ