

Programowanie sterowane testami w Pythonie : jak stworzyć skalowalne zestawy testów i aplikacji / Alessandro Molina. – Gliwice, © 2022

Spis treści

O autorze	9
O korektorze merytorycznym	10
Wprowadzenie	11
Część 1. Testowanie oprogramowania i programowanie sterowane testami	15
Rozdział 1. Rozpoczęcie pracy z testowaniem oprogramowania	17
Wymagania techniczne	18
Wprowadzenie do testowania oprogramowania i kontroli jego jakości	18
Plan testów	19
Wprowadzenie do testów automatycznych i zbiorów testów	21
Wiele zbiorów testów ²³	
Grupowanie testów	24
Wprowadzenie do programowania sterowanego testami i testów jednostkowych	26
Programowanie sterowane testami	27
Testy jednostkowe	30
Testy integracji i testy funkcjonalne	32
Testy integracji	32
Testy funkcjonalne	35
Piramida testów i trofeum testów	36
Model piramidy testów	37
Model trofeum testów	38
Rozkład testów i pokrycie kodu testami	39
Podsumowanie	40
Rozdział 2. Dublery używane podczas testów na przykładzie aplikacji czatu	41
Wymagania techniczne	42
Wprowadzenie do dublerów używanych podczas testów	42
Aplikacja czatu opracowana z wykorzystaniem modelu TDD	44
Stosowanie obiektów pozornych	48
Zastępowanie komponentów ich namiastkami	50
Sprawdzanie zachowania za pomocą szpiegów	54
Stosowanie imitacji	58

Zastępowanie zależności atrapami	60
Testy akceptacji i dublery używane podczas testów	64
Zarządzanie zależnościami za pomocą mechanizmu wstrzykiwania zależności	67
Stosowanie frameworków mechanizmu wstrzykiwania zależności	70
Podsumowanie	72
Rozdział 3. Programowanie sterowane testami na przykładzie aplikacji listy rzeczy do zrobienia	74
Wymagania techniczne	75
Rozpoczęcie pracy nad projektem wykorzystującym model TDD	75
Tworzenie aplikacji z użyciem modelu TDD	88
Zapobieganie regresji	108
Podsumowanie	115
Rozdział 4. Skalowanie zbioru testów	116
Wymagania techniczne	116
Skalowanie testów	117
Przejście od testów E2E do testów funkcjonalnych	123
Praca z wieloma zbiorami testów	126
Zbiór testów kompilacji	126
Testy podczas przekazywania kodu źródłowego do repozytorium	127
Testy dymne	128
Przeprowadzanie testów wydajności	129
Włączanie ciągłej integracji	131
Testowanie wydajności w chmurze	135
Podsumowanie	136
Część II. Framework pytest	137
Rozdział 5. Wprowadzenie do frameworka pytest	139
Wymagania techniczne	139
Wykonywanie testów za pomocą frameworka pytest	140
Definiowanie warunków początkowych testów pytest	143
Stosowanie warunków początkowych testu i mechanizmu wstrzykiwania zależności	147
Stosowanie argumentu tmp path do zarządzania danymi tymczasowymi	148
Stosowanie wtyczki capsys do testowania wejścia - wyjścia	149
Wykonywanie podzbioru testów	150
Podsumowanie	152
Rozdział 6. Testy parametryzowane i dynamiczna konfiguracja testów	153
Wymagania techniczne	154
Konfiguracja zbioru testów	154

Generowanie konfiguracji	156
Generowanie testów parametryzowane	161
Podsumowanie	163

Rozdział 7. Funkcje dopasowania na przykładzie aplikacji książki adresowej **164**

Wymagania techniczne	165
Tworzenie testów akceptacji	165
Zdefiniowanie pierwszego testu	166
Informacje pochodzące od zespołu odpowiedzialnego za produkt	167
Zaliczenie testów	169
Stosowanie modelu programowania sterowanego zachowaniem	172
Definiowanie pliku funkcjonalności	173
Deklarowanie scenariusza	173
Wykonanie scenariusza testowego	174
Dalsza konfiguracja z użyciem kroku And	175
Wykonywanie akcji za pomocą kroku When	176
Ocena warunku za pomocą kroku Then	177
Zaliczenie scenariusza	177
Uwzględnianie specyfikacji na przykładzie	179
Podsumowanie	185

Rozdział 8. Najważniejsze wtyczki dla frameworka pytest **186**

Wymagania techniczne	187
Stosowanie wtyczki pytest-conv do generowania informacji dotyczących stopnia pokrycia kodu źródłowego testami	187
Pokrycie testami jako usługa	191
Stosowanie wtyczki pytest-benchmark do przeprowadzania testów wydajności	193
Porównywanie wyników testów wydajności	195
Stosowanie wtyczki flaky do ponownego wykonywania niepewnych testów	196
Stosowanie wtyczki pytest-testmon do ponownego wykonywania testów po wprowadzeniu zmiany w kodzie	199
Jednoczesne wykonywanie testów za pomocą wtyczki pytest-xdist	201
Podsumowanie	203

Rozdział 9. Zarządzanie środowiskami testowymi za pomocą narzędzia Tox **204**

Wymagania techniczne	205
Wprowadzenie do narzędzia Tox	205
Testowanie wielu wersji Pythona za pomocą narzędzia Tox	208
Stosowanie środowisk dla więcej niż jednej wersji Pythona	210
Stosowanie narzędzia Tox w połączeniu z usługą Travis CI	212
Podsumowanie	215

Rozdział 10. Testowanie dokumentacji i testowanie na podstawie właściwości	217
Wymagania techniczne	218
Testowanie dokumentacji	218
Dodawanie przewodnika po kodzie aplikacji	220
Tworzenie zweryfikowanego podręcznika użytkownika	222
Testowanie na podstawie właściwości	227
Generowanie testów przeznaczonych dla najczęściej stosowanych właściwości	232
Podsumowanie	234
Część III. Testowanie aplikacji internetowych	237
Rozdział 11. Testowanie na potrzeby internetu – WSGI kontra HTTP	239
Wymagania techniczne	240
Testowanie HTTP	240
Testowanie klientów HTTP	244
Testowanie WSGI za pomocą biblioteki webtest	248
Stosowanie biblioteki webtest z frameworkami przeznaczonymi do tworzenia aplikacji internetowych	256
Tworzenie testów Django za pomocą klienta testów oferowanego przez Django	264
Testowanie projektu Django za pomocą frameworka pytest	267
Testowanie projektu Django za pomocą klienta testów Django	269
Podsumowanie	272
Rozdział 12. Testy E2E wykonywane za pomocą Framework Robot	273
Wymagania techniczne	274
Wprowadzenie do frameworka Robot	274
Testowanie za pomocą przeglądarek WWW	277
Rejestracja procesu wykonywania testów	281
Testowanie z użyciem przeglądarki WWW działającej w trybie headless	284
Testowanie wielu przeglądarek WWW	286
Rozbudowa frameworka Robot	289
Dodawanie niestandardowych słów kluczowych	290
Rozbudowa frameworka Robot w Pythonie	291
Podsumowanie	293