

Spis treści

Wstęp	11
Rozdział 1. Podstawy TypeScriptu	17
Wprowadzenie	17
Ewolucja TypeScriptu	18
Cele projektowe TypeScriptu	20
Pierwsze kroki z TypeScriptem	20
Kompilator TypeScriptu	21
Konfigurowanie projektu TypeScriptu	22
Ćwiczenie 1.01. Użycie pliku tsconfig.json i pierwsze kroki z TypeScriptem	23
Typy i ich zastosowanie	24
TypeScript i funkcje	26
Ćwiczenie 1.02. Praca z funkcjami w TypeScriptie	31
TypeScript i obiekty	33
Ćwiczenie 1.03. Praca z obiektami	36
Typy proste	37
Ćwiczenie 1.04. Operator typeof	39
Łańcuchy znaków	40
Liczby	41
Typy logiczne (boolowskie)	41
Tablice	42
Krotki	44
Ćwiczenie 1.05. Stosowanie tablic i krotek do wydajnego sortowania obiektów	46
Typy wyliczeniowe	49
Typy any i unknown	51
Wartości null i undefined	52
Typ never	53
Typy funkcji	54
Tworzenie własnych typów	54
Ćwiczenie 1.06. Tworzenie funkcji kalkulatora	55
Zadanie 1.01. Tworzenie biblioteki do pracy z łańcuchami znaków	58
Podsumowanie	60

Rozdział 2. Pliki deklaracji	61
Wprowadzenie	61
Pliki deklaracji	62
Ćwiczenie 2.01. Tworzenie pliku deklaracji od podstaw	63
Wyjątki	68
Zewnętrzne biblioteki kodów	70
Biblioteka DefinitelyTyped	70
Analiza zewnętrznego pliku deklaracji	70
Ćwiczenie 2.02. Tworzenie typów z wykorzystaniem zewnętrznych bibliotek	72
Programistyczny przepływ pracy dla DefinitelyTyped	75
Ćwiczenie 2.03. Tworzenie aplikacji baseballowej karty składu	75
Zadanie 2.01. Tworzenie pliku deklaracji mapy ciepłej	77
Podsumowanie	78
Rozdział 3. Funkcje	79
Wprowadzenie	79
Funkcje w TypeScriptie	80
Ćwiczenie 3.01. Pierwsze kroki z funkcjami w TypeScriptie	80
Słowo kluczowe function	82
Parametry funkcji	83
Argument a parametr	84
Parametry opcjonalne	84
Parametry domyślne	85
Wiele argumentów	86
Parametry reszty	86
Destrukturyzacja typów zwracanych	87
Konstruktor funkcji	88
Ćwiczenie 3.02. Porównywanie tablic liczb	88
Wyrażenia funkcyjne	90
Funkcje strzałkowe	92
Inferencja typów	93
Ćwiczenie 3.03. Pisanie funkcji strzałkowych	94
Słowo kluczowe this	97
Ćwiczenie 3.04. Stosowanie this w obiekcie	99
Domknięcia i zakres	101
Ćwiczenie 3.05. Tworzenie metody wytwórczej zamówień	106
z wykorzystaniem domknięć	106
Rozwijanie funkcji	108
Ćwiczenie 3.06. Refaktoryzacja do postaci funkcji rozwijanych	109
Programowanie funkcyjne	112
Organizowanie funkcji w obiekty i klasy	114
Ćwiczenie 3.07. Refaktoryzacja kodu JavaScript do TypeScriptu	115
Słowa kluczowe import, export i require	117
Ćwiczenie 3.08. Importowanie i eksportowanie	119
Zadanie 3.01. Budowanie systemu rezerwacji lotów z wykorzystaniem funkcji	121
Wykonywanie testów jednostkowych za pomocą ts-jest	123
Zadanie 3.02. Pisanie testów jednostkowych	127
Obsługa błędów	127
Podsumowanie	128

Rozdział 4. Klasy i obiekty	130
Wprowadzenie	130
Czym są klasy i obiekty?	131
Ćwiczenie 4.01. Budowanie pierwszej klasy	132
Rozszerzanie zachowania klasy za pomocą konstruktora	134
Słowo kluczowe this	134
Ćwiczenie 4.02. Definiowanie atrybutów klasy i uzyskiwanie do nich dostępu	135
Ćwiczenie 4.03. Integrowanie typów z klasami	137
Interfejsy TypeScriptu	138
Ćwiczenie 4.04. Budowanie interfejsu	139
Generowanie kodu HTML w metodach	141
Ćwiczenie 4.05. Generowanie i przeglądanie kodu HTML	142
Praca z wieloma klasami i obiektami	143
Ćwiczenie 4.06. Łączenie klas	143
Zadanie 4.01. Tworzenie modelu użytkownika za pomocą klas, obiektów i interfejsów	147
Podsumowanie	148
Rozdział 5. Interfejsy i dziedziczenie	149
Wprowadzenie	149
Interfejsy	150
Studium przypadku: pisanie pierwszego interfejsu	151
Ćwiczenie 5.01. Implementacja interfejsów	155
Ćwiczenie 5.02. Implementacja interfejsów — tworzenie prototypowej aplikacji blogowej	158
Ćwiczenie 5.03. Tworzenie interfejsów dla funkcji aktualizacji bazy danych użytkownika	160
Zadanie 5.01. Budowanie komponentu zarządzania użytkownikami za pomocą interfejsów	161
Dziedziczenie w TypeScriptie	163
Ćwiczenie 5.04. Tworzenie klasy bazowej i dwóch rozszerzonych klas potomnych	170
Ćwiczenie 5.05. Tworzenie klas bazowych i rozszerzonych przy użyciu dziedziczenia wielopoziomowego	174
Zadanie 5.02. Tworzenie prototypowej aplikacji internetowej dla salonu samochodowego przy użyciu dziedziczenia	177
Podsumowanie	178
Rozdział 6. Typy zaawansowane	179
Wprowadzenie	179
Aliasy typów	180
Ćwiczenie 6.01. Implementowanie aliasów typów	184
Literały typów	185
Ćwiczenie 6.02. Literały typów	187
Typy części wspólnej	188
Ćwiczenie 6.03. Tworzenie typów części wspólnej	191

Typy unii	194
Ćwiczenie 6.04. Aktualizowanie zasobów produktów za pomocą interfejsu API	195
Typy indeksowe	198
Ćwiczenie 6.05. Wyświetlanie komunikatów o błędach	200
Zadanie 6.01. Typ części wspólnej	201
Zadanie 6.02. Typ unii	202
Zadanie 6.03. Typ indeksowy	202
Podsumowanie	203
Rozdział 7. Dekoratory	204
Wprowadzenie	204
Refleksja	205
Konfigurowanie opcji kompilatora	205
Znaczenie dekoratorów	206
Problem zagadnień przekrojowych	207
Dekoratory i metody wytwórcze dekoratorów	210
Składnia dekoratorów	211
Metody wytwórcze dekoratorów	212
Dekoratory klas	213
Wstrzykiwanie właściwości	213
Ćwiczenie 7.01. Tworzenie prostej metody wytwórczej dekoratora klas	214
Rozszerzenie konstruktora	216
Ćwiczenie 7.02. Użycie dekoratora rozszerzeń konstruktora	216
Opakowywanie konstruktora	218
Ćwiczenie 7.03. Tworzenie dekoratora rejestrowania dla klasy	219
Dekoratory metod i akcesorów	221
Dekoratory funkcji instancji	222
Ćwiczenie 7.04. Tworzenie dekoratora, który oznacza funkcję jako wyliczalną	224
Dekoratory funkcji statycznych	226
Dekoratory opakowywania metod	226
Ćwiczenie 7.05. Tworzenie dekoratora rejestrowania dla metody	227
Zadanie 7.01. Tworzenie dekoratorów do liczenia wywołań	229
Stosowanie metadanych w dekoratorach	231
Obiekt Reflect	232
Ćwiczenie 7.06. Dodawanie metadanych do metod za pomocą dekoratorów	233
Dekoratory właściwości	235
Ćwiczenie 7.07. Tworzenie i stosowanie dekoratora właściwości	237
Dekoratory parametrów	238
Ćwiczenie 7.08. Tworzenie i stosowanie dekoratora parametrów	240
Zastosowanie wielu dekoratorów do pojedynczego elementu docelowego	242
Zadanie 7.02. Stosowanie dekoratorów w celu dodania zagadnień przekrojowych	243
Podsumowanie	245

Rozdział 8. Wstrzykiwanie zależności w TypeScriptie	246
Wprowadzenie	246
Wzorzec projektowy DI	247
DI we frameworku Angular	253
Ćwiczenie 8.01. Dodawanie przechwytywacza HttpInterceptor do aplikacji Angulara	257
DI we frameworku Nest.js	260
Biblioteka InversifyJS	261
Ćwiczenie 8.02. Tworzenie aplikacji „Witaj, świecie” z wykorzystaniem InversifyJS	262
Zadanie 8.01. Kalkulator oparty na DI	265
Podsumowanie	267
Rozdział 9. Typy sparametryzowane i warunkowe	268
Wprowadzenie	269
Typy sparametryzowane	270
Interfejsy sparametryzowane	272
Typy sparametryzowane	274
Klasy sparametryzowane	274
Ćwiczenie 9.01. Klasa sparametryzowana Set	275
Funkcje sparametryzowane	278
Ograniczenia typów sparametryzowanych	281
Ćwiczenie 9.02. Funkcja sparametryzowana memoize	282
Wartości domyślne typów sparametryzowanych	285
Typy warunkowe	286
Zadanie 9.01. Tworzenie typu DeepPartial<T>	288
Podsumowanie	289
Rozdział 10. Pętle zdarzeń i zachowania asynchroniczne	290
Wprowadzenie	290
Podejście wielowątkowe	291
Wykonywanie asynchroniczne	292
Wykonywanie kodu JavaScript	294
Ćwiczenie 10.01. Układanie funkcji w stos	296
Przeglądarki i JavaScript	297
Zdarzenia w przeglądarce	299
Interfejsy API środowiska	299
Metoda setTimeout	300
Ćwiczenie 10.02. Korzystanie z setTimeout	300
AJAX (asynchroniczny JavaScript i XML)	302
Zadanie 10.01. Przeglądarka filmów wykorzystująca obiekt xhr i wywołania zwrotne	304
Obietnice	308
Ćwiczenie 10.03. Liczenie do 5	308
Czym są obietnice?	310
Ćwiczenie 10.04. Liczenie do 5 za pomocą obietnic	312

Zadanie 10.02. Przeglądarka filmów wykorzystująca API fetch i obietnice	314
Funkcjonalność async/await	316
Ćwiczenie 10.05. Liczenie do 5 z wykorzystaniem async i await	318
Zadanie 10.03. Przeglądarka filmów wykorzystująca API fetch i funkcjonalność async/await	319
Podsumowanie	321
Rozdział 11. Funkcje wyższego rzędu i wywołania zwrotne	323
Wprowadzenie	323
Wprowadzenie do funkcji HOC — przykłady	324
Funkcje wyższego rzędu	325
Ćwiczenie 11.01. Stosowanie funkcji wyższego rzędu do organizowania filtrowania danych i operowania na nich	329
Wywołania zwrotne	331
Pętla zdarzeń	332
Wywołania zwrotne w Node.js	335
Piekło wywołań zwrotnych	335
Unikanie piekła wywołań zwrotnych	337
Wyodrębnienie procedur obsługi wywołań zwrotnych do deklaracji funkcji na poziomie pliku	338
Łączenie wywołań zwrotnych w łańcuch	339
Obietnice	341
Funkcjonalność async/await	342
Zadanie 11.01. Funkcja wyższego rzędu pipe	343
Podsumowanie	344
Rozdział 12. Przewodnik po obietnicach w TypeScriptie	346
Wprowadzenie	346
Ewolucja i powody wprowadzenia obietnic	347
Anatomia obietnic	350
Wywołanie zwrotne obietnicy	350
Metody then i catch	350
Stan pending	351
Stan fulfilled	351
Stan rejected	352
Łączenie w łańcuch	352
Ćwiczenie 12.01. Łączenie w łańcuch obietnic	353
Metoda finally	355
Metoda Promise.all	356
Ćwiczenie 12.02. Rekurencyjna metoda Promise.all	357
Metoda Promise.allSettled	360
Ćwiczenie 12.03. Promise.allSettled	362
Metoda Promise.any	364
Metoda Promise.race	364
Ulepszanie obietnic za pomocą typów	365
Ćwiczenie 12.04. Renderowanie asynchroniczne	366
Biblioteki i natywne obietnice — zewnętrzne biblioteki, Q i Bluebird	368
Wypełnienia obietnic	369

Promisyfikacja	371
Funkcjonalność util.promisify frameworku Node.js	373
Asynchroniczny system plików	374
Metoda fs.readFile	374
Metoda fs.readFileSync	374
API obietnic biblioteki fs	375
Ćwiczenie 12.05. API obietnic biblioteki fs	375
Praca z bazami danych	377
Programowanie z wykorzystaniem API RESTful	380
Ćwiczenie 12.06. Implementowanie API RESTful wspieranego przez sqLite	380
Połączenie wszystkich elementów w celu zbudowania aplikacji korzystającej z obietnic	391
Zadanie 12.01. Budowanie aplikacji korzystającej z obietnic	392
Podsumowanie	393
Rozdział 13. Funkcjonalność async/await w TypeScriptie	394
Wprowadzenie	394
Ewolucja i powody wprowadzenia async/await	396
Funkcjonalność async/await w TypeScriptie	397
Ćwiczenie 13.01. Środowiska docelowe transpilacji	397
Wybór środowiska docelowego	400
Składnia	401
Słowo kluczowe async	401
Ćwiczenie 13.02. Słowo kluczowe async	401
Ćwiczenie 13.03. Rozwiązywanie funkcji async za pomocą then	402
Słowo kluczowe await	403
Ćwiczenie 13.04. Słowo kluczowe await	403
Ćwiczenie 13.05. Oczekiwanie na obietnicę	404
Lukier składniowy	405
Obsługa wyjątków	405
Ćwiczenie 13.06. Obsługa wyjątków	406
Funkcjonalność await najwyższego poziomu	409
Metody obiektu Promise	411
Ćwiczenie 13.07. async/await we frameworku Express.js	411
Ćwiczenie 13.08. NestJS	414
Ćwiczenie 13.09. TypeORM	418
Zadanie 13.01. Użycie await do refaktoryzacji łańcucha obietnic	422
Podsumowanie	423
Rozdział 14. TypeScript i biblioteka React	425
Wprowadzenie	425
Typowanie Reacta	426
TypeScript w Reakcie	426
Witaj, Reakcie	427
Komponent	428
Komponenty stanowe	428
Komponenty bezstanowe	429
Komponenty czyste	429
Komponenty wyższego rzędu	429

