НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ ТА СПЕЦІАЛІЗОВАНИХ КОМП’ЮТЕРНИХ СИСТЕМ

**Лабораторна робота №7  
з дисципліни «Вступ до функціонального програмування»**

**Варіант 6**

Виконав  
студент 3-го курсу  
групи КВ-41  
Горпинич-Радуженко Іван

Київ – 2017

**Постановка задачі**

Написати програму “Рейтинг фільмів” (назва, режисер, рік виходу на екран, рейтинг). Меню програми повинно включати наступні команди:

*List all % Вивести усі фільми*

*Add new %Додати новий фільм*

*Delete film %Видалити фільм*

*List film by name mask %Вивести список фільмів по шаблону імені*

*Find by name film %Знайти по назві*

*Find by name producer %Знайти по імені режисера*

*Find by rating %Знайти по рейтингу*

**Лістинг**

(defparameter \*db\* nil)

(defun add-new-film (title director year rating)

(push (list :title title :director director :year year :rating rating) \*db\*))

(defun delete-film (title director)

(dolist (film \*db\*)

(when (and (equal title (getf film :title)

(equal director (getf film :director)))

(setq \*db\* (remove film \*db\*))))

\*db\*)

(defun masking (title mask)

(cond

((string= mask "\*")

t)

((or (string= "" title) (string= "" mask))

nil)

((char= (char title 0) (char mask 0))

(masking (subseq title 1) (subseq mask 1)))

((char= #\\* (char mask 0))

(masking title (subseq mask 1)))

(t

(masking (subseq title 1) mask))))

(defun does-mask-fit (title mask)

(if (and (char/= #\\* (char mask 0)) (char/= (char title 0) (char mask 0)))

nil

(masking title mask)))

(defun search-by-mask (mask &aux res)

(dolist (film \*db\*)

(when (does-mask-fit (getf film :title) mask)

(push film res)))

res)

(defun search-by-property (property value &aux res)

(dolist (film \*db\*)

(when (equal (getf film property) value)

(push film res)))

res)

(defun search-by-title (title)

(search-by-property :title title))

(defun search-by-director (director)

(search-by-property :director director))

(defun search-by-rating (rating)

(search-by-property :rating rating))

(defun save-database (filename)

(with-open-file (out filename :direction :output :if-exists :supersede)

(with-standard-io-syntax

(print \*db\* out))))

(defun load-database (filename)

(with-open-file (in filename)

(with-standard-io-syntax

(setf \*db\* (read in)))))

(defun get-answer (answer)

(format \*query-io\* "~a" answer)

(force-output \*query-io\*)

(read-line \*query-io\*))

(defun get-title ()

(get-answer "Title: "))

(defun get-director ()

(get-answer "Director: "))

(defun get-year ()

(get-answer "Year: "))

(defun get-rating ()

(get-answer "Rating: "))

(defun get-mask ()

(get-answer "Mask: "))

(defun print-film-list (film-list &aux (i 0))

(format t "~%")

(if film-list

(dolist (film film-list)

(format t "~a.~%" (setq i (1+ i)))

(format t "Title: ~a~%Director: ~a~%Year: ~a~%IMDB rating: ~a~%~%"

(getf film :title)

(getf film :director)

(getf film :year)

(getf film :rating)))

(format t "Nothing found~%")))

(defun print-all-films ()

(print-film-list (reverse \*db\*)))

(defun show-menu ()

(format t "~%1. Show all films~%2. Add new film~%3. Delete film~%4. Search by title mask~%5. Search by title~%6. Search by director~%7. Search by rating~%0. Quit~%"))

(load-database "C:/Documents/films.db")

(loop named main with choice = 0 do (progn

(when (member choice '(0 1 2 3 4 5 6 7))

(show-menu))

(setq choice (or (parse-integer (get-answer "") :junk-allowed t) 8))

(cond

((= 0 choice)

(save-database "C:/Documents/films.db")

(return-from main))

((= 1 choice)

(print-all-films))

((= 2 choice)

(add-new-film (get-title) (get-director) (get-year) (get-rating)))

((= 3 choice)

(delete-film (get-title) (get-director)))

((= 4 choice)

(print-film-list (search-by-mask (get-mask))))

((= 5 choice)

(print-film-list (search-by-title (get-title))))

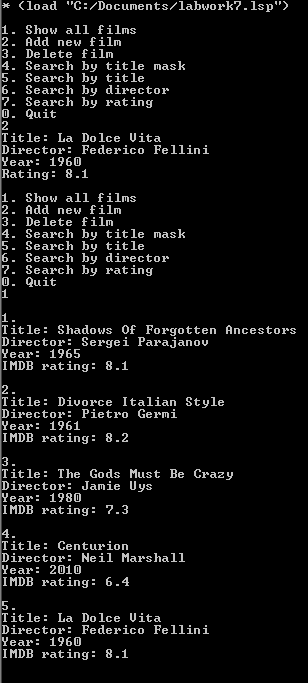
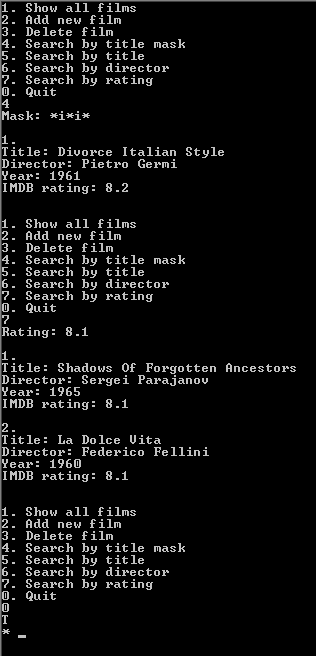
((= 6 choice)

(print-film-list (search-by-director (get-director))))

((= 7 choice)

(print-film-list (search-by-rating (get-rating)))))))

**Приклад роботи програми**

****