



Министерство науки и высшего образования Российской Федерации  
федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ      «Робототехника и комплексная автоматизация»  
КАФЕДРА        «Системы автоматизированного проектирования (РК-6)»

## РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по дисциплине «Модели и методы анализа проектных  
решений»

на тему

«Разработка механизма вывода типов с использованием линейных  
систем типов»

Студент РК6-75Б  
группа

\_\_\_\_\_  
подпись, дата

Никитин В.Л.  
ФИО

Руководитель КП

\_\_\_\_\_  
подпись, дата

Соколов А.П.  
ФИО

Консультант

\_\_\_\_\_  
подпись, дата

Соколов А.П.  
ФИО

Москва, 2023

Министерство науки и высшего образования Российской Федерации  
федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

---

УТВЕРЖДАЮ

Заведующий кафедрой РК-6  
индекс

\_\_\_\_\_ А.П. Карпенко

« \_\_\_\_\_ » \_\_\_\_\_ 2023 г.

## ЗАДАНИЕ

### на выполнение курсового проекта

Студент группы: РК6-75Б

Никитин Владимир Леонидович

\_\_\_\_\_ (фамилия, имя, отчество)

Тема курсового проекта: Разработка механизма вывода типов с использованием линейных систем типов

Источник тематики (кафедра, предприятие, НИР): кафедра

Тема курсового проекта утверждена на заседании кафедры «Системы автоматизированного проектирования (РК-6)», Протокол № \_\_\_\_\_ от « \_\_\_\_\_ » \_\_\_\_\_ 2023 г.

### Техническое задание

#### Часть 1. Аналитический обзор литературы.

Более подробная формулировка задания. Следует сформировать, исходя из исходной постановки задачи, предоставленной руководителем изначально. Формулировка включает краткое перечисление подзадач, которые требовалось реализовать, включая, например: анализ существующих методов решения, выбор технологий разработки, обоснование актуальности тематики и пр. Например: «Должен быть выполнен аналитический обзор литературы, в рамках которого должны быть изучены вычислительные методы, применяемые

для решения задач кластеризации больших массивов данных. Должна быть обоснована актуальность исследований.»

**Часть 2.** Математическая постановка задачи, разработка архитектуры программной реализации, программная реализация.

Более подробная формулировка задания. Формулировка заголовка части может отличаться от работы к работе (например, может быть просто «Математическая постановка задачи» или «Архитектура программной реализации»), что определяется конкретной постановкой задачи. Содержание задания должно детальнее и обязательно конкретно раскрывать заголовок. Например: «Должна быть создана математическая модель распространения вирусной инфекции и представлена в форме системы дифференциальных уравнений».

**Часть 3.** Проведение вычислительных экспериментов, тестирование.

Более подробная формулировка задания. Должна быть представлена некоторая конкретизация: какие вычислительные эксперименты требовалось реализовать, какие тесты требовалось провести для проверки работоспособности разработанных программных решений. Формулировка задания должна включать некоторую конкретику, например: какими средствами требовалось пользоваться для проведения расчетов и/или вычислительных экспериментов. Например: «Вычислительные эксперименты должны быть проведены с использованием разработанного в рамках ВКР программного обеспечения».

### **Оформление курсового проекта:**

Расчетно-пояснительная записка на 21 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

количество: 1 рис., 1 табл., 4 источн.
/здесь следует ввести по пунктам наименования чертежей, графических материалов, плакатов/

Дата выдачи задания «01» октября 2023 г.

**Студент**

\_\_\_\_\_  
подпись, дата

Никитин В.Л.  
ФИО

**Руководитель курсового проекта**

\_\_\_\_\_  
подпись, дата

Соколов А.П.  
ФИО

Примечание. Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

## РЕФЕРАТ

курсовой проект: 21 с., 1 рис., 1 табл., 4 источн.

@KEYWORDSRU@.

@Начать можно так: “Работа посвящена...”. Объём около 0.5 страницы. Здесь следует кратко рассказать о чём работа, на что направлена, что и какими методами было достигнуто. Реферат должен быть подготовлен так, чтобы после её прочтения захотелось перейти к основному тексту работы.@

**Тип работы:** курсовой проект.

**Тема работы:** *«Разработка механизма вывода типов с использованием линейных систем типов».*

**Объект исследования:** @Объект исследований@.

**Основная задача, на решение которой направлена работа:** @Основная задача, на решение которой направлена работа@.

**Цели работы:** @Цель выполнения работы@

В результате выполнения работы: 1) предложено ...; 2) создано ...; 3) разработано ...; 4) проведены вычислительные эксперименты ...

# СОКРАЩЕНИЯ

**ПО** Программное обеспечение.

# СОДЕРЖАНИЕ

<b>СОКРАЩЕНИЯ</b> .....	6
<b>ВВЕДЕНИЕ</b> .....	8
<b>1 Постановка задачи</b> .....	9
1.1 Концептуальная постановка задачи . . . . .	9
1.2 Математическая постановка задачи . . . . .	9
<b>2 Программная реализация</b> .....	13
2.1 Архитектура . . . . .	13
<b>3 Тестирование и отладка</b> .....	15
3.1 ... . . . .	15
<b>4 Вычислительный эксперимент</b> .....	16
4.1 ... . . . .	16
<b>5 Анализ результатов</b> .....	17
5.1 ... . . . .	17
<b>ЗАКЛЮЧЕНИЕ</b> .....	18
<b>Литература</b> .....	19
<b>ПРИЛОЖЕНИЯ</b> .....	20
<b>А</b> .....	20

# ВВЕДЕНИЕ

Повышение удобства и эффективности разработки ПО является важной задачей в сфере информатики. Сюда входит как проектирование различных библиотек и фреймворков, так и создание пакетов и средств разработки, в том числе и языков программирования.

Говоря о языках программирования, их можно разделить на две группы: со статической типизацией - языки, в которых типы переменных известны во время компиляции

с динамической типизацией - языки, где проверка соответствия типов происходит по время работы программы

У обоих подходов есть как плюсы, так и минусы, однако подход со статической типизацией дает возможность разработчику заранее увидеть возможные ошибки, поэтому при проектировании больших программ удобнее использовать именно его.

Чаще всего у компилятора есть достаточно информации и для **вывода типов**. Этот механизм позволяет разработчику не указывать большую часть информации о типах в процессе написания программы.

Для вывода типов и их проверки существует большое количество алгоритмов, описанных и формализованных посредством **теории типов**.

**В последнем абзаце** введения следует указывать цель работы в целом.



Д-  
а  
-  
И,  
О-  
у  
ь  
я

2021

## a

## sa2s

5

Alexa

Таблица 1. Изоморфизм Карри-Ховарда

Логическое высказывание	Язык программирования
Высказывание, $F, Q$	Тип, $A, B$
Доказательство высказывания $F$	$x : A$
Высказывание доказуемо	Тип $A$ обитаем
$F \implies Q$	Функция, $A \rightarrow B$
$F \wedge Q$	Тип-произведение, $A \times B$
$F \vee Q$	Тип-сумма, $A + B$
Истина	Единичный тип, $\top$
Ложь	Пустой тип, $\perp$
$\neg F$	$A \rightarrow \perp$

Тип  $T$  обитаем (англ. *inhabitat*), если выполняется следующее:  $\exists t : \Gamma \vdash t : T$

Наборы суждений образуют предположения (англ. *assumptions*), которые образуют контекст. Правила вывода записываются следующим образом, например правило подстановки:

$$\frac{\Gamma \vdash t : T_1, \Delta \vdash T_1 = T_2}{\Gamma, \Delta \vdash t : T_2} \quad (1.1)$$

Выражение 1.1 можно трактовать следующим образом: если в контексте  $\Gamma$  терм  $t$  имеет тип  $T_1$ , а в контексте  $\Delta$  тип  $T_1$  равен типу  $T_2$ , то можно судить, что при наличии обоих контекстов, терм  $t$  имеет тип  $T_2$ .

### 1.2.2 Система типов Хиндли-Милнера

В результате работы над научно-исследовательской работой, было принято решение использовать систему типов Хиндли-Милнера. Среди прочих ее особенностей, важно отметить то, что она способна вывести наиболее общий тип выражения, основываясь на аннотациях типов программиста и окружающем контексте. Здесь приведена её небольшая модификация с добавлением типов-объединений и примитивных типов.

Наиболее классическим алгоритмом в этой области является так называемый алгоритм  $W$  [4].

Для определения системы типов необходимо 3 составляющие: набор термов, набор типов и набор суждений.

### Термы:

$a, b, c ::=$

$x$	(переменная)
$\lambda x. a$	(лямбда-функция)
$a(b)$	(применение аргумента к функции)
$let\ a = b\ in\ c$	(объявление переменной)
$1, 2, 3, \dots$	(целочисленный литерал)
$1.1, 1.2, 10.0, \dots$	(вещественный литерал)
$(a, b)$	(объединение)

### Типы:

$\iota ::=$	(примитивный тип)
$Integral$	(целочисленный)
$Floating$	(вещественный)
$\tau, \sigma ::=$	(мономорфный тип)
$\iota$	
$T$	(переменная типа)
$\tau \rightarrow \sigma$	(функциональный тип)
$(\tau, \sigma)$	(тип-объединение)
$\Lambda$	(пользовательский тип)
$\alpha ::=$	(полиморфный тип)
$\tau$	
$\forall a. \alpha$	(параметрический тип)

Алгоритм  $W$  работает, исходя из набора суждений:

$$\overline{\Gamma \vdash x : \sigma} \quad (\text{TAUT})$$

$$\frac{\Gamma \vdash x : \sigma, \sigma' < \sigma}{\Gamma \vdash x : \sigma'} \quad (\text{INST})$$

Запись  $\sigma' < \sigma$  означает, что тип  $\sigma'$  более конкретный, чем  $\sigma$ .

$$\frac{\Gamma \vdash x : \sigma, a \notin \text{free}(\Gamma)}{\Gamma \vdash x : \forall a. \sigma} \quad (\text{GEN})$$

$$\frac{\Gamma \vdash f : \tau \rightarrow \tau', x : \tau}{\Gamma \vdash f(x) : \tau'} \quad (\text{COMB})$$

$$\frac{\Gamma \cup x : \tau \vdash y : \tau'}{\Gamma \vdash \lambda x. y : \tau \rightarrow \tau'} \quad (\text{ABS})$$

$$\frac{\Gamma \vdash x : \sigma, \Gamma \cup y : \sigma \vdash z : \tau}{\Gamma \vdash (\text{let } y = x \text{ in } z) : \tau} \quad (\text{LET})$$

$$\frac{\Gamma \vdash x_1 : \tau_1, x_2 : \tau_2, x_3 : \tau_3, \dots}{\Gamma \vdash (x_1, x_2, x_3, \dots) : (\tau_1, \tau_2, \tau_3, \dots)} \quad (\text{TUPLE})$$

Исходя из этих суждений, алгоритм  $W$  составляет так называемое дерево вывода. Если дерево построить удалось, то написанная программа считается верной.

## 2 Программная реализация

### 2.1 Архитектура

Изначально компилятор был написан на языке программирования Kotlin, однако было принято решение переписать его на другой язык - Rust. Основные преимущества:

- Rust работает быстрее, так как использует при компиляции `llvm`, что значит более хорошую оптимизацию, а так же применяет более строгие требования к разработке в целом,
- он предоставляет больше гарантий разработчику, как посредством его системы типов, так и другими средствами, например, `borrow checker`,
- система сборки создает нативный файл программы, в то время как система сборки Kotlin - файл, зависимый от JRE

Основной проблемой как раз и стало недостаточное количество гарантий во время компиляции и усложнение разработки из-за этого. Даже если ошибка отлавливается с помощью проверок во время исполнения, её бывало трудно воспроизвести.

Отсюда и возникли дополнительные задачи по проектированию новой структуры AST программы, взаимодействием с этим деревом, а так же самым главным - разработке семантического анализатора.

Работать с большими проектами в разы удобнее при грамотном разбиении на модули. В экосистеме Rust такие модули именуются крейтами (англ. `crates`). На диаграмме ниже представлено разбиение на модули проекта `Kodept`.

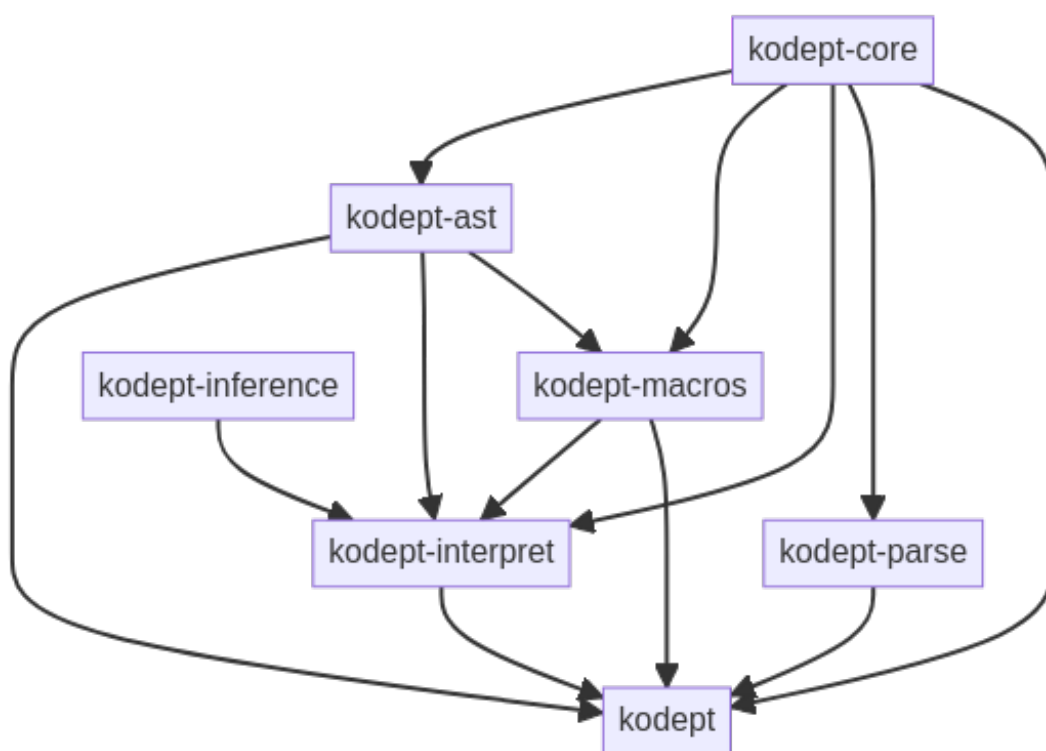


Рисунок 1. Иерархия модулей в проекте

## 3 Тестирование и отладка

### 3.1 ...

В разделе следует представить описания тестовых примеров, включая входные данные, принципы запуска и указать ожидаемый результат и фактически полученный.

Допускается включение скриншотов, однако, каждый должен быть подписан и представлено обоснование его включение в РПЗ.

Обязательность представления: раздел представляется в зависимости от постановки задачи.

Объём: около 4-5 страниц.

## 4 Вычислительный эксперимент

### 4.1 ...

В разделе следует представить описания каждого вычислительного эксперимента, включая указание особенностей их проведения, используемые программные средства, используемые исходные данные, принципы запуска с указанием ожидаемого и полученного результата.

Обязательно представление графических результатов в форме графиков, поверхностей.

Обязательность представления: раздел представляется в зависимости от постановки задачи.

Объём: объём не ограничен, но, как правило, не должен быть меньше 5-6 страниц.



## 5 Анализ результатов

### 5.1 ...

В разделе следует представить анализ полученных результатов, включая указание перспектив развития созданных научно-технических решений.

Обязательность представления: раздел обязателен.

Объём: объём не ограничен, но, как правило, не должен быть меньше 2 страниц.

## ЗАКЛЮЧЕНИЕ

В разделе следует представить выводы по работе в целом. Каждый вывод **не должен** быть банальным указанием факта реализации поставленных задач. Каждый вывод должен быть результатом проведенной работы в целом, включая результаты тестирования, вычислительных экспериментов и анализа результатов.

Обязательность представления: раздел обязателен.



Объём: как правило, не должен быть больше 1-2 страниц.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Milner Robin. A theory of type polymorphism in programming // Journal of computer and system sciences 17. 1978. С. 348–375.
- 2 The Foundations of Arithmetic // The Legacy of Mario Pieri in Geometry and Arithmetic. Boston, MA: Birkhäuser Boston, 2007. С. 289–329.
- 3 Свиридов Сергей. Теория типов [Электронный ресурс]. 2023. (Дата обращения 19.04.2024). URL: <https://habr.com/ru/articles/758542/>.
- 4 Urban Christian, Nipkow Tobias. Nominal verification of algorithm W // From Semantics to Computer Science. Essays in Honour of Gilles Kahn / под ред. G. Huet, J.-J. Lévy, G. Plotkin. Cambridge University Press, 2009. С. 363–382.

### Выходные данные

*Никитин В.Л.. Разработка механизма вывода типов с использованием линейных систем типов по дисциплине «Модели и методы анализа проектных решений». [Электронный ресурс] — Москва: 2023. — 21 с. URL: <https://sa2systems.ru:88> (система контроля версий кафедры РК6)*

Постановка:  @должность научного руководителя@, Соколов А.П.  
Решение и вёрстка:  студент группы РК6-75Б, Никитин В.Л.

2023, осенний семестр

**A**