



Projeto de Recurso

ATAD – 2019/2020

Docente: Patrícia Macedo

Discente: Iúri Tomé 180221123

Índice

Descrição dos ADTs utilizados	2
Complexidade algorítmica dos comandos	2
LOADM	2
LOADR	2
TOP5.....	2
RATING	2
SAMEDIR	2
VOTERS.....	2
COUNTRY	2
YEAR	2
MATRIX	2
PREDICTION.....	2
Pseudo-Código.....	2
Country	2
Year.....	4
Limitações	6
Conclusão	6

Descrição dos ADTs utilizados

Neste projeto utilizei os ADT List e ADT Map com a estrutura de dados Array List, pois tomei em consideração que não existe uma ligação concreta de entre os variados filmes sendo então mais simples a procura dos mesmos em cada ADT através de índices.

Complexidade algorítmica dos comandos

LOADM

Complexidade $O(n^2)$

LOADR

Complexidade $O(n^2)$

TOP5

Complexidade $O(n^2)$

RATING

Complexidade $O(n^2)$

SAMEDIR

Complexidade $O(n^2)$

VOTERS

Não implementado

COUNTRY

Complexidade $O(n)$

YEAR

Complexidade $O(n^2)$

MATRIX

Complexidade $O(n^2)$

PREDICTION

Não implementado

Pseudo-Código

Country

Input:

list – pointer to list

map – pointer to map

Output:

Void

Implementação:

Country():

Declare string countryName

Declare integers ISize, tempSize, totalProducedMovies as 0

Declare double score as 0.0

Declare ptList temp as listCreate(10)

Declare ptMovie m as createemptyMovie()

Declare ptRating mRat as create EmptyRating()

Print("\nIntroduza o nome do país sobre o qual pretende obter informação: ")

Read(countryName);

For integer i = 0 until i < ISize do

listGet(list, i, m)

if countryName and m.country equal then

listAdd(temp, 0, *m)

listSize(temp,&tempSize)

totalProducedMovies = tempSize;

if tempSize = 0 then

print("\nO país que solicitou não produziu nenhum filme.\n")

else if tempSize > 0 then

for i = 0 until i < tempSize then

listGet(temp, i, m)

mapGet(map, createNewKey(m.id), mRat);

score /= totalProducedMovies;

```
print("Country: "+countryName+"\n")
print("Total Produced movies: "+totalProducedMovies+"\n")
print("Average Score: "+score+"\n")
```

Year

Input:

list – pointer to list

map – pointer to map

Output:

Void

Implementação:

Year()

Declare interger year = 0

Declare action temp as listCreate(5)

Declare animation temp as listCreate(5)

Declare comedy temp as listCreate(5)

Declare drama temp as listCreate(5)

Declare horror temp as listCreate(5)

Declare musical temp as listCreate(5)

Declare musical temp as listCreate(5)

Declare romance temp as listCreate(5)

Declare thriller temp as listCreate(5)

Declare ISize

listSize(list, &ISize)

for integer i = 0 and i < ISize do

declare Movie m

listGet(list, i, &m)

if m.gender is "ACTION" and m.year = year then

```
listAdd(action, 0, m)
```

```
else if m.gender is "ANIMATION" and m.year = year then
```

```
listAdd(animation, 0, m)
```

```
else if m.gender is "COMEDY" and m.year = year then
```

```
listAdd(comedy, 0, m)
```

```
else if m.gender is "DRAMA" and m.year = year then
```

```
listAdd(drama, 0, m)
```

```
else if m.gender is "HORROR" and m.year = year then
```

```
listAdd(horror, 0, m)
```

```
else if m.gender is "MUSICAL" and m.year = year then
```

```
listAdd(musical, 0, m)
```

```
else if m.gender is "ROMANCE" and m.year = year then
```

```
listAdd(romance, 0, m)
```

```
else if m.gender is "THRILLER" and m.year = year then
```

```
listAdd(thriller, 0, m)
```

```
declare array categoryNames as {"ACTION", "ANIMATION", "COMEDY", "DRAMA",  
"MUSICAL", "HORROR", "ROMANCE", "THRILLER"}
```

```
declare categoryLists as {action, animation, comedy, drama, horror, musical,  
romance , thriller}
```

```
declare integer categoryListsSize as 8
```

```
declare double score as 0.0
```

```
declare integer totalProducedMovies as 0
```

```
print(year)
```

```
for i as 0 and until i < categoryListsSize do
```

```
score = 0.0
```

```
totalProducedMovies = 0
```

```
listSize(categoryList[i], &totalProducedMovies)
```

```
for integer j = 0 until j < totalProducedMovies do
```

```
declare PtMovie m = createEmptyMovie()
declare PtRating r = createEmptyRating()
listGet(categoryLists[i], j , m)
mapGet(map, createNewKey(m.id),r)
score += r.score

score/=totalProducedMovies

score = floor(score*10)/10;
print(categoryListsNames[i])
print("Average Score"+score)
print("Total Produced Movies"+totalProducedMovies)

listDestroy(&action)
listDestroy(&animation)
listDestroy(&comedy)
listDestroy(&drama)
listDestroy(&horror)
listDestroy(&musical)
listDestroy(&romance)
listDestroy(&thriller)
```

Limitações

Os comandos não implementados são: "Voters" e "Predictions". Quanto a problemas no código o problema que mais prevalece ao longo do projeto é a gestão de memória.

Conclusão

Neste projeto uso a ADT's List, para a lista de filmes, e Map para a coleção de avaliações de cada filme, ambas implementadas com uma estrutura de dados (ED) Array List, como referido anteriormente. Sendo então a utilização dos mesmos o grande foco deste projeto. Concluindo ao realizar este projeto a compreensão de cada um destes ADT's tornou-se mais clara, devido à prática.