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 №5

Тема: Работа со строками.  
Вариант №10

Студент: Лапин К.С.   
Группа: 6303-020302D   
  
Преподаватель: Лобанков А.А.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2024

**Цель и постановка задания:**

В процессе выполнения заданий ознакомиться c обработкой строк на языке Prolog.

**Задание 1**

Реализовать программу в соответствии со своим индивидуальным вариантом. Если в прологе существует предикат - аналог, позволяющий решить ваше задание в одну строку - необходимо написать свою реализацию, не используя его.

**Задание 2**

Продемонстрировать работу программы. Подготовить тестовые примеры для рассмотрения граничных случаев.

**Задание варианта:**

Копирование n символов строки, начиная с i-той позиции. Если n не указано, копируем строку до конца.

**Результаты выполнения:**

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

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описаниеРисунок 1 – Выполненные юнит-тесты

На следующем рисунке 2 мы видим, как пользователь вводит информацию с возможностью ввода количества символов для копирования и вывод результата после обработки введённых данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание  
Рисунок 2 – Пользовательский интерфейс и результат обработки с количеством символов для копирования

На рисунке 3 пользователь вводит те же данные, но не использует количество символов для копирования.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание  
Рисунок 3 – Пользовательский интерфейс и результат обработки без количества символов для копирования

**Код программы**

% Копирование n символов строки, начиная с i-той позиции. Если n не

% указано, копируем строку до конца.

app(SubString):-

write('Введите строку: '),

read\_string(user\_input, '\n', '\r', \_, String),

write('Введите i-ую позицию: '),

read\_string(user\_input, '\n', '\r', \_, SIndex),

number\_string(Index, SIndex),

write('Хотите ли ввести число? y/n: '),

read\_string(user\_input, '\n', '\r', \_, Choice),

atom\_string(Symbol, Choice),

(

Symbol == 'y' ->

write('Введите количество символов для копирования: '),

read\_string(user\_input, '\n', '\r', \_, SNumber),

number\_string(Number, SNumber),

write(''),

copy\_symbols(String, Index, Number, SubString);

copy\_symbols(String, Index, SubString)

).

validator(StartIndex, LastIndex, Number):-

StartIndex >= 0,

StartIndex =< LastIndex,

Number >= 0.

% Предикат для копирования n символов из строки начиная с i-ой позиции

copy\_symbols(String, StartIndex, Number, SubString) :-

string\_length(String, Length),

LastIndex is Length - 1,

(

String == "" ->

% Базовый случай: если строка пустая, возвращаем пустую подстроку

SubString = "";

% Проходим валидацию данных для StartIndex, LastIndex, Number

validator(StartIndex, LastIndex, Number),

AcceptNumber is LastIndex - StartIndex + 1,

(

% Проверяем, превышает ли Number допустимое значение AcceptNumber

Number > AcceptNumber ->

sub\_string(String, StartIndex, AcceptNumber, \_, SubString);

sub\_string(String, StartIndex, Number, \_, SubString)

)

).

% Предикат для копирования всех символов из строки начиная с i-ой

% позиции

copy\_symbols(String, StartIndex, SubString):-

string\_length(String, Length),

LastIndex is Length - 1,

Number is LastIndex - StartIndex + 1,

copy\_symbols(String, StartIndex, Number, SubString).

% Юнит Тесты для предиката copy\_symbols

% run\_tests

%:-use\_module(plunit).

:-begin\_tests(copy\_symbols).

test(empty\_string, [true(Result="")]):-

copy\_symbols("", 0, 0, Result).

test(all\_symbols, [true(Result="hello")]):-

copy\_symbols("hello", 0, 5, Result).

test(start\_index\_0, [true(Result="hel")]):-

copy\_symbols("hello", 0, 3, Result).

test(start\_index\_not\_zero, [true(Result="llo")]):-

copy\_symbols("hello", 2, 3, Result).

test(number\_less\_than\_to\_copy, [true(Result="he")]):-

copy\_symbols("hello", 0, 2, Result).

test(number\_greater\_than\_to\_copy, [true(Result="hello")]):-

copy\_symbols("hello", 0, 10, Result).

test(invalid\_start\_index\_less\_than\_zero, [fail]):-

copy\_symbols("hello", -1, 3, \_).

test(invalid\_start\_index\_great\_than\_length, [fail]):-

copy\_symbols("hello", 10, 3, \_).

test(invalid\_number, [fail]):-

copy\_symbols("hello", 0, -3, \_).