Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный** **исследовательский политехнический университет»**

Факультет: Прикладной математики и механики

Кафедра: Вычислительной математики, механики и биомеханики

Направление: 09.03.02 «Информационные системы и технологии»

У Т В Е Р Ж Д А Ю

**Зав. кафедрой ВММБ**

\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Ю. Столбов

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

***ЗАДАНИЕ***

***НА КУРСОВУЮ РАБОТУ***

***по дисциплине***

**«ЯЗЫКИ ПРОГРАММИРОВАНИЯ И МЕТОДЫ ТРАНСЛЯЦИИ»**

Васин Михаил Александрович ИСТ-22-1б

(фамилия, имя, отчество студента; группа)

1. **Тема курсовой работы**

Язык управления роботом

2. **Срок сдачи студентом отчета:**

09.01.2024

3. **Содержание отчета:**

В курсовой работе был разработан язык программирования для управления роботом, который способен

двигаться в произвольном направлении, выполнять заданное количество действий в зависимости от условий внешней среды, считать количество выполненных действий.

С.Е. Батин

Руководитель курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (расшифровка)

М.А. Васин

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (расшифровка)

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный** **исследовательский**

**политехнический университет»**

Факультет: Прикладной математики и механики

Кафедра: Вычислительной математики, механики и биомеханики

Направление: 09.03.02 «Информационные системы и технологии»

**КУРСОВАЯ РАБОТА**

по дисциплине

**«ЯЗЫКИ ПРОГРАММИРОВАНИЯ И МЕТОДЫ ТРАНСЛЯЦИИ»**

Тема: **«Создание языка для управления роботом»**

Выполнил:

ИСТ-22-1б

студент гр.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Васин Михаил Александрович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*(подпись, дата)*

Принял:

Доцент каф. ВММБ, к.т.н. С.Е. Батин

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(должность, ФИО)*

*\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

Пермь, 2023

Введение

Эта курсовая работа посвящена созданию нового языка программирования для управления роботом. Основная цель - разработать язык, который позволит роботу эффективно выполнять свои задачи и адаптироваться к разным условиям.

В работе используются инструменты flex и bison для создания лексического и синтаксического анализаторов. Flex помогает разбить входной текст программы на отдельные элементы (токены), а bison - упорядочить эти токены согласно правилам нашего языка.

В курсовой представлено описание грамматики нового языка, примеры кода анализаторов на flex и bison, а также обсуждаются возможности и ограничения языка. Целью является создание языка, который был бы прост в использовании для управления роботом-маляром и при этом достаточно гибок для различных задач.

Теория

**Лексический Анализатор (Flex)**

Лексический анализатор, созданный с помощью Flex, является первым шагом в процессе обработки исходного кода вашего языка программирования. Его основная задача - сканировать текст программы и преобразовывать его в серию токенов, которые представляют собой осмысленные элементы языка, такие как ключевые слова, идентификаторы, числа, операторы и так далее.

Основные функции лексического анализатора:

* Разбиение Текста: Проходит через весь исходный текст и разбивает его на отдельные элементы (токены).
* Удаление Пробелов и Комментариев: Игнорирует незначащие символы, такие как пробелы, табуляции и комментарии.
* Классификация Токенов: Определяет тип каждого токена (например, является ли он числом, ключевым словом или оператором).
* Отправка Токенов: Передает токены в синтаксический анализатор для дальнейшей обработки.

Синтаксический Анализатор (Bison)

Синтаксический анализатор, созданный с помощью Bison, принимает токены от лексического анализатора и строит из них синтаксическое дерево, которое представляет структуру исходного кода согласно грамматике языка программирования.

Основные функции синтаксического анализатора:

* Проверка Структуры: Проверяет, соответствует ли последовательность токенов правилам грамматики языка.
* Построение Синтаксического Дерева: Формирует дерево, которое представляет иерархическую структуру программы.
* Обработка Синтаксических Ошибок: Обнаруживает и сообщает об ошибках в структуре кода, например, если в коде пропущена закрывающая скобка.
* Подготовка к Дальнейшему Анализу: Подготавливает структурированные данные для семантического анализа или генерации кода.

Грамматика языка

Основные Конструкции:

* Команды Движения: Включают команды для перемещения робота, например, up, down, left, right. Каждая команда может сопровождаться параметром, указывающим количество шагов.
* Операции: Определяют действия робота, такие как paint (красить) или nothing (ничего не делать). Эти операции могут использоваться в сочетании с командами движения.
* Условные Выражения: Позволяют роботу принимать решения на основе условий окружающей среды. Пример: if (area is clear up) {...}.
* Циклы: Используются для повторения действий, пока выполняется определенное условие, например, while (area is clear right) {...}.

В контексте курсовой работы, используются следующие файлы, каждый из которых выполняет определенную функцию в процессе создания и работы языка:

* lexer.l

Это файл Flex, содержащий определения шаблонов для лексического анализатора. Он используется для разбиения исходного кода на токены (например, ключевые слова, числа, идентификаторы).

* parser.y

Этот файл является основным файлом Bison, который содержит описание грамматики вашего языка программирования. В нем определяются правила синтаксиса, которые используются для анализа исходного кода и построения синтаксических структур (например, синтаксических деревьев).

* input.txt

Этот файл содержит входные данные для программы. В контексте управления роботом, он включает размерность и описание лабиринта, в котором должен работать робот (0 – свободная территория, 1 – занятая территория).

* position.txt

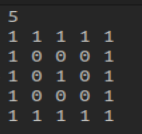
Файл position.txt содержит начальную позицию робота в лабиринте. Эти данные используются программой для инициализации состояния робота перед началом его работы.

* prog.txt

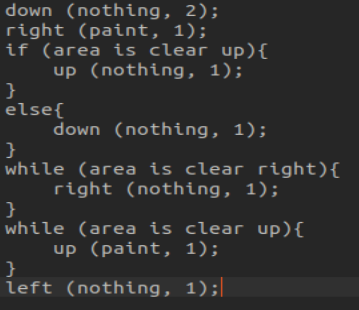
В этом файле находится программа, написанная на разработанном языке программирования. Она содержит последовательность команд, которые робот должен выполнить.

Пример работы языка программирования

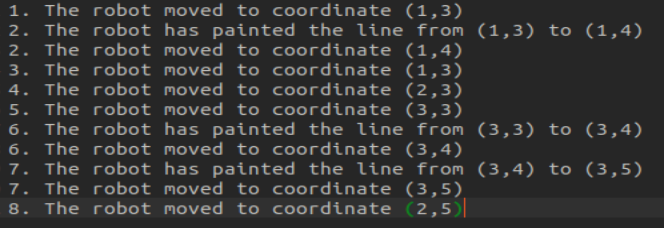
Входные данные из input.txt



Входные данные из prog.txt



Результат работы языка программирования в output.txt



Анализ возможностей и ограничений

**Возможности:**

* Легкость изучения и использования: Благодаря ограниченному и простому синтаксису, язык легко изучить и использовать, что делает его доступным.
* Четкое определение поведения робота: Язык позволяет точно и ясно определять поведение робота, что уменьшает вероятность ошибок в его работе.
* Возможности для расширения: Хотя язык и имеет ограниченный функционал, его структура позволяет потенциальное расширение возможностей в будущем.

**Ограничения:**

* Ограниченная область применения: Язык предназначен исключительно для управления роботом, что ограничивает его применение в других сценариях.
* Отсутствие поддержки пользовательских переменных: Невозможность создания пользовательских переменных ограничивает гибкость языка и может затруднить написание сложных программ.
* Зависимость от специализированных инструментов: Работа языка зависит от инструментов flex и bison, что может ограничивать его портативность и доступность.

Заключение

В рамках данной курсовой работы был разработан специализированный язык программирования для управления роботом. Используя инструменты flex и bison, созданы лексический и синтаксический анализаторы, что позволило реализовать четко определенную грамматику языка. Язык ориентирован на выполнение конкретных задач по управлению движениями робота и реакции на условия окружающей среды.