

# Úvod do automatizace v testování

Jiří Koudelka

Brno

# Obsah lekce

- Co je to automatizace v testování?
- Základy automatizace webových aplikací
- Ochutnávka o čem to automatizované testování je
- Spouštění existujících testů
- Psaní vlastních testů

**Co je to automatizace?**



# Automatizace obecně

- Strojové vykonávání činností které by jinak vykonával člověk
- Na základě vstupů vznikají nějaké výstupy
  - Vstupy mohou být vždy stejné nebo se mohou měnit
- Automatizovaná činnost by *měla být* idempotentní
  - = pro stejné vstupy produkuje vždy stejné výstupy

# Automatizace v testování

- Automatizace testovacích scénářů
  - Smoke testy, regresní testy, akceptační testy, ...
- Automatizace repetitivních činností souvisejících s testováním
  - Příprava dat
  - Nastavování prostředí
- Automatizace na různých platformách
  - Webové, mobilní, desktopové, embedded
- Opakování stejných testů
  - na různých prostředích
  - v různých prohlížečích
  - na různých verzích

# Výhody využití automatizace v testování

- Šetří čas při **vykonávání** testů
- Snižuje pravděpodobnost chyby při **vykonávání** testu
- Umožňuje kontinuální testování (24/7)
  - teoreticky nekonečné spouštění testů
- Dostáváme **feedback** na testovaný produkt **často a rychle**
  - = pozitivní vliv na regresní testování

# Nevýhody využití automatizace v testování

- Je **náročná na technické** znalosti testerů
- Údržba testů může zabírat mnoho času
- Může produkovat **false positives** i **false negatives**
- Tester může ztrácet spojení s testovaným systémem a doménou
  - Záleží na tom kdo a kdy automatizuje
- Testy s časem odhalují čím dál méně chyb
  - Náchylné na tzv. **pesticidní paradox**
- **Automatizace není testování a nemůže nahradit testování**
  - testing vs. checking
  - Je možné provádět tool assisted testing

# Jaké typy testů můžeme automatizovat?

## V podstatě všechny typy!

- Funkcionální
  - Testovací scénáře = regresní testy, akceptační testy, ...
- Nefunkcionální
  - Usability, responsivity, compatibility, performance, ...
- Dynamické i statické
  - Automatická kontrola dokumentace, kontrola dodržování standardů (compliance)
- Pomoc při manuálním testování
  - Tool assisted testing, generování dat, vstupů, zátěže, ...



# 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

# Co automatizovat?

- Poměr náročnosti vývoje testu a počet jeho možných spuštění = **cena vs. výkon**
- **Náročnost údržby**
  - vs. effort pro manuální testování
  - vs. práce s opravou automatizace
- **Rizika** spojená s testovaným software
  - náchylnost testovaného software k regresí a závažnost důsledků vzniku regrese
  - četnost změn v testovaném software
- **Testovací data**
  - Dostupnost testovacích dat
  - Náročnost údržby testovacích dat
- Závislosti na dalších systémech

# Chcete automatizovat testy, které...

- Vám **rychle dají informace** o celkovém stavu software a případné regresi
  - Regresní testy, smoke testy, happy path scénáře
- Potvrdí že **kritické části programu** fungují
- Vykonnávají se **často** a nejsou příliš složité
  - nebudou mít tendenci reportovat falešné výsledky
  - nebudeme je často opravovat
- Komplexní testy, které je naopak **složité vykonávat manuálně**
  - Pomůže alespoň částečná automatizace
  - Hrozí, že bude nákladné je udržovat, ale stále to může být levnější než manuální testování

# Nechcete automatizovat testy, které...

- Testy, jichž **vykonávání je** rychlé **a nenáročné** ale automatizace a **údržba by náročná byla**
- Nejdou automatizovat tak, aby jejich vykonávání nebo vyhodnocení bylo **spolehlivé**
- Scénáře u kterých nemáme dostatečnou **kontrolu nad vstupy a výstupy**
  - Nepředvídatelná data
  - V některých případech integrace 3rd party systémů

# Kdy automatizovat?

- Ve vývoji obecně platí, že čím menší odstup od uživatele tím lépe pro kvalitu software. Psaní testů není výjimkou.
- Je dobré automatizovat v co nejmenším odstupě od vývoje, v ideálním případě paralelně s vývojem, v jednom sprintu, atd...
  - Čím delší doba uplyne
    - tím více se ztrácejí znalosti a informace
    - snižuje se ochota vývojářů něco měnit pro lepší funkci automatizace
  - Automatizovat lze začít i před vývojem
    - Lze si připravit scénáře, a jejich vykonávání naprogramovat později
    - Lze použít tvz. mocky (více pokročilé)

# Kdy automatizovat - realita

- Testování a automatizace jsou ve firmách někdy oddělené
- Často existují oddělené teamy pro testování a pro pozdější automatizaci testů
- Ztrácí se kontakt s uživatelem i s vývojem, testy mohou být dokonce škodlivé!
  - Špatné pochopení scénáře může vést ke vzniku testu který ověřuje chybný výsledek jako správný
  - Tendence sklouzávat k technickému pohledu na automatizované testy
  - Automatické testy jsou degradovány na technicky orientované scénáře místo uživatelských scénářů

# Další funkce automatizace

- **Dokumentace**

- Dobře napsané testy krásně dokumentují uživatelské požadavky v podstatě na příkladech

- **Reporting**

- Většina testovacích frameworků je generuje

- **Monitoring**

- Automatizované scénáře lze použít na monitoring zdraví systémů

# Low-Code Automation

The background features a gradient from light pink to deep magenta. On the left, there's a rectangular area with a light pink dotted pattern. On the right, a large, semi-circular shape contains a white line-art illustration of a person's head and shoulders, pointing towards the left. Below this, another semi-circular shape contains a white dotted pattern. In the bottom right corner, there's a small circular shape with diagonal white stripes.



# Low-Code Automation

- Způsob testování aplikace rychle pomocí plánování, generování a spouštění testů
- Tester nepotřebuje rozsáhlé znalosti programování
- Nutné mít k dispozici takzvaný “framework”
- Testujeme na UI vrstvě

# Výhody Low-Code Automation

- Jednoduché na naučení a použití
- Zvyšuje produktivitu
- Oproti Full-Code 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

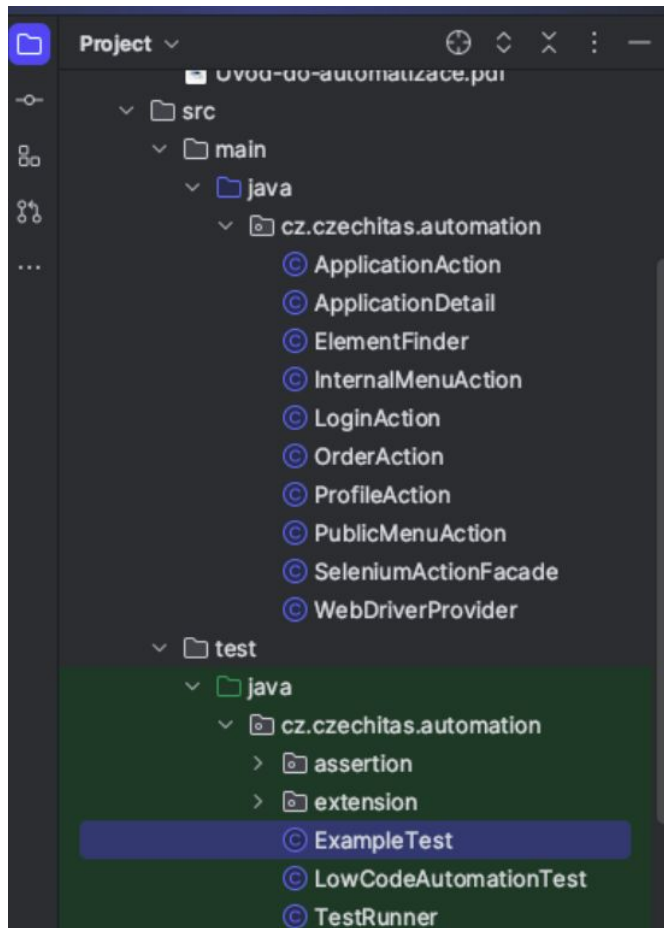


# Low-Code Automation Framework

- Funguje všem spuštění ukázkového testu?

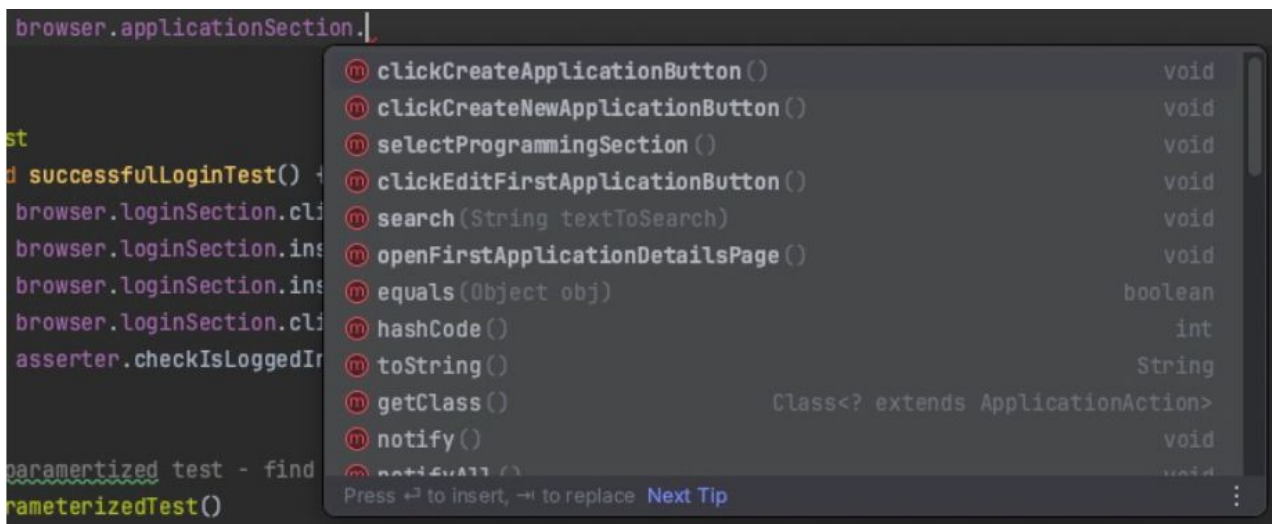
# IntelliJ Idea IDE

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



# Knihovní funkce

- Předpřipravený set akcí (dokumentovaný v README.md)
- Připravuje programátor nebo zkušený tester
- Slouží k provádění jednotlivých kroků testů



The screenshot shows an IDE with a code completion menu open for the text `browser.applicationSection.`. The menu lists several methods with their return types:

- `clickCreateApplicationButton()` void
- `clickCreateNewApplicationButton()` void
- `selectProgrammingSection()` void
- `clickEditFirstApplicationButton()` void
- `search(String textToSearch)` void
- `openFirstApplicationDetailsPage()` void
- `equals(Object obj)` boolean
- `hashCode()` int
- `toString()` String
- `getClass()` Class<? extends ApplicationAction>
- `notify()` void
- `notifyAll()` void

At the bottom of the menu, it says "Press → to insert, ⇧ to replace" and "Next Tip".

## 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 přidat?
- Jak spustit test, který breakpoint zohledňuje?

```
15      @Test
16      void contactsPageUrlTest() {
17          browser.headerMenu.goToContactsSection();
18           asserter.checkPageUrl("www.czechitas.cz");
19      }
```

The background is a solid pink color. In the top-left corner, there is a rectangular area with a light pink dotted pattern. In the bottom-left corner, there is a circular area with a light pink dotted pattern. In the bottom-right corner, there is a circular area with a light pink dotted pattern. In the top-right corner, there is a circular area with a light pink dotted pattern. In the center-right area, there is a large, light pink circular shape. Inside this shape, there is a line-art illustration of a woman's head and shoulders, facing right. The woman has short, wavy hair and is wearing a top with a large, pointed collar. The overall design is modern and minimalist.

# 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 například:
  - 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 vývojáři přidají do frameworku

## Přidání assertace do testu

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

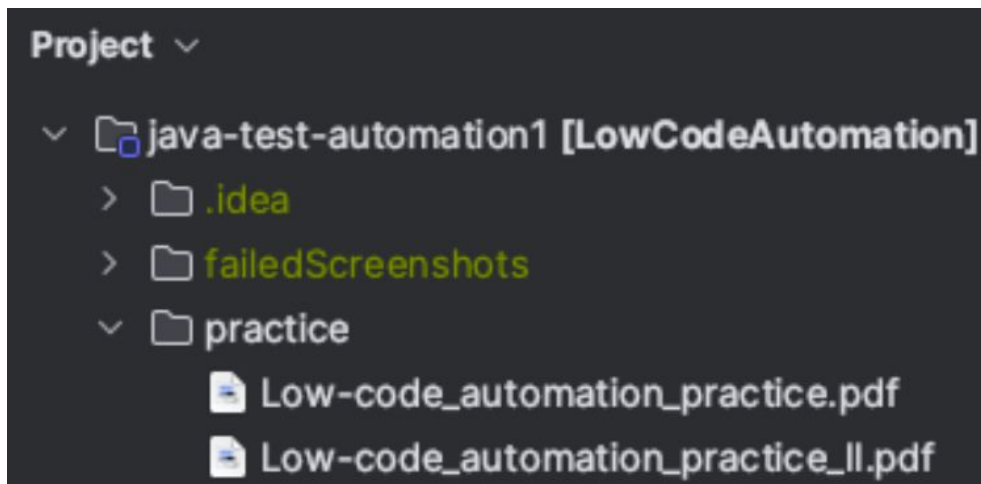
```
@Test
void contactsPageUrlTest() {
    browser.headerMenu.goToContactsSection();
    asserter.checkPageUrl("www.czechitas.cz");
}
```

# Praktické úkoly



## Samostatná práce

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





Děkuji za pozornost