

Programowanie funkcyjne

Wymagania wstępne: brak

Formuła nauczania: wykład 30 godzin, ćwiczenia 30 godzin

Metoda oceny/forma zaliczenia przedmiotu: egzamin

Język wykładowy: polski

Prowadzący: Włodzimierz Moczurad

Treści kształcenia:

Kurs przedstawia programowanie funkcyjne jako paradygmat programowania, ukazując zarówno jego korzenie matematyczne, jak i powszechnie przyjęte techniki programistyczne. Kurs obejmuje wprowadzenie do języka Haskell, należącego do głównego nurtu badawczego i aplikacyjnego współczesnego programowania funkcyjnego, oraz krótką wzmiankę o języku Erlang, ważnym ze względu na rozbudowaną obsługę współbieżności.

Programowanie funkcyjne jako paradygmat programowania: funkcje „matematyczne” a funkcje-podprogramy, funkcje jako obiekty „pierwszej kategorii”, rachunek lambda, przegląd języków funkcyjnych (Lisp, Scheme, ML, Haskell, Erlang).

Podstawy pracy z interpreterem Haskell: sesje, skrypty, wyrażenia, wyliczanie wyrażen, standardowe biblioteki, semantyka operacyjna języka.

Typy: typy wbudowane, silne typowanie, definiowanie typów, typy rekurencyjne, typy polimorficzne, klasy typów, wnioskowanie o typie.

Definiowanie funkcji: rekursja, składanie funkcji, dopasowywanie do wzorca, funkcje wyższego rzędu, postać rozwinięta, polimorfizm, operatory, wyliczanie leniwe, definicje „bezpunktowe”.

Listy: konstrukcja, filtrowanie, aplikacje listowe z użyciem map, definicje z użyciem foldl/foldr, listy nieskończone.

Monady: wejście-wyjście z użyciem monad, aksjomaty, operatory, definiowanie monad, obliczenia posiadające stan, listy i inne struktury jako monady, aplikacje z użyciem fmap.

Efektywność obliczeń: wyliczanie leniwe i strategie redukcji a efektywność, obliczenia na strukturach nieskończonych, łączenie w pary/n-tki, rekursja ogonowa.

Programowanie współbieżne w języku Erlang: procesy, komunikacja międzyprocesowa, programowanie w środowisku rozproszonym.

Zalecana literatura:

1. R. Bird, *Introduction to Functional Programming using Haskell*, Prentice Hall, 1988.
2. G. Hutton, *Programming in Haskell*, Cambridge University Press, 2007.
3. J. Reynolds, *Theories of Programming Languages*, Cambridge University Press, 1998.
4. M. Moczurad, *Wybrane zagadnienia z teorii rekursji*, Wydawnictwo UJ, 2002.
5. R. Sebesta, *Concepts of Programming Languages*, Addison-Wesley, 2008.
6. J. Armstrong, *Programming Erlang: Software for a Concurrent World*, Pragmatic Bookshelf, 2007.