Python a Robot Framework

Den třetí

ENGETO s.r.o.

Prezentované materiály jsou dostupné na adrese
https://github.com/ChaoticRoman/python-a-robot

Osnova kurzu

1. den: Filosofie, instalace a základy Pythonu, Python 2 a 3, standardní knihovna, objektově orientované programování, malé projekty s knihovnami numpy, matplotlib a Tkinter, distribuce aplikací pomocí setuptools a distutils

2. den: Středně velké projekty, organizace kódu, dokumentace, stanovení požadavků, teorie softwarového testování, unittesting, doporučené praktiky

3. den: Větší projekty, softwarové testování: integrační testování, systémové testování, akceptační testování, TDD, BDD, Robot Framework: základy

4. den: Kooperace pomocí gitu, logování, standardy, společný projekt,
Robot Framework: pokročilé funkce a doporučené praktiky

5. den: Testování a mocking, Robot Framework: uživatelská rozšíření, společný projekt

Den třetí

- Větší projekty
- Software testing: unittesting, integration testing, system a end-to-end testing, acceptance testing
- Test driven development (TDD), behavior driven testing (BDD), behavior driven development (BDD)
- Robot Framework: základy

Větší projekty

- >10 kLOC
- Lepší menší! vs. Lepší velký projekt než dva nebo tři malé!
  - KISS, YAGNI, DRY
- SOLID:
  - Single responsibility principle, Open-closed principle, Liskov substitution principle, Interface segregation principle (God class antipattern), Dependency inversion principle
- Abstrakční vrstvy: HAL, OSAL, …
- ~1 chyba / 1000 řádků → testování se stává klíčové

Větší projekty

- Antipatterns:
  - http://www.laputan.org/mud/
  - Data duplication
  - Data promiscuity (God class)
  - High coupling
    - Preferujte skládání před děděním objektů
    - Menší vs. jednoduché signatury

Software testing

- Unittesting
- Integration testing
- System a end-to-end testing
- Acceptance testing

Co testujeme: Funkce, použitelnost, výkon, škálovatelnost, limity, bezpečnost, kompatibilita, spolehlivost (robustnost), portabilita (vč. instalace)

Přístupy: waterfall, agile, TDD, BDD, ATDD

Jiné typy verifikace: statická analýza, review, inspekce, podobnost

Software testing

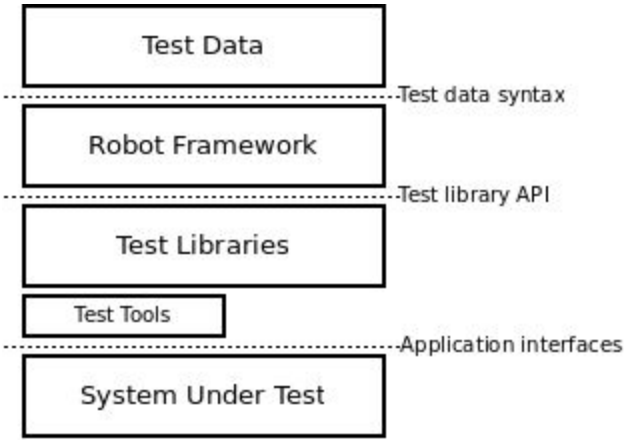
- Unittesting a integration testing v Pythonu
  - From scratch
  - Unittest knihovna

Hands-on: testování čistých funkcí a funkcí se side-efekty

Robot Framework: základy

- Obecný framework pro automatizaci testů
- Keyword-based
- Podporuje Gherkin specifikaci scénářů (Given-When-Then)
- Počátky: Pekka Klärck, 2005-2008
- Apache License 2.0
- Kvalitní dokumentace
- Důraz na BDD, ATDD a Data-driven testing
- Vstupní formát: plain text, TSV, HTML, ReStructured text, Excel spreadsheet
- Výstupní formát: XML, HTML
- Integrace: bash, Jenkins, …
- Snadná rozšířitelnost: high-level keywords, uživatelské knihovny, XMLRPC
- Tagy, Templates,
- RIDE

Robot Framework: základy



Zdroj: Robot Framework User Guide,

http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html

Robot Framework: základy

Oficiální dokumentace

Hands-on:

- Hello world!
- Timeout
- Bash command status
- High level keywords, dokumentace, tagy
- Small test suite organization

Robot Framework: základy

- Built-in Keywords
- Standardní knihovna

http://robotframework.org/robotframework/#standard-libraries

Robot Framework: základy

Doporučené postupy

- Doména problému, ne robota nebo programovacího jazyka
- Abstraktní x konkrétní x technická klíčová slova
- Popis “co” ne “jak”
- Proveď → Zkontroluj
- Separace scénáře a dat

Den třetí: rekapitulace

- Větší projekty
- Software testing: unittesting, integration testing, system a end-to-end testing, acceptance testing
- Test driven development (TDD), behavior driven testing (BDD), behavior driven development (BDD)
- Robot Framework: základy