



Filozofia Clojure

Mateusz Dereniowski

Instytut Informatyki Uniwersytetu Wrocławskiego

15 lipca 2010



Wprowadzenie

Clojure w skrócie

Wieloplatformowość aka JVM

Standardy

Czym jest JVM

Co łączy Clojure i JVM?

Cechy języka

Cechy Clojure

Struktury agregujące

Cechy języka funkcyjnego

Cechy Javy

Współbieżność

Odejście od programowania obiektowego

Programowanie czysto funkcyjne

Niemodyfikowalny stan

Stan a tożsamość

Programowa pamięć transakcyjna

W skrócie



Clojure

Nowoczesny dialekt języka Lisp:



W skrócie



Clojure

Nowoczesny dialekt języka Lisp:

- twórca: Rich Hickey



W skrócie



Clojure

Nowoczesny dialekt języka Lisp:

- twórca: Rich Hickey
- początek: 16 października 2007



W skrócie



Clojure

Nowoczesny dialekt języka Lisp:

- twórca: Rich Hickey
- początek: 16 października 2007
- paradygmat: funkcyjny, wieloparadygmatowy



W skrócie



Clojure

Nowoczesny dialekt języka Lisp:

- twórca: Rich Hickey
- początek: 16 października 2007
- paradygmat: funkcyjny, wieloparadygmatowy
- typowanie: dynamiczne, silne



W skrócie



Clojure

Nowoczesny dialekt języka Lisp:

- twórca: Rich Hickey
- początek: 16 października 2007
- paradygmat: funkcyjny, wieloparadygmatowy
- typowanie: dynamiczne, silne
- platforma: **wieloplatformowy**

Standardy



Clojure

Pierwszy przemysłowo wykorzystywanym dialekt Lispu?



Czym jest JVM

Java Virtual Machine:

środowisko uruchomieniowe dla programów, niezależne od platformy, stabilne, szybkie, uznane przez przemysł



Co łączy Clojure i JVM?

- Clojure został zaprojektowany od podstawy, aby działać w środowisku Javy,



Co łączy Clojure i JVM?

- Clojure został zaprojektowany od podstawy, aby działać w środowisku Javy,
- kompilowany bezpośrednio do kodu bajtowego JVM,



Co łączy Clojure i JVM?

- Clojure został zaprojektowany od podstawy, aby działać w środowisku Javy,
- kompilowany bezpośrednio do kodu bajtowego JVM,
- funkcje zdefiniowane w Clojure mogą być użyte zamiennie z kodem Javy i vice versa.



Cechy Clojure

- **REPL**: Read, Eval, Print, Loop,



Cechy Clojure

- **REPL**: Read, Eval, Print, Loop,
- bogaty zestaw typów podstawowych (Lisp),



Cechy Clojure

- **REPL**: Read, Eval, Print, Loop,
- bogaty zestaw typów podstawowych (Lisp),
- struktury agregujące:
 - listy,
 - wektory,
 - mapy,
 - zbiory,



Cechy Clojure

- **REPL**: Read, Eval, Print, Loop,
- bogaty zestaw typów podstawowych (Lisp),
- struktury agregujące:
 - listy,
 - wektory,
 - mapy,
 - zbiory,
- makra (Lisp)



Cechy Clojure

- **REPL**: Read, Eval, Print, Loop,
- bogaty zestaw typów podstawowych (Lisp),
- struktury agregujące:
 - listy,
 - wektory,
 - mapy,
 - zbiory,
- makra (Lisp)
- wsparcie dla wielowątkowości



Struktury agregujące

Wszystkie wbudowane w *Clojure* struktury implementują iteratory, comparatory, interfejsy kolekcji itp.

Cechy języka funkcyjnego

- funkcje obywatelami pierwszej klasy, domknięcia,

Cechy języka funkcyjnego

- funkcje obywatelami pierwszej klasy, domknięcia,
- trwałe struktury danych

Cechy języka funkcyjnego

- funkcje obywatelami pierwszej klasy, domknięcia,
- trwałe struktury danych
- rekursja zamiast pętli z efektami ubocznymi,

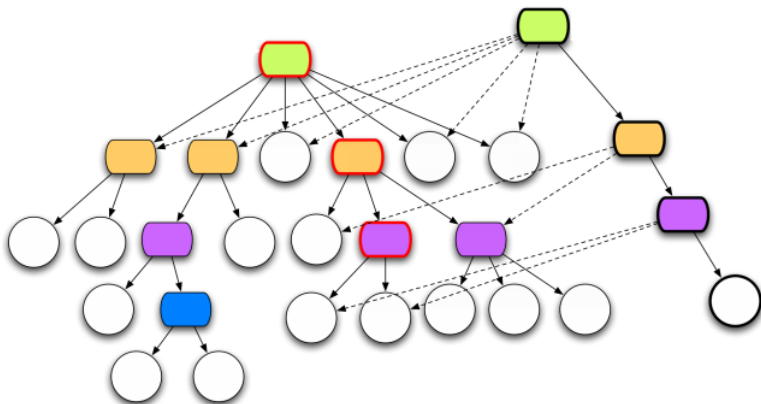
Cechy języka funkcyjnego

- funkcje obywatelami pierwszej klasy, domknięcia,
- trwałe struktury danych
- rekursja zamiast pętli z efektami ubocznymi,
- leniwe sekwencje

Cechy języka funkcyjnego

- funkcje obywatelami pierwszej klasy, domknięcia,
- trwałe struktury danych
- rekursja zamiast pętli z efektami ubocznymi,
- leniwe sekwencje
- list comprehensions

Trwałe struktury danych





Cechy Javy

- dostęp do bogatego zbioru bibliotek,



Cechy Javy

- dostęp do bogatego zbioru bibliotek,
- proxy do interfejsów/class zdefiniowanych w Javie,



Stan

Mutable objects are the new spaghetti code.
Rich Hickey



TDD

Musi być coś na rzeczy, skoro przemysł tak silnie promuje TDD.



Stan obiektów

Programiści języków obiektowych doskonale zdają sobie sprawę z niemożności kontrolowania stanu obiektów zawartych w swoich programach.

Zalety programowania czysto funkcyjnego

- czyste funkcje są łatwe do uwspółbieżniania,

Zalety programowania czysto funkcyjnego

- czyste funkcje są łatwe do uwspółbieżniania,
- czystość funkcji zapewnia ich lepsze zrozumienie...

Zalety programowania czysto funkcyjnego

- czyste funkcje są łatwe do uwspółbieżniania,
- czystość funkcji zapewnia ich lepsze zrozumienie...
- ...co w połączeniu z kompozycyjnym charakterem programów daje lepszą możliwość enkapsulacji poszczególnych fragmentów kodu, co zwiększa poziom abstrakcji i reusability,

Zalety programowania czysto funkcyjnego

- czyste funkcje są łatwe do uwspółbieżniania,
- czystość funkcji zapewnia ich lepsze zrozumienie...
- ...co w połączeniu z kompozycyjnym charakterem programów daje lepszą możliwość enkapsulacji poszczególnych fragmentów kodu, co zwiększa poziom abstrakcji i reusability,
- kompozycja funkcji daje także lepszą możliwość śledzenia działania programu,



Efekty uboczne

Niemniej *Clojure* nie idzie w ślady za *Haskellem* wymuszając bezwarunkową czystość funkcji. Dzięki przechowywaniu stanu programu w strukturach odpornych na współbieżną rywalizację, jest on zawsze bezpieczny i spójny.



Stan a tożsamość

Dzięki zastosowaniu trwałych struktur danych każda "zmiana" stanu danej struktury wiąże się z utworzeniem nowego obiektu.

Obiekty jako takie nie ulegają zmianie.

Programowa pamięć transakcyjna

- atomowość - wszystkie operacje zostają wykonane, albo transakcja nie dochodzi do skutku.

Programowa pamięć transakcyjna

- atomowość - wszystkie operacje zostają wykonane, albo transakcja nie dochodzi do skutku.
- spójność - transakcje są sprawdzane przed zaakceptowaniem,



Programowa pamięć transakcyjna

- atomowość - wszystkie operacje zostają wykonane, albo transakcja nie dochodzi do skutku.
- spójność - transakcje są sprawdzane przed zaakceptowaniem,
- odizolowanie - żadna transakcja nie widzi efektów działania innych transakcji podczas swojego wykonywania,