МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра программных систем\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**  
  
по лабораторной работе №1

«Работа со списками, Unit тестирование»

по дисциплине «Логическое программирование»

Обучающаяся в группе 6301-020302D Соколова Алёна

Преподаватель Лобанков Антон Алексеевич

Самара 2025

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc190778412)

[ЗАДАНИЕ 1 4](#_Toc190778413)

[ЗАДАНИЕ 2 5](#_Toc190778414)

[ЗАКЛЮЧЕНИЕ 7](#_Toc190778415)

# ВВЕДЕНИЕ

Целью работы является в процессе выполнения заданий ознакомиться обработкой списков на языке Prolog, с модульным тестированием и созданием графического пользовательского интерфейса на языке Prolog.

Задания:

1. реализовать программу в соответствии с вариантом;
2. написать юнит тесты для предиката из задания 2. рассмотреть все граничные случаи, воспользоваться параметризованными тестами.

Вариант 26: написать программу, возвращающую количество элементов, равных своему предыдущему.

# ЗАДАНИЕ 1

Была написана программа на Prolog, реализующая предикат count\_equal\_previous/2, который возвращает количество элементов в списке, равных своему предыдущему. Код программы отображен на рисунке 1.

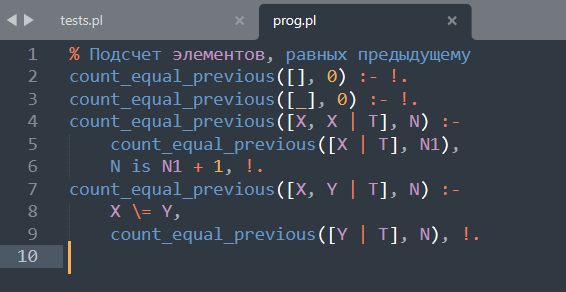


Рисунок 1 – Код файлы prog.pl

# ЗАДАНИЕ 2

Для проверки корректности работы предиката были разработаны юнит-тесты в файле tests.pl с использованием библиотеки plunit. В тестах рассмотрены все граничные случаи. Код программы отображен на рисунке 2.



Рисунок 2 – Код файлы tests.pl

Программа корректно обрабатывает списки различной длины и учитывает повторяющиеся элементы.

Описание тестов:

1. empty\_list: проверяет, что для пустого списка результат равен 0;
2. single\_element: проверяет, что для списка с одним элементом результат равен 0;
3. ascending\_list: проверяет, что для списка с уникальными элементами в порядке возрастания результат равен 0;
4. equal\_elements: проверяет, что для списка с одинаковыми элементами результат равен 3;
5. mixed\_elements: проверяет, что для списка с повторяющимися элементами результат равен 3;
6. alternating\_elements: проверяет, что для списка с чередующимися элементами результат равен 0;
7. count\_equal\_parametrized: проверяет результат для списка с некоторыми повторяющимися элементами.

Для запуска программы и тестов в среде SWI-Prolog были выполнены следующие шаги:

1. загрузка основного файла программы: ?- [prog].
2. загрузка тестов: ?- [tests].
3. запуск тестирования: ?- run\_tests.

После запуска тестов система вывела следующее сообщение, отображенное на рисунке 3. Все тесты успешно пройдены, и программа работает корректно.

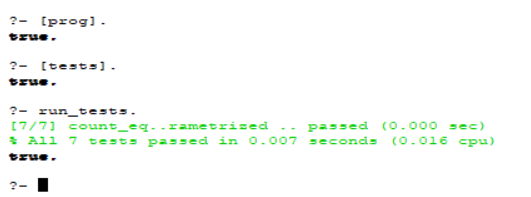


Рисунок 3 – Успешное выполнение тестов

# ЗАКЛЮЧЕНИЕ

В ходе работы изучены обработка списков на Prolog, модульное тестирование и создание графического интерфейса:

1. реализован интерфейс для ввода данных и отображения результатов в модульном окне;
2. программа подсчитывает количество элементов, равных своим предыдущим;
3. написаны юнит-тесты с граничными случаями и параметризованными тестами.