

Programowanie funkcyjne w F#

Cele szkolenia

Celem szkolenie jest przedstawienie głównych cech programowania funkcyjnego oraz nauka języka F#. Język F# pozwala tworzyć kod stabilny, wydajny oraz czytelny. Stworzone aplikacje mogą być uruchomione na platformie .NET, a także kompilowane do innych technologii takich jak np. JavaScript, czy Python. Programowanie funkcyjne zmienia sposób myślenia, wiedzę przedstawioną na kursie można wykorzystać niezależnie od stosowanej technologii, czy języka programowania.

Umiejętności

Po szkoleniu uczestnik prawidłowo:

- wskazuje główne cechy programowania funkcyjnego
- rozpoznaje język programowania F#
- tworzy nowoczesne aplikacje działające na platformie .NET i .NET Core z wykorzystaniem języka F#
- identyfikuje miejsca gdzie obiektowe podejście nie koniecznie jest najlepszym rozwiązaniem danego problemu oraz jaką alternatywę warto zastosować
- wyjaśnia kierunki i cele ostatnich zmian popularnych języków programowania takich jak Java, C#, czy JavaScript

Profil uczestników

Szkolenie przeznaczone jest dla programistów dowolnego języka programowania chcących poznać język F# oraz paradygmat programowania funkcyjnego.

Przygotowanie uczestników

Znajomość dowolnego języka programowania. Znajomość platformy .NET jest pomocna, ale nie jest niezbędna.

Szczegółowy program szkolenia

Wprowadzenie do języka F#

- typy proste
- definicja wartości vs definicja zmiennych
- definicja funkcji
- wnioskowanie typów
- formatowanie kodu ma znaczenie
- wszystko jest wyrażeniem

- dopasowanie do wzorca
- komplikacja vs interpretacja kodu F# (fsi)

Funkcje

- definicja funkcji
- rekurencja
- funkcja jako wartość, funkcja anonimowa (lambda), domknięcie
- currying oraz częściowe zaaplikowanie argumentów
- funkcje wyższego rzędu
- operator "pipe", kompozycja funkcji

Typy danych

- aliasy do typów
- krotki, rekordy, sumy rozłączne
- typ unit
- algebraiczne typy danych
- typy generyczne
- rzutowanie typów
- type driven development, domain modeling

Kolekcje

- kolekcje vs leniwe sekwencje
- niezmiennicze vs zmiennicze kolekcje
- operacje na kolekcjach: map, filter, fold, zip, ...
- slices

Standardowa biblioteka F#

- wbudowane funkcje: printfn, ignore, id, string, min, max, ...
- wbudowane operatory: |>, ||>, >>, <<, ...
- wbudowane typy: Option, List, Result, Choice
- moduły dla kolekcji .NET: Array, ResizeArray, sekwencje

- moduły dla kolekcji F#: List, Map

Computation expressions

- computation expressions
- query expressions
- programowanie asynchroniczne (async, task, promise)
- tworzenie własnych "computation expressions"

Pozostałe zagadnienia

- moduły
- przestrzenie nazwa
- obsługa wyjątków
- komunikacja z kodem .net (delegaty, wartości wyliczeniowe, null)
- atrybuty

Programowanie imperatywne

- instrukcja warunkowa if..then..else
- pętle while..do, for..in, for..to/downto
- modyfikowanie stanu, ref

Programowanie obiektowe

- interfejsy, klasy, struktury
- object expression

Cechy programowania funkcyjnego

- funkcja jako wartość
- wszystko jest wyrażeniem
- niezmienniczość, efekty uboczne, czyste funkcje
- leniwa ewaluacja, sekwencje
- imperatywne vs deklaratywne programowanie
- obiektowa vs funkcyjna organizacja kodu

- algebraiczne typy danych
- opcjonalność, obsługa błędów
- funktory, monady, aplikatywne funktory

Aplikacje webowe w F#

- REST-owe usługi, MVC
- Aplikacje SPA (Fable)
- Stos SAFE

Metoda realizacji szkolenia

Szkolenie realizowane jest w formie naprzemiennie następującej po sobie części teoretycznej w postaci mini wykładów oraz części praktycznej w postaci ćwiczeń.

Liczba dni, liczba godzin szkoleniowych

3 dni, 8 godzin szkoleniowych

Ścieżka rozwoju po szkoleniu

Tworzenie aplikacji Internetowych z wykorzystaniem ASP.NET Core
Nowoczesny C#
Nowoczesny JavaScript