

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра многопроцессорных систем и сетей**

Аннотация к дипломной работе
«Расширение системы типов языка Clojure»

Иванюк Андрей Иванович

Научный руководитель – доцент Рафеенко Е.Д.

2015

Реферат

Дипломная работа, 42 страницы, 2 рисунка, 7 источников.

CLOJURE, СИСТЕМЫ ТИПОВ, СТАТИЧЕСКАЯ ТИПИЗАЦИЯ, ВЫВОД ТИПОВ, АЛГОРИТМ ХИНДЛИ-МИЛНЕРА, АНАЛИЗ КОДА

Объект исследования — статические системы типов, исходный код на языке программирования Clojure.

Цель работы — расширение системы типов языка Clojure путем разработки системы для проверки исходного кода на корректность с точки зрения соответствия типов.

В процессе работы для достижения поставленной цели были реализованы следующие задачи: изучены различные способы реализации статической системы типов, реализована система для проверки типов на корректность для проектов на языке Clojure, реализован анализатор исходного кода на языке Clojure, реализован вывод типов с использованием алгоритма Хиндли-Милнера, разработаны утилиты для взаимодействия программиста с системой проверки типов.

Работа имеет практическое значение в сфере разработки программного кода, так как позволяет разработчикам, использующим язык программирования Clojure, производить анализ исходного кода их проектов на предмет соответствия типов. При этом не возникает необходимость аннотировать все функции и переменные в проекте конкретными типами.

Разработанная система может быть свободно использована в любом проекте на языке программирования Clojure. Система не требует затрат времени на изменение проекта для ее использования и не замедляет в значительной степени разработку в дальнейшем.

Abstract

Diploma thesis, 41 pages, 2 figures, 7 sources.

CLOJURE, TYPE SYSTEMS, STATIC TYPING, TYPE INFERENCE,
HINDLEY-MILNER ALGORITHM, CODE ANALYSIS

Object of research — static type systems, source code in the programming language Clojure.

Purpose — expansion of type system of language Clojure by developing a system for checking source code for correctness in terms of matching types.

Following tasks have been completed during the work: explore different ways to implement a static type system, implemented a system for type checking for projects in the language Clojure, implemented parser for source code in Clojure, implemented Hindley-Milner type inference algorithm, developed tools for programmer to interact with the type checking system.

The work is of practical importance in the development of code, because it allows Clojure developers to analyze source code of their projects for compliance types. And there is no need to annotate specific types for all functions and variables.

The developed system can be freely used in any project that uses programming language Clojure. The system does not require time-consuming design change for its use and does not greatly retards further development.