Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

по курсу «ЛОИС»

Вариант 14

Выполнил: Германенко В.В.,

гр. 321701

Проверил: Ивашенко В.П.

**МИНСК**

2025

**Тема:** Логическое программирование поиска решения задачи.

**Цель**: Приобрести навыки логического программирования поиска решения задачи.

**Условие задачи**: Три путешественника едут на велосипедах по прямому шоссе из пункта А в пункт Б. Скорость движения велосипедиста на велосипеде - v км\ч. Один из велосипедов ломается, определить порядок движения путешественников из пункта А в пункт Б, при котором время движения t минимально, при условии, что: скорость транспортировки велосипедистом сломанного велосипеда – y км\ч; скорость движения пешехода – w км\ч; скорость пешехода, ведущего велосипед, – u км\ч. (v >= y >= w >= u)

**Дополнительные теоретические сведения**

Грамматика языка PROLOG.

<ПРОЛОГ-предложение> ::= <правило> | <факт> | <запрос>

<правило> ::= <заголовок> ‘:-’<тело>

<факт> ::= <заголовок> ‘.’

<запрос> ::= <тело>‘.’

<тело> ::= <цель> /’,’<цель>/’.’

<заголовок>::= <предикат>

<цель>::= <предикат> |<выражение>

<предикат>::= <имя>/ ‘(‘<терм> /’,’<терм>/ ‘)’/

<терм>::= <атом> |<предикат>|<список>

<атом>::= <переменная> |<число> |<строка>|<имя>

<список>::= <список с заголовком>| <простой список>

<список с заголовком >::= ‘[‘ <терм >/’,’<терм>/’|’ < терм>’]’

<простой список>::= ‘[‘ <терм >/’,’<терм>/’]’|‘['’]’

<выражение>::= <терм> /<оператор><терм>/

<оператор>::= ‘is’ | '=' | ‘==' | ’\=' | ’>=' | ’=<’ | ‘=\=' | ‘=:=’

**Программная реализация:** В рамках лабораторной работы средствами языка Prolog был реализован алгоритм, решающий задачу о трех путешественниках. Алгоритм основан на аналитическом расчете оптимальной стратегии, при которой два велосипедиста движутся до середины пути, один оставляет велосипед, другой доезжает до пункта Б, а пешеход с велосипедом идет от середины до конца.

Пример выполнения:

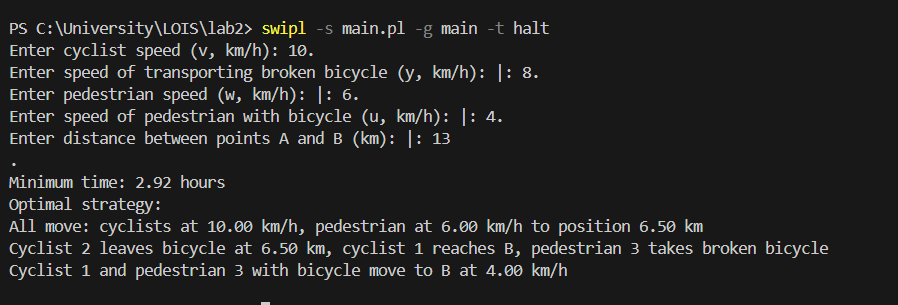


Рис. 1 – Пример выполнения программы

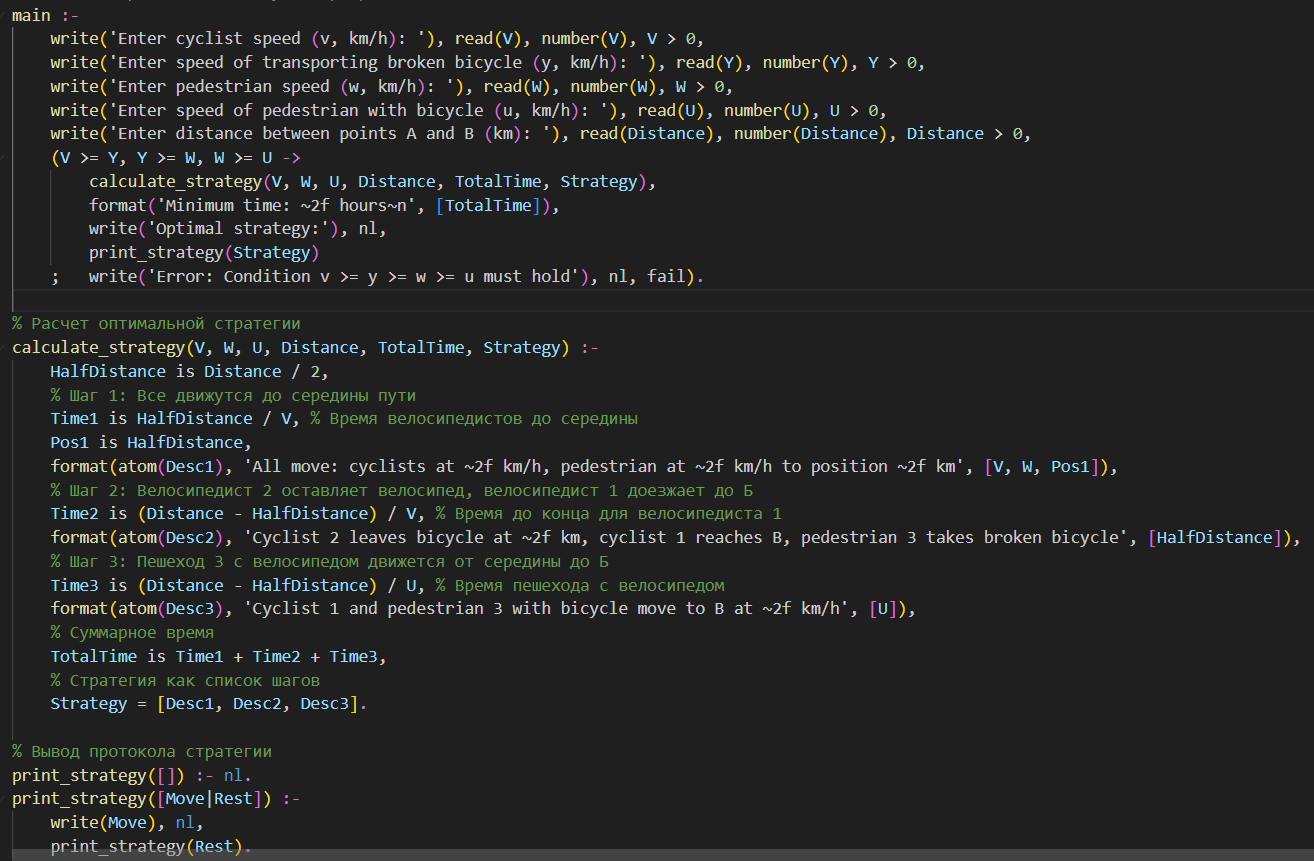
Листинг программы:

Рис. 2 – Листинг программы

### В программе используются:

В программе для решения задачи о трех путешественниках применяются следующие компоненты языка Prolog:

#### 1. Встроенные предикаты

* **write/1**: Выводит текстовую строку в консоль. Используется для отображения приглашений для ввода данных.
* **read/1**: Считывает терм, введенный пользователем. Применяется для получения значений скоростей и расстояния.
* **number/1**: Проверяет, является ли введенный терм числом. Используется для валидации ввода.
* **>/2, >=/2**: Операторы сравнения чисел. Используются для проверки корректности введенных данных и выполнения условия v >= y >= w >= u.
* **is/2**: Вычисляет арифметическое выражение справа и унифицирует результат с переменной слева. Применяется для всех расчетов времени и расстояния.
* **format/2, format/3**: Форматирует атом или выводит форматированную строку в консоль. Используется для создания описания шагов стратегии и вывода итогового времени.
* **nl/0**: Переводит курсор на новую строку в консоли. Применяется для форматирования вывода.
* **fail/0**: Предикат, который всегда завершается неудачей. Используется для остановки программы при неверных входных данных.

#### 2. Переменные

* **V, Y, W, U**: Переменные для хранения скоростей: велосипедиста, транспортировки сломанного велосипеда, пешехода и пешехода с велосипедом соответственно.
* **Distance**: Хранит общее расстояние от пункта А до пункта Б.
* **TotalTime**: Итоговое минимальное время, рассчитанное программой.
* **Strategy**: Список, содержащий атомы с описанием каждого шага оптимальной стратегии.
* **HalfDistance, Time1, Time2, Time3, Pos1**: Вспомогательные переменные для промежуточных расчетов расстояний и времени на каждом этапе.
* **Desc1, Desc2, Desc3**: Переменные, хранящие текстовое описание шагов стратегии.

#### 3. Правила

* **main/0**: Основное правило, запускающее программу. Оно организует ввод данных, их проверку, вызывает правило расчета стратегии и выводит результат.
* **calculate\_strategy/6**: Ядро программы. Это правило рассчитывает оптимальную стратегию и общее время в пути на основе аналитической модели, описанной в алгоритме.
* **print\_strategy/1**: Рекурсивное правило, которое последовательно выводит каждый шаг сохраненной стратегии в консоль.

Дерево вывода:

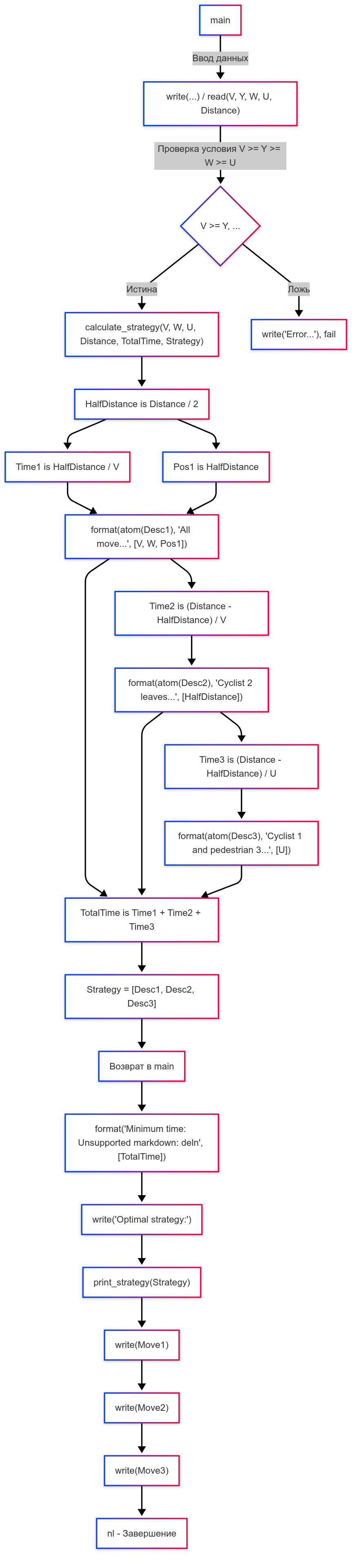


Рис. 3 – Дерево вывода

Формализация на языке логики предикатов первого порядка:

**Константы:**

* 0, 2, 3 - числовые константы, используемые в расчетах и для определения длины списка.
* A, B - символические константы для пунктов отправления и назначения.

**Предикаты:**

* Calculate\_strategy(V,W,U,D,T,S): Истинно, если T и S являются итоговым временем и стратегией для входных данных V,W,U,D.
* Length(L, N): Истинно, если список L имеет длину N.
* Is\_Step1(Desc, V, W, D), Is\_Step2(Desc, D), Is\_Step3(Desc, U): Истинно, если Desc является текстовым описанием соответствующего шага.

**Аксиомы:**

∀V, ∀W, ∀U, ∀D, ∀T, ∀S(Calculate\_strategy(V,W,U,D,T,S)→T=(D/2)/V+(D−D/2)/V+(D−D/2)/U)

∀V, ∀W, ∀U, ∀D, ∀T, ∀S (Calculate\_strategy(V,W,U,D,T,S)→Length(S,3))

∀V, ∀W, ∀U, ∀D, ∀T, ∀S(Calculate\_strategy(V,W,U,D,T,S)→S=[Desc1,Desc2,Desc3])

∀V, ∀W, ∀U, ∀D, ∀T, ∀S(Calculate\_strategy(V,W,U,D,T,S)→Is\_Step1(Desc1,V,W,D))

∀V, ∀W, ∀U, ∀D, ∀T, ∀S(Calculate\_strategy(V,W,U,D,T,S)→Is\_Step2(Desc2,D))

∀V, ∀W, ∀U, ∀D, ∀T, ∀S(Calculate\_strategy(V,W,U,D,T,S)→Is\_Step3(Desc3,U))

Вывод: В ходе выполнения лабораторной работы были приобретены навыки логического программирования поиска решения задачи. Разработана программа, решающая задачу о трех путешественниках с велосипедами, которая определяет оптимальное время и стратегию перемещения из пункта А в пункт Б.

Список литературы:

Derek Banas (2015) Prolog Tutorial [обучающее видео]//YouTube. (<https://youtu.be/SykxWpFwMGs>).

2. SWI-Prolog Documentation. (https://www.swi-prolog.org).