

Condiciones de aprobación

Para aprobar es necesario simultáneamente:

- completar el 60% del examen, y
- obtener al menos la mitad de los puntos en cada paradigma.

En todas tus respuestas sé puntual, no pierdas el foco de lo que se pregunta. Respuestas en exceso generales son tan malas como respuestas incompletas.



PARTE A - Lógico

Dada la siguiente base de conocimiento que modela las visitas de gente a atracciones de un parque de diversiones, y si disfrutaron de las mismas:

```
subioA(prudencio, montaniaRusa(hulk)).
subioA(prudencio, autosChocadores).
subioA(hermenegildo, montaniaRusa(rockit)).
subioA(hermenegildo, simulador(minions)).
subioA(hermenegildo, autosChocadores).
subioA(rigoberta, simulador(ikran)).
subioA(brunilda, simulador(simpsons)).
```

```
disfruto(prudencio, montaniaRusa(hulk)).
disfruto(prudencio, autosChocadores).
disfruto(hermenegildo, simulador(minions)).
disfruto(rigoberta, simulador(ikran)).
foo(Persona):- forall(subioA(Persona, Atraccion), disfruto(Persona, Atraccion)).
```

Se pide:

1. ¿Cuál es el objetivo del predicado foo/1? Elegir un nuevo nombre que sea más expresivo y explicar para qué personas debería ser cierto dada la base de conocimientos actual.
2. Analizar la inversibilidad del predicado foo/1 y, en caso de que no sea inversible, modificar todo lo que sea necesario para que lo sea.
3. Decimos que una persona es valiente cuando subió a una montaña rusa, sin importar si fue la de hulk o la de rockit. ¿Cómo se implementaría esa definición?

PARTE B - Funcional

Dado el siguiente fragmento de código en Haskell:

```
data Cafe =
```

```
  Cafe { tipo :: String, tamaño :: String, calorías :: Double, precio :: Double } deriving(Show)
```

```
type Agregado = Cafe -> Cafe
```

```
aumentarPrecio monto cafe = cafe{ precio = precio cafe + monto }
```

```
modificarCalorias modificacion cafe = cafe{ calorías = (modificacion . calorías) cafe }
```

```
almendras :: Agregado
```

```
almendras = ... implementar ...
```

```
especialDulceDeLeche :: Agregado
```

```
especialDulceDeLeche = ... implementar ...
```

Se pide:

1. Indicar los tipos de las funciones aumentarPrecio y modificarCalorias.
2. Implementar los agregados de almendras y especialDulceDeLeche usando composición y aplicación parcial de funciones de forma tal que:

- a. Las almendras aumenten en 80 las calorías del café y su precio sea 150 más caro.
 - b. El especial dulce de leche triplice las calorías del café y su precio sea 350 más caro.
3. Dada una lista infinita de agregados:
- a. ¿Podríamos conocer el estado final de un café luego de realizar todos los agregados?
 - b. ¿Sería posible conocer cómo quedaría el café luego de realizar cada agregado individualmente de las otras si sólo nos interesan los primeros resultados?

Justificar conceptualmente cada respuesta.

PARTE C - Objetos

Se cuenta con información de los restaurantes: se conocen los platos de su menú, y para cada uno de los platos se sabe el origen, chef que lo cocina, la cantidad de calorías y si contiene gluten.

También hay críticos de restaurantes que conforman un jurado. Cada crítico evalúa los restaurantes teniendo en cuenta su preferencia. Algunos de los críticos son los siguientes:

- Anacleto. Prefiere restaurantes con algo de comida de origen chino.
- Eustaquia. Prefiere los restaurantes con variedad, entendiendo esta como más de 50 platos en su menú.
- Margarito. Prefiere los restaurantes "light", que son aquellos que tienen sólo platos de menos de 300 calorías.

Se cuenta con la clase Restaurante, que ya tiene definidos los atributos necesarios para representar la información indicada. También hay una clase Chef con su nombre y si ganó algún premio. Estas clases pueden extenderse en los puntos posteriores, si se considera necesario. Asumir que hay métodos de acceso para todos los atributos que pueden usarse en la solución.

1. Necesitamos implementar un método **elegidosDelJurado** que resuelve el requerimiento principal de determinar qué restaurantes de un festival cumplen con las preferencias de todos los críticos del jurado.
 - a. Representar a los críticos con sus preferencias de modo que puedan formar parte del jurado.
 - b. Implementar **elegidosDelJurado** justificando la decisión sobre quién es el receptor del mensaje y qué parámetros recibe.
2. Necesitamos agregar a Rosenda, una crítica que prefiere los restaurantes con variedad y también los que tienen comida peruana. ¿Sería posible incorporar a Rosenda sin cambiar lo anteriormente desarrollado? Justificar y en caso de que la solución no lo permita, explicar el motivo.
3. Se quiere poder representar un jurado exigente, que además de preferir los restaurantes que cumplen con las preferencias de todos los críticos que lo componen, los restaurantes elegidos deben tener algún chef que haya sido premiado.
 - a. Realizar los cambios necesarios para resolver este problema.
 - b. Explicar qué conceptos del paradigma se aplicaron para resolverlo y qué beneficios aportan.

7 (siete)

A	B	C
1,5	2,5	3,5

Condiciones de aprobación

Para aprobar es necesario simultáneamente:

- completar el 60% del examen, y
- obtener al menos la mitad de los puntos en cada paradigma.

En todas tus respuestas sé puntual, no pierdas el foco de lo que se pregunta. Respuestas en exceso generales son tan malas como respuestas incompletas.

Nombre y apellido: Francisco Massera

Parte A

Dadas las siguientes definiciones

tomar:: Bebida -> Persona -> Persona
Devuelve a la persona luego de haberse tomado la bebida

mezclar:: Bebida -> Bebida -> Bebida
Obtiene la nueva bebida resultante de mezclar las dos bebidas dadas

1. Completar la definición de una función que permita obtener cómo queda un conjunto de personas luego de que cada una se tome un "cóctel" producto de mezclar varias bebidas.

seTomanUnCoctelConEstasBebidas bebidas personas =

map (tomar . (bebidas . fold (bebida , mezclar))) personas

Indicar qué concepto/s se están utilizando en la solución anterior:

☒ Orden Superior | ☐ Listas infinitas | ☒ Aplicación Parcial | ☐ Efecto

Parte B

Asumiendo que en la base de conocimiento hay numerosos hechos del siguiente predicado:
tiene(Persona, Cosa, Cantidad).

por ejemplo:

tiene(tito, arboles, 5).

tiene(tito, problemas, 2).

predicadoMisterioso(Persona , Cosa , CantidadM) :-

tiene(Persona , Cosa , CantidadM),

forall(tiene(Persona , Cosa , Cantidad), CantidadM >= Cantidad).

1. Completar los ... de la solución propuesta del predicadoMisterioso,
2. hacer las consultas adecuadas
3. y, para ganar expresividad, ponerle mejor/es nombre/s a predicadoMisterioso, para:

- a. permitir encontrar cuál es la cosa que tito tiene en mayor cantidad:
- b. permitir averiguar la cantidad máxima que cualquier persona tenga de una cosa dada:

class Radio (

Tomar como no está marcada