UBA – Facultad de Ciencias Exactas y Naturales – Departamento de Computación

Algoritmos y Estructuras de Datos I

Segundo cuatrimestre de 2013

27 de septiembre de 2013

TPF - Cines

1. Introducción

En el TPE, hemos especificado el comportamiento de una serie de problemas relacionados con el mundo de los cines. Nuestro próximo paso será obtener un programa funcional que respete la especificación que adjunta la cátedra para esta nueva etapa del proyecto. A continuación diremos cómo decidimos implementar cada tipo y qué funciones exporta (interfaz). La interfaz de los tipos no podrá ser modificada, salvo por expresa (expresa==escrito) indicación de la cátedra. En cada tipo pueden implementar su propio `show`, de tal manera que se sientan cómodos con su visualización. **NO** pueden agregar el `Eq` si el mismo no se encuentra en el presente enunciado. Se pueden agregar funciones auxiliares dentro de cada módulo, pero **NO** se pueden exportar.

Para la resolución del trabajo, se pueden usar únicamente las funciones y operadores `last`, `init`, `head`, `tail`, `!!`, `reverse`, `++`, `elem`, `length`, `fst`, `snd` y los operadores de comparación entre elementos de un mismo tipo. Notar que la especificación que se adjunta **NO** es una solución al tp de especificación.

Los tipos **NO** pueden ser transformados en listas para la implementación de los problemas.

Su trabajo consistirá en implementar todos los problemas especificados por la cátedra (ver archivo aparte). A continuación se detalla la definición de los módulos que deben implementar.

2. Módulo Tipos

```
module Tipos where

type Sala = Int
type Nombre = String
type Actor = String

data Genero = Aventura | Comedia | Drama | Romantica | Terror deriving (Show, Eq)
```

3. Módulo Película

```
module Pelicula (Pelicula, nuevaP, nombreP, generosP, actoresP, es3DP, agruparPelisPorGeneroP, generarSagaDeP

import Tipos

data Pelicula = P Nombre [Genero] [Actor] Bool deriving (Show, Eq)
```

donde el constructor `P` recibe los parámetros en el siguiente orden:

- Nombre de la película.
- Lista de géneros de la película.
- Lista de actores de la película.
- Un Bool que indica si la película es 3D.

4. Módulo Ticket

```
module Ticket (Ticket, nuevoT, salaT, peliculaT, usadoT, usarT, peliculaMenosVistaT, todosLosTicketsParaLaMis

import Tipos
import Pelicula
```

```
data Ticket = TicketSinUsar Sala Pelicula | TicketUsado Ticket deriving (Show, Eq)
```

donde el constructor `TicketSinUsar` indica que el ticket está sin usar, mientras que el constructor `TicketUsado` indica que el ticket fue usado. El primer contructor recibe los parámetros en el siguiente orden:

- Sala del ticket.
- Película del ticket.

El segundo constructor recibe un único parámetro que es el ticket sin usar.

5. Módulo Cine

```
module Cine (nuevoC, nombreC, peliculasC, salasC, espectadoresC, salaC, ticketsVendidosC,  abrirSalaC, agrega

import Tipos
import Pelicula
import Ticket

data Cine = C Nombre |
SalaSinPelicula Cine Sala |
SalaConPelicula Cine Sala Pelicula Int |
TicketVendido Cine Ticket deriving (Show)
```

donde el contructor `C` recibe el nombre de un Cine sin salas ni películas. El constructor `SalaSinPelicula` agrega una sala sin película al cine que recibe como parámetro. El constructor `SalaConPelicula` agrega una sala con una película asociada, el útimo parámetro del constructor indica la cantidad de espectadores que ingresaron a la sala. El constructor `TicketVendido` agrega un ticket vendido al cine que recibe como parámetro.