

# Úvod do automatizace v testování + Low code automation

Jiří Koudelka

17.10.2022

Brno - Online



- Lektor programování
- Vysokoškolský pedagog
- Puntičkář
- Komunikativní a přátelský

**Jiří Koudelka**

*Software Architect*

# CÍL LEKCE

Ukázat si co jsou, k čemu jsou a jak se dají tvořit automatizované testy.

# OSNOVA

- Co je automatizace?
- Co je low-code automatizace?
- Selenium
- Spuštění testů
- Psaní vlastních testů

# AUTOMATIZACE V TESTOVÁNÍ

"Je používání jakéhokoliv nástroje v průběhu testování"

# VÝHODY VYUŽITÍ AUTOMATIZACE V TESTOVÁNÍ

- Šetří čas
- Snižuje pravděpodobnost chyby při vykonávání testu
- Umožňuje "nekonečně" mnoho spouštění testů (24/7)
- Pozitivní vliv na regresní testování
- Dostáváme zpětnou vazbu na testovaný produkt často a rychle

# NEVÝHODY VYUŽITÍ AUTOMATIZACE V TESTOVÁNÍ

- Je náročná na technické znalosti testerů
- Správa/údržba zabírá mnoho času
- Tester ztrácí spojení s testovaným systémem
- Testy časem odhalují čím dál méně chyb
- "Není" to testování

# ZÁKLADNÍ PRINCIPY V AUTOMATIZACI

- Zvažte poměr mezi časem vývoje jednoho testu (cena) a počtem jeho možných spuštění (výkon)
- Zaměřte se na happy path cases
- Automatizujte regresní testy
- Když vložíte úsilí do jakékoliv automatizace, využijte její potenciál
- Využijte čas, který získáte automatizováním scénářů k exploraci
- Nezapomeňte na to, že jste testeři, ne opraváři automatických testů



# MANUÁLNÍ VS AUTOMATIZOVANÉ TESTY

- Nejdou proti sobě, ale vedle sebe
- Většina testů je v dnešní době stále manuální
- Není reálné kompletně nahradit manuální testy automatizovanými

# DOTAZY

Máme základní představu, co je to automatizace testování?

# LOW-CODE AUTOMATION

# LOW-CODE AUTOMATION

- Způsob testování aplikace rychle pomocí naplánování, generování a spouštění testů
- Tester nepotřebuje rozsáhlé znalosti programování
- Nutno mít k dispozici takzvaný "framework"
- Testujeme na UI vrstvě

# VÝHODY LOW-CODE AUTOMATION

- Jednoduché na naučení a použití
- Zvyšuje produktivitu
- Oproti klasické automatizaci zvyšuje rychlost reakce na změny

# NEVÝHODY LOW-CODE AUTOMATION

- Pro komplexnější testovací scénáře může být pomalejší
- Pro větší změny je zapotřebí zkušenějšího vývojáře

# LOW-CODE AUTOMATION FRAMEWORK

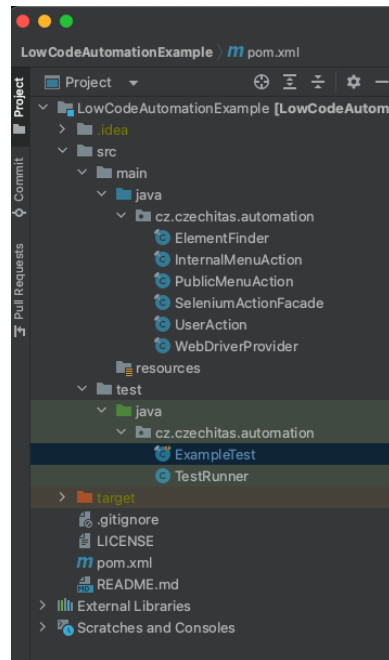
# LOW-CODE AUTOMATION FRAMEWORK

- Funguje všem spuštění ukázkového testu?



# IntelliJ Idea IDE

- Co je to IDE?
- Navigace v projektu
- Spuštění ukázkového testu



# Knihovná funkce

- Předpřipravený set akcí
- Připravuje programátor nebo zkušený tester
- Slouží k provádění jednotlivých kroků testů

```
prohlizec.;
prohliz m jdiDoSekceKontakt () void
publicMenuAction PublicMenuAction
m vyplnEmail (String email) void
m overAdresuWwwStranky (String wwwAdresa) void
st m klikniNaTlacitkoPrihlasit () void
d overUs m provedOdhlaseni () void
m jdiDoSekceExporty () void
prohliz m overPrihlaseniUzivatele () void
prohliz m provedPrihlaseni () void
prohliz m vyplnHeslo (String heslo) void
prohliz m jdiDoSekceAktuality () void
prohliz m jdiDoSekceDomu () void
Thread
```

## Přidání vlastního testu

- Vycházíme ze vzoru ve třídě ExampleTest
- Přidáme si vlastní metodu s anotací @Test
- Používáme předchystané knihovní funkce

```
/**  
 * Test class for custom student tests  
 *  
 * @author Jiri Koudelka  
 * @since 1.0.0  
 */  
final class LowCodeAutomationTest extends TestRunner {  
  
}
```

# Krokování (Debugging) testu

- Co je breakpoint?
- Jak ho do programu přidám?
- Jak spustím test, který breakpoint zohledňuje?



The screenshot shows a code editor with a dark background. On the left side, there is a vertical toolbar with icons for running, debugging, and breakpoints. A red circle breakpoint is placed on the line `firefox.overAdresuWwwStranky(`. The code is as follows:

```
@Test
void overKontaktNiWwwAdresu() {
    firefox.jdiDoSekceKontakt();
    firefox.overAdresuWwwStranky( wwwAdresa: "www.czechitas.cz");
}
```

ASSERTACE

# ASSERTACE

- Assert = Tvrzení -> v našem kontextu vyhodnocení, neboli očekávané chování
- Naše automatizované testy umí zatím jen klikat, vyplňovat a navigovat
- Jak takový assert do našich testů přidat?

# DRUHY ASSERTACÍ

- Obecně můžeme testovat kroky, které nám vyhodnotí např:
  - Přítomnost elementu na stránce
  - Nepřítomnost elementu na stránce
  - Konkrétní hodnotu v elementu
  - Konkrétní text zobrazený v daném elementu
  - Že je/není možné pole editovat
  - A mnoho dalších...
- Záleží jen na tom, co nám vývojaři přidají do frameworku

# PŘIDÁNÍ ASSERTACE DO TESTU

- Logicky odděleno od navigace v prohlížeči
- Často požaduje na vstupu parametr

```
@Test
void overKontaktNiWwwAdresu() {
    prohlizec.jdiDoSekceKontakt();
    overeni.overAdresuWwwStranky( wwwAdresa: "www.czechitas.cz");
}
```



Praktické úkoly

# SAMOSTATNÁ PRÁCE

- Pracujte samostatně v breakout rooms
- K dispozici máte kouče
- Příklady naleznete v PDF dokumentu ve složce **practice**

