



**Русенски Университет „Ангел Кънчев”**

Факултет: Природни науки и образование

Катедра: Информатика и информационни технологии

Специалност: Компютърни науки

Курс: втори

# **КУРСОВА РАБОТА**

**Непроцедурно програмиране**

на Радина Русева, ФН 146529

**Тема №13 - Кинотеатри**

дата: .....

проверил: .....

## Съдържание:

1. [Условие на задачата](#)
2. [Анализ и решение на LISP](#)
3. [Анализ и решение на Prolog](#)

### Задача №13:

Дадена е информация за филмите, представени през последната година на европейска сцена. За всеки филм е дадено: име, режисьор, сценарист, фирма - разпространител.

1. Да се избере и опише подходящо представяне на информацията чрез средствата на Lisp. Да се дефинира функция на Lisp, която по зададено име на сценарист, извежда списък с имената на филмите, които са по негов сценарий. За всеки филм да се извежда и съответния режисьор (или режисьори).

2. Да се избере подходящо представяне на информацията на Prolog. Да се дефинира предикат, който извършва същото действие като функцията на Lisp.

### Анализ и решение на LISP

За решението на задачата ще използвам асоциативни списъци, където ще разгледаме обекта филм със следните свойства: идентификатор, име, режисьор, сценарист, фирма-разпространител.

#### Код:

//помощна функция за четене на символен низ до достигане на символ ']' ;

```
(defun readList (l)
  (LET ((name1))
    (setq name1 (read-delimited-list #\])) (terpri)
    (setq l (append l name1 ))
  )
)
```

//функция за създаване на обект филм

```
(defun makeMovie (id name1 director1 scriptW1 prop1)
  (setf (get ID 'name) name1)
  (setf (get id 'director) director1)
  (setf (get id 'scriptW) scriptW1)
  (setf (get id 'prop) prop1)
  id
)
```

//функция за създаване на списък от филми;

```
(defun makeMovieList (l)
  (LET ((id1 ) (name1) (director1) (scriptW1) (prop1))
    (LOOP
      (princ "Enter movie id/ 0 - for exit")(terpri)
      (setq id1 (read)) (terpri)
      (when (equal id1 0 ) (return l) ) (terpri)
      (princ "Enter movie name") (terpri)
      (setq name1 (readList '())) (terpri)
    )
  )
)
```

```

(princ "Enter movie director name") (terpri)
(setq director1 (readList '() )) (terpri)
(princ "Enter movie script-writer name") (terpri)
(setq scriptW1 (read)) (terpri)
(princ "Enter movie propagator") (terpri)
(setq prop1 (read)) (terpri)
(setq l (append l (list (makeMovie id1 name1 director1 scriptW1 prop1))))
)
)
)
//функцията, която по зададено име на сценарист извежда списък с имената на филмите,
//които са по негов сценарий.
(defun findScrW (list scr)
  (cond ( (null list) '())
        ( (equal (get (car list) 'scriptW) scr)
          (cons (append (list (get (car list) 'name)
                              (get (car list) 'director)
                              )
                  )
                )
          )
        )
    )
  (findScrW (cdr list) scr)))
(t (findScrW (cdr list) scr))
)
)

```

Функцията има два аргумента: списък(от филмите) и символен низ(име на сценарист). Тя решава задачата използвайки рекурсия. Граничното условие е когато списъкът от филми е празен, тогава със сигурност няма да намерим филм на този сценарист. След това взимаме главата на списъка, т.е. първия филм и ако името на сценариста съвпада с даденото име – образуваме списък от името на този филм и режисьора на този филм, този списък съединяваме със рекурсивното извикване на функцията с опашката на списъка. Във всички останали случаи извикваме функцията с опашката на списъка.

### Примерен вход-изход:

```

CL-USER 1 > (setq l (makeMovieList '() ))
Enter movie id/ 0 - for exit
m1
Enter movie name
Pulp Fiction ]
Enter movie director name
Ivan Petrov
Petyr Ivanov ]
Enter movie script-writer name
Pesho
Enter movie propagator
cinemCity
Enter movie id/ 0 - for exit
m2
Enter movie name
F and Furius ]
Enter movie director name
Stoqn Ivanov
Vin Disel ]
Enter movie script-writer name
Pesho

```

```
Enter movie propagator
cinema
Enter movie id/ 0 - for exit
0
(M1 M2)
```

```
CL-USER 2 : 1 >(findScrW I 'Pesho)
(((PULP FICTION) (IVAN PETROV PETYR IVANOV) END) ((F AND FURIUS) (STOQN IVANOV
VIN DIESEL) END))
```

## Анализ и решение на Prolog

За решението на задачата ще използвам структурата *movie* с аргументи: име на филм, име на режисьор, име на сценарист, фирма-разпространител. Имената на сценарист и режисьора също са представени чрез структура *names* (име,фамилия).

### Код:

```
movies([
%movie(movie name,Director name, Screenplay writer names, film
distributor),
    movie("Pulp Fiction",names(quentin,tarantino),
    names(quentin,tarantino), miramax),
    movie("Django unchained",names(quentin,tarantino),
    names(quentin,tarantino),imax),
    movie("The wolf of wall street",names(martin,scorsese),
    names(terence,winter),cinema733),
    movie("Get on Up",names(tate,taylor),names(john-
henry,butterworth),miramax),
    movie("Ray",names(taylor,hackford ),names(james,white),
imax3D),
    movie("V for vendetta",names(james,mcTeigue ),
    names(terence,winter),imax)
]).

fromScrW( _,_, [], []) .%гранично условие
fromScrW(S1,S2,
    [movie(Mn,names(D1,D2),names(S1,S2),A)|T],
    [movie(Mn,names(D1,D2))|NewT]):-fromScrW(S1,S2,T,NewT).
fromScrW(S1,S2,[_|T],NewTt):-fromScrW(S1,S2,T,NewTt).

?-movies(L),
    write("enter Screenplay writer names"), read(S1),read(S2), nl,
    fromScrW(S1,S2,L,R),write(R),nl.
```

### Примерен вход-изход:

```
Compiling the file:
H:\146529\NP\kursova prolog\movieDB.pro
0 errors, 0 warnings.
```

```
enter Screenplay writer names: quentin,tarantino
```

```
[movie("Pulp Fiction",names(quentin,tarantino)),movie("Django  
unchained",names(quentin,tarantino))]  
Yes.
```

```
Compiling the file:  
H:\146529\NP\kursova prolog\movieDB1  
0 errors, 0 warnings.
```

```
enter Screenplay writer names: james,white  
[movie("Get on Up",names(tate,taylor)),movie("Ray",names(taylor,hackford))]  
Yes.
```

```
Saving.
```

```
Compiling the file:  
H:\146529\NP\kursova prolog\movieDB1  
0 errors, 0 warnings.
```

```
enter Screenplay writer names: terAnce,winter  
[]  
Yes.  
No.
```