МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»  
(Самарский университет)   
  
  
Факультет информатики  
Кафедра программных систем  
  
Дисциплина  
**Логическое программирование  
  
  
  
ОТЧЕТ**по лабораторной работе №1

Тема: Работа со списками, Unit тестирование, GUI.  
Вариант №9

Студент: Колбанов Д.О.   
Группа: 6301-020302D   
  
Преподаватель: Лобанков А.А.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2024

**Цель и постановка задания:**

В процессе выполнения заданий ознакомиться обработкой списков на языке Prolog, с модульным тестированием и созданием графического пользовательского интерфейса на языке Prolog.

**Задание:**

Написать программу разделения списка на два так, чтобы в первом

списке были элементы с первого до заданного значения элемента, а во втором - остальные.

**Результаты выполнения:**

На рисунке 1 мы можем видеть результаты успешного прохождения тестов для разных случаев.

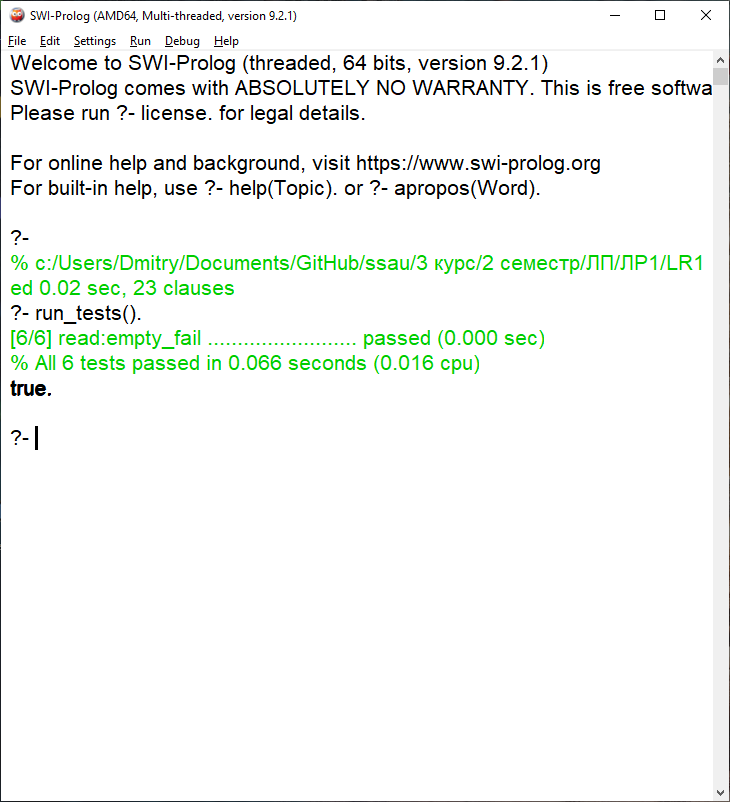


Рисунок 1 – Выполненные юнит-тесты

На следующем рисунке мы видим окно для ввода информации пользователя и окно с выводом результата после обработки введённых данных.

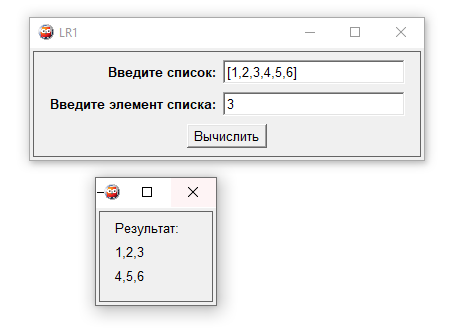


Рисунок 2 – Пользовательский интерфейс и результат обработки

**Код программы**

:-use\_module(library(pce)).

gui:-

new(D, dialog('LR1')),

send(D, append, new(List, text\_item("Введите список"))),

send(D, append, new(Element, text\_item("Введите элемент списка"))),

send(D, append, button("Вычислить", message(@prolog, input, List?selection, Element?selection))),

send(D, open).

input(TextList, TextElement) :-

atom\_codes(TextList, CodesList),

read\_from\_codes(CodesList, List),

atom\_codes(TextElement, CodesElement),

read\_from\_codes(CodesElement, Int),

split\_list\_until\_value(Int, List, List1, List2),

calcResult(List1, List2). % Передаем результаты разделения в предикат calcResult

% Нашли значение, завершаем разделение

split\_list\_until\_value(Value, [Value|Tail], [Value], Tail) :- !.

split\_list\_until\_value(Value, [Head|Tail], [Head|List1], List2) :-

split\_list\_until\_value(Value, Tail, List1, List2).

calcResult(List1, List2):-

new(R, dialog('Result')),

send(R, append, new(text('Результат: '))),

atomic\_list\_concat(List1, ',', Atom1),

atom\_string(Atom1, Result1),

atomic\_list\_concat(List2, ',', Atom2),

atom\_string(Atom2, Result2),

send(R, append, new(text(Result1))), % выводим первую часть списка

send(R, append, new(text(Result2))), % выводим вторую часть списка

send(R, open).

%:-use\_module(plunit).

:-begin\_tests(read).

% Заданный элемент в начале списка

test(first, [true(R1==[1])]):- split\_list\_until\_value(1, [1,2,6,4], R1, [2,6,4]).

% Заданный элемент в середине списка

test(middle):- split\_list\_until\_value(1, [2,6,1,8,4], [2,6,1], [8,4]).

% Заданный элемент в конце списка

test(last):- split\_list\_until\_value(1, [2,6,8,4,1], [2,6,8,4,1], []).

% Тест с пустым списком

test(empty) :- not(split\_list\_until\_value(1, [], [], [])).

% Тест с анонимной переменной и пустым списком

test(anon\_empty, [nondet]):- not(split\_list\_until\_value(\_, [], [], [])).

% Тест с fail

test(empty\_fail, [fail]) :- split\_list\_until\_value(\_, [], \_, \_).

:-end\_tests(read).