

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики

РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ
по дисциплине «Функциональное и логическое программирование»

Вариант 7

Выполнил:

студент группы ИС-242

Любицкий М. Е.
ФИО студента

Работу проверил: Сороковых Д. А.
ФИО преподавателя

Новосибирск 2024 г.

Оглавление

Задание	3
Выполнение	4
Листинг программы	7

Задание

- 1.7. Осуществите циклический сдвиг в списке на n элементов влево.
- 2.7. В каждой строке текстового файла переставьте слова в обратном порядке. Сформируйте новый файл.
- 3.7. Создайте базу данных о товарах: наименование товара, фасовка, стоимость. Найдите все товары с минимальной стоимостью.

Выполнение

Задача 1: Циклический сдвиг списка на N элементов влево

Ход выполнения:

1. Реализован предикат `cyclic_shift_left/3`, принимающий:
 - Входной список,
 - Количество элементов для сдвига,
 - Результирующий список после сдвига.
2. Предикат построен следующим образом:
 - Сначала вычисляется остаток от деления N на длину списка (`ModN`), чтобы избежать избыточных операций.
 - Затем список разбивается на две части: первые `ModN` элементов и оставшуюся часть.
 - Результирующий список формируется путём конкатенации оставшейся части и первых `ModN` элементов.

Тестирование:

```
1 ?- cyclic_shift_left([1, 2, 3, 4, 5], 2, Res).  
Res = [3, 4, 5, 1, 2].  
  
2 ?- cyclic_shift_left([1, 2, 3, 4, 5], 6, Res).  
Res = [2, 3, 4, 5, 1].
```

Задача 2: Перестановка слов в строках текстового файла

Ход выполнения:

1. Реализован предикат `reverse_words_in_file/2`, который:
 - Считывает строки из исходного файла.
 - Для каждой строки:
 - Разбивает её на слова,
 - Переворачивает порядок слов,
 - Собирает строку обратно.
 - Записывает результаты в новый файл.
2. Используются стандартные предикаты для работы с файлами (`open/3`, `read_line_to_string/2`, `write/2`) и обработки строк.

Тестирование:

Входной файл:

```
rgr > input.txt
1 Hello world
2 Reverse the words in this line
```

Выходной файл:

```
rgr > output.txt
1 world Hello
2 line this in words the Reverse
3 |
```

Запуск программы

```
?- reverse_words_in_file('input.txt', 'output.txt').
true.
```

Задача 3: Работа с базой данных товаров

1. Загружать базу данных из файла.
2. Обеспечивать работу с меню, включая просмотр, добавление, удаление записей и выполнение запроса.
3. Сохранять базу данных обратно в файл.

Ход выполнения:

1. **Структура данных:**
 - Записи базы данных представлены динамическими фактами `product(Name, Packaging, Price)`.
2. **Функционал:**
 - **Просмотр базы данных:** Предикат `view_database` перебирает все записи и выводит их в удобном формате.
 - **Добавление записей:** Предикат `add_records` позволяет вводить новые факты, используя рекурсивное чтение до команды `stop`.
 - **Удаление записей:** Предикат `delete_records` удаляет указанные записи аналогично добавлению.
 - **Запрос:** Предикат `find_min_price` находит товары с минимальной стоимостью.

- **Сохранение:** Все изменения записываются в файл через `save_database`.

3. Меню:

- Реализовано через предикат `menu`, предоставляющий пользователю интерфейс выбора действий.

Тестирование:

```
?- start.
Меню:
1. Просмотр содержимого базы данных
2. Добавление записей
3. Удаление записей
4. Запрос: Найти товары с минимальной стоимостью
5. Выход
|: 2.
Введите записи. Введите 'stop.' для завершения.
|: product('Milk', '1L', 50).
|: product("Apples", "1kg", 40).
|: stop.
Меню:
1. Просмотр содержимого базы данных
2. Добавление записей
3. Удаление записей
4. Запрос: Найти товары с минимальной стоимостью
5. Выход
|: 1.
[Milk,1L,50]
[Apples,1kg,40]
Меню:
1. Просмотр содержимого базы данных
2. Добавление записей
3. Удаление записей
4. Запрос: Найти товары с минимальной стоимостью
5. Выход
|: 4.
Товары с минимальной стоимостью:
[Apples,1kg,40]
Меню:
1. Просмотр содержимого базы данных
2. Добавление записей
3. Удаление записей
4. Запрос: Найти товары с минимальной стоимостью
5. Выход
|: 5.
Данные сохранены. Выход.
true .
```

Содержимое базы данных:

```
rgr > database.pl
1  :- dynamic product/3.
2
3  product('Milk', '1L', 50).
4  product("Apples", "1kg", 40).
5
6
```

Листинг программы

Задача 1:

cyclic_shift_left(List, N, Result) :-

length(List, Len),

Shift is N mod Len,

shift_left(List, Shift, Result).

shift_left(List, 0, List) :- !.

shift_left([H|T], N, Result) :-

append(T, [H], Temp),

N1 is N - 1,

shift_left(Temp, N1, Result).

Задача 2:

reverse_words_in_file(InputFile, OutputFile) :-

open(InputFile, read, InStream),

open(OutputFile, write, OutStream),

process_lines(InStream, OutStream),

close(InStream),

close(OutStream).

process_lines(InStream, OutStream) :-

read_line_to_string(InStream, Line),

(Line \= end_of_file

-> split_string(Line, " ", "", Words),

reverse(Words, ReversedWords),

atomic_list_concat(ReversedWords, " ", ReversedLine),

```
writeln(OutStream, ReversedLine),  
process_lines(InStream, OutStream)  
; true  
).
```

Задача 3:

`:- dynamic product/3.`

`load_database(File) :-`

`exists_file(File), !,`

`consult(File).`

`load_database(_).`

`save_database(File) :-`

`tell(File),`

`listing(product/3),`

`told.`

`view_database :-`

`forall(product(Name, Packaging, Price),`

`writeln([Name, Packaging, Price])).`

`add_records :-`

`writeln("Введите записи. Введите 'stop.' для завершения."),`

`read_record.`

`read_record :-`


```

read(Product),

( Product == stop

-> true

; assertz(Product),

  read_record

).

```

delete_records :-

```

writeln("Введите записи для удаления. Введите 'stop.' для завершения."),

read_deletion.

```

read_deletion :-

```

read(Product),

( Product == stop

-> true

; retract(Product),

  read_deletion

).

```

find_min_price :-

```

findall(Price, product(_, _, Price), Prices),

min_list(Prices, MinPrice),

findall([Name, Packaging, MinPrice], product(Name, Packaging, MinPrice), Results),

writeln("Товары с минимальной стоимостью:"),

forall(member(Result, Results), writeln(Result)).

```

menu :-

```
writeln("Меню:"),  
  
writeln("1. Просмотр содержимого базы данных"),  
  
writeln("2. Добавление записей"),  
  
writeln("3. Удаление записей"),  
  
writeln("4. Запрос: Найти товары с минимальной стоимостью"),  
  
writeln("5. Выход"),  
  
read(Choice),  
  
handle_choice(Choice).
```

handle_choice(1) :- view_database, menu.

handle_choice(2) :- add_records, menu.

handle_choice(3) :- delete_records, menu.

handle_choice(4) :- find_min_price, menu.

**handle_choice(5) :- save_database('database.pl'), writeln("Данные сохранены. Выход."),
!.**

handle_choice(_) :- writeln("Неверный выбор, попробуйте снова."), menu.

start :-

```
load_database('database.pl'),  
  
menu.
```