

Spis treści |

Wprowadzenie	15
---------------------------	-----------

Część I. Rozpoczęcie pracy i podstawy TDD

ROZDZIAŁ 1

Przygotowanie pierwszej implementacji TDD	23
Wymagania techniczne	23
Wybór zintegrowanego środowiska programistycznego	24
Microsoft Visual Studio	24
JetBrains Rider	26
Visual Studio Code	27
Wersje .NET i C#	27
Utworzenie szkieletu rozwiązania razem z testami jednostkowymi	28
Wymagania	28
Utworzenie szkieletu projektu	29
Zapoznanie się z wbudowanymi narzędziami przeznaczonymi do przeprowadzania testów	35
Implementacja wymagań z zastosowaniem programowania sterowanego testami	37
SUT	38
Klasa testów	39
Warunki i oczekiwania	40
Czerwony — zielony	41
Wzorzec AAA	42
Jeszcze więcej testów	43
Podsumowanie	48
Dalsza lektura	48

ROZDZIAŁ 2

Wprowadzenie do mechanizmu wstrzykiwania zależności	49
Wymagania techniczne	50
Aplikacja WFA	50
Utworzenie przykładowej aplikacji	50
Dodawanie komponentu odpowiedzialnego za dostarczanie rzeczywistej prognozy pogody	53
Poznanie mechanizmu wstrzykiwania zależności	58
Typy abstrakcyjne i konkretne	58
Czym jest zależność?	59
Znaczenie zależności	60
Definiuj zależności od abstrakcji, a nie od konkretnej implementacji	63
Wprowadzenie do mechanizmu wstrzykiwania zależności	67
Pierwszy przykład wstrzykiwania zależności	67
Testowanie API	68
Czym jest szew?	71
Odwroć kontrolę	72
Używanie kontenerów wstrzykiwania zależności	73
Rola kontenera	74
Kontenery podmiotów zewnętrznych	75
Cykl życiowy usługi	75
Refaktoryzacja pod kątem wstrzykiwania zależności	77
Rzeczywisty scenariusz użycia wstrzykiwania zależności	84
Wstrzykiwanie metody	88
Wstrzykiwanie właściwości	89
Lokalizator usługi	89
Podsumowanie	90
Dalsza lektura	90

ROZDZIAŁ 3

Rozpoczęcie pracy z testami jednostkowymi	91
Wymagania techniczne	92
Wprowadzenie do testów jednostkowych	92
Czym jest testowanie jednostkowe?	92
Frameworki testów jednostkowych	93
Wyjaśnienie struktury projektu stosującego testy jednostkowe	94
Dodawanie projektu xUnit za pomocą wiersza poleceń	94
Konwencje nazw w projekcie testów jednostkowych	95
Wykonanie przykładowego testu jednostkowego	96
Okno Test Explorer	97

Analiza anatomii klasy testu jednostkowego	97
Konwencja nadawania nazwy klasie	98
Metody testowe	98
Wzorzec „przygotowanie, działanie, asercja”	101
Testowany system	103
Omówienie podstaw frameworka xUnit	104
Atrybuty Fact i Theory	104
Wykonywanie testów	105
Klasa Assert	106
Klasa Record	107
Omówienie powiązań zachodzących między regułami SOLID a testami jednostkowymi	107
Reguła jednej odpowiedzialności	108
Reguła otwarte-zamknięte	109
Zasada podstawień Barbary Liskov	111
Zasada rozdzielania interfejsów	112
Zasada odwrócenia zależności	113
Podsumowanie	114
Dalsza lektura	114

ROZDZIAŁ 4

Rzeczywiste stosowanie testów jednostkowych

z wykorzystaniem dublerów używanych podczas testów

Wymagania techniczne	115
Wprowadzenie do koncepcji dublerów używanych podczas testów	116
Typy dublerów używanych podczas testów	116
Którego rozwiązania należy używać w programowaniu sterowanym testami?	129
Omówienie kolejnych kategorii testów	130
Testy integracyjne	130
Testy sinegration	135
Testy akceptacyjne	137
Wybór kategorii testów	138
Podsumowanie	139
Dalsza lektura	139

ROZDZIAŁ 5

Programowanie sterowane testami

Wymagania techniczne	140
Filary programowania sterowanego testami	141
Najpierw testy	141
Czerwony, zielony, refaktoryzacja	141

Programowanie sterowane testami w praktyce	143
Utworzenie rozwiązania w wierszu poleceń	144
Dodawanie zadania programistycznego	144
Krótkie podsumowanie	155
Najczęściej zadawane pytania i zastrzeżenia do programowania sterowanego testami	156
Dlaczego potrzebne jest programowanie sterowane testami?	
Czy nie można po prostu używać testów jednostkowych?	156
Podejście w stylu TDD podczas tworzenia oprogramowania wydaje się nienaturalne	156
Stosowanie programowania sterowanego testami będzie nas spowalniać	157
Czy programowanie sterowane testami ma znaczenie dla startupów?	158
Nie lubię programowania sterowanego testami i wolę najpierw zająć się swoją bazą kodu	159
Testy jednostkowe nie sprawdzają rzeczywistych aspektów kodu	159
Podobno istnieją dwie szkoły w zakresie programowania sterowanego testami — londyńska i klasyczna. Jakie są między nimi różnice?	160
Dlaczego niektórzy programiści nie lubią testów jednostkowych i programowania sterowanego testami?	160
Jaki związek zachodzi między programowaniem sterowanym testami a programowaniem ekstremalnym?	161
Czy system jest w stanie przetrwać bez programowania sterowanego testami?	161
Programowanie sterowane testami i testy syntezy	162
Testy syntezy jako alternatywa dla testów jednostkowych w programowaniu sterowanym testami	163
Wyzwania pojawiające się podczas stosowania testów syntezy	165
Podsumowanie	165
Dalsza lektura	166

ROZDZIAŁ 6

Wskazówki FIRSTHAND dotyczące programowania

sterowanego testami	167
Wymagania techniczne	168
Wskazówka „pierwszy”	168
Później oznacza nigdy	168
Przygotowanie do użycia mechanizmu wstrzykiwania zależności	168
Opracowanie z perspektywy klienta	169

Promowanie testowania sposobu działania	169
Eliminowanie fałszywych alarmów	170
Eliminowanie kodu spekulatywnego	170
Wskazówka „intencja”	170
NazwaMetody_Warunek_Oczekiwanie	171
NazwaMetody_Should_When	171
Struktura testu jednostkowego	172
Wskazówka „czytelność”	173
Inicjalizacja konstruktora testowanego systemu	173
Wzorzec budowniczego	175
Wskazówka „jeden sposób działania”	177
Czym jest sposób działania?	177
Przykład sposobu działania	178
Testowanie jedynie zewnętrznie zdefiniowanego sposobu działania	179
Dlaczego nie testujemy elementów wewnętrznych?	180
Test sprawdza tylko jeden sposób działania	181
Wskazówka „dokładność”	181
Testy jednostkowe przeznaczone do testowania zależności	181
Co oznacza pokrycie kodu testami?	182
Bycie dokładnym	186
Wskazówka „wysoka wydajność”	187
Integracja jako ukryte jednostki	187
Testy jednostkowe w ogromnym stopniu wykorzystujące procesor i pamięć operacyjną	188
Istnienie zbyt wielu testów	188
Wskazówka „automatyzacja”	189
Automatyzacja począwszy od dnia pierwszego	189
Niezależność od platformy	189
Wysoka wydajność w potoku ciągłej integracji	189
Wskazówka „brak współzależności”	190
Odpowiedzialność frameworka testów jednostkowych	190
Odpowiedzialność programisty	191
Wskazówka „deterministyczność”	192
Przypadki niedeterministycznych testów jednostkowych	192
Przykład zamrożenia czasu	193
Podsumowanie	194

Część II. Stworzenie aplikacji z zastosowaniem podejścia w stylu TDD

ROZDZIAŁ 7

Pragmatyczne omówienie architektury DDD	197
Wymagania techniczne	198
Praca z przykładową aplikacją	198
Projekt aplikacji	199
Projekt obiektów kontraktu	199
Projekt warstwy dziedziny	200
Poznanwanie dziedzin	201
Obiekty dziedziny	201
Encje i obiekty wartości	202
Agregacja	204
Modele anemiczne	204
Wszzechobecny język	205
Poznanwanie usług	206
Zarządzanie postem	206
Usługi aplikacji	209
Usługi infrastruktury	209
Cechy charakterystyczne usługi	209
Poznanwanie repozytoriów	211
Przykład repozytorium	211
Entity Framework i repozytoria	214
Połączenie wszystkiego w całość	214
Okno Solution Explorer	214
Widok architekuralny	215
Podsumowanie	216
Dalsza lektura	217

ROZDZIAŁ 8

Opracowanie aplikacji pozwalającej na rezerwowanie wizyt	218
Wymagania techniczne	219
Zebranie wymagań biznesowych	219
Cele biznesowe	219
Historyjki użytkownika	220
Projektowanie w duchu architektury DDD	227
Obiekty dziedziny	227
Usługi dziedziny	228
Architektura systemu	228

Implementowanie tras	229
Frontend	230
Backend w postaci relacyjnej bazy danych	231
Backend w postaci bazy danych opartej na dokumentach	231
Używanie wzorca mediatora	231
Podsumowanie	232
Dalsza lektura	232

ROZDZIAŁ 9

Wykorzystanie Entity Framework i relacyjnej bazy danych do opracowania aplikacji pozwalającej na rezerwowanie wizyt 233

Wymagania techniczne	234
Planowanie kodu źródłowego i struktury projektu	235
Analiza struktury projektu	235
Utworzenie projektów i konfiguracja zależności	236
Konfiguracja projektu dziedziny	238
Przygotowanie Entity Framework	239
Przygotowanie projektu witryny internetowej	241
Implementacja Web API z użyciem programowania sterowanego testami	241
Używanie działającego dostawcy EF dla magazynu danych w pamięci	242
Implementacja pierwszej historyjki użytkownika	245
Implementacja piątej historyjki użytkownika (zarządzanie czasem)	249
Udzielenie odpowiedzi na najczęściej zadawane pytania	252
Czy te testy jednostkowe są wystarczające?	253
Dlaczego nie utworzyliśmy testów jednostkowych dla kontrolerów?	253
Czy system został wystarczająco przetestowany?	253
Pominęliśmy testowanie pewnych obszarów, więc jak osiągnąć wysoki poziom pokrycia testami?	253
Podsumowanie	254

ROZDZIAŁ 10

Wykorzystanie wzorca repozytorium i bazy danych opartej na dokumentach do opracowania aplikacji pozwalającej na rezerwowanie wizyt 255

Wymagania techniczne	256
Planowanie kodu źródłowego i struktury projektu	257
Analiza struktury projektu	257
Utworzenie projektów i konfiguracja zależności	259
Konfiguracja projektu dziedziny	261

Wzorzec repozytorium	262
Przygotowanie projektu witryny internetowej	266
Implementacja Web API z użyciem programowania sterowanego testami	266
Implementacja pierwszej historyjki użytkownika	267
Implementacja piątej historyjki użytkownika (zarządzanie czasem)	271
Udzielenie odpowiedzi na najczęściej zadawane pytania	274
Czy te testy jednostkowe są wystarczające?	274
Dlaczego nie utworzyliśmy testów jednostkowych dla kontrolerów?	274
Dlaczego nie utworzyliśmy testów jednostkowych dla implementacji repozytoriów?	275
Czy system został wystarczająco przetestowany?	275
Pominęliśmy testowanie pewnych obszarów, więc jak osiągnąć wysoki poziom pokrycia testami?	276
Podsumowanie	276

Część III. Zastosowanie programowania sterowanego testami we własnych projektach

ROZDZIAŁ 11

Wdrożenie potoku ciągłej integracji

za pomocą usługi GitHub Actions	281
Wymagania techniczne	281
Wprowadzenie do systemu ciągłej integracji	282
Sposób działania systemu ciągłej integracji	283
Zalety systemu ciągłej integracji	284
Implementacja procesu ciągłej integracji za pomocą usługi GitHub Actions	285
Utworzenie przykładowego projektu w repozytorium GitHub	286
Zdefiniowanie sposobu działania	286
System ciągłej integracji i testowanie	292
Symulowanie testów zakończonych niepowodzeniem	292
Omówienie sposobu działania	294
Podsumowanie	294
Dalsza lektura	295

ROZDZIAŁ 12

Praca z uaktualnianymi projektami	296
Wymagania techniczne	296
Analizowanie trudności	297
Obsługa mechanizmu wstrzykiwania zależności	297
Trudności związane z modyfikowaniem kodu źródłowego	299
Trudności związane z czasem i wysiłkiem	300
Strategia pozwalająca na zastosowanie programowania sterowanego testami	300
Rozważ ponowne utworzenie projektu	300
Zmiany kodu źródłowego	301
Natywne obsługa dla mechanizmu wstrzykiwania zależności	303
Poziom pokrycia testami przed dodaniem testów jednostkowych	303
Refaktoryzacja na potrzeby testów jednostkowych	304
Tworzenie egzemplarzy w kodzie	304
Zastępowanie statycznych elementów składowych	307
Zmiana struktury kodu źródłowego	308
Podsumowanie	312
Dalsza lektura	313

ROZDZIAŁ 13**Zawiłości związane ze stosowaniem programowania**

sterowanego testami	314
Trudności techniczne	314
Projekt nowy czy uaktualniany?	315
Narzędzia i infrastruktura	315
Trudności w zespole	316
Doświadczenie zespołu	317
Ochota do działania	318
Czas	319
Trudności biznesowe	319
Korzyści biznesowe wynikające z programowania sterowanego testami	319
Wady testów jednostkowych z perspektywy biznesowej	321
Argumenty za programowaniem sterowanym testami i związane z nim błędne koncepcje	322
Testy jednostkowe, a nie programowanie sterowane testami	322
Testy jednostkowe nie są implementowane przez testerów	323
Sposób tworzenia i obsługi technicznej dokumentacji	323
Mamy niekompetentnych programistów	324
Podsumowanie	324

DODATEK A**Biblioteki, których najczęściej używa się**

podczas testów jednostkowych	325
Wymagania techniczne	325
Frameworki testów jednostkowych	326
MSTest	326
NUnit	328
Biblioteki imitacji	331
Moq	331
Biblioteki pomocnicze dla testów jednostkowych	333
Fluent Assertions	333
AutoFixture	334
Dalsza lektura	336

DODATEK B**Zaawansowane scenariusze związane z używaniem imitacji** 337

Wymagania techniczne	337
Utworzenie biblioteki klienta OpenWeather	337
API One Call	338
Utworzenie szkieletu rozwiązania	338
Rozpoczęcie pracy nad implementacją z użyciem programowania sterowanego testami	339
Niezaliczenie i późniejsze zaliczenie testu	342
Podsumowanie	343
Analizowanie skomplikowanych scenariuszy imitacji	343
Dalsza lektura	344