

## FINAL DE PARADIGMAS DE PROGRAMACION

Legajo.: \_\_\_\_\_

Alumno: \_\_\_\_\_

### Ejercicio 1

A partir del siguiente fragmento de programa Haskell

```
doble x = x + x
```

y de estas definiciones de las funciones head y map

```
head (h:t) = h  
map f [] = []  
map f (h:t) = (f h) : map f t
```

se evalúa esta expresión

```
head (map doble [1..])
```

**Indique sobre la expresión evaluada:**

- ¿cuál es la función de orden superior que aparece en la función evaluada? ¿cómo la reconoce?
- Si se considera la expresión `map doble [1..]` ¿Termina la evaluación de esta función?. Explique y justifique el porque de su respuesta.
- Si se considera la expresión `head (map doble [1..])` ¿Termina la evaluación de esta función?. ¿Con qué concepto está relacionado?. Justifique, indicando la respuesta que se obtiene.
- Especifique el dominio e imagen de la función `map`.

---

### Ejercicio 2

A partir del siguiente programa Prolog

```
padre(homerο,bart).  
padre(homerο,lisa).  
hermano(X,Y):- padre(Z,X), padre(Z,Y), X \= Y.
```

**Indique**

- El predicado `hermano/2`, ¿Es inversible? Exponga ejemplos que justifiquen su respuesta.
- ¿Cuál va a ser la respuesta de Prolog si se carga el programa anterior y se hace esta consulta?

```
?- hermano(nico,mariano).
```

Justifique relacionando con algún concepto visto en la materia.

- Agregamos ahora estas cláusulas al programa:

```
madre(marge,bart).  
madre(marge,lisa).
```

y queremos modificar el programa anterior, para que el predicado `hermano` relacione dos personas cuando tengan el mismo padre y la misma madre, y agregando el predicado `hermanastro` que relacione los pares de personas que compartan o bien su padre o bien su madre.

Indique qué tienen de similar y qué tienen de distinto los predicados `hermano` y `hermanastro` así definidos, implementando ambos predicados y mostrando cómo se nota la diferencia en la implementación.

## FINAL DE PARADIGMAS DE PROGRAMACION

### Ejercicio 3

Se nos pide construir una aplicación que maneje algunos eventos que ocurren en un hipódromo.

Uno de ellos son las apuestas; cada apuesta se refiere a una carrera.

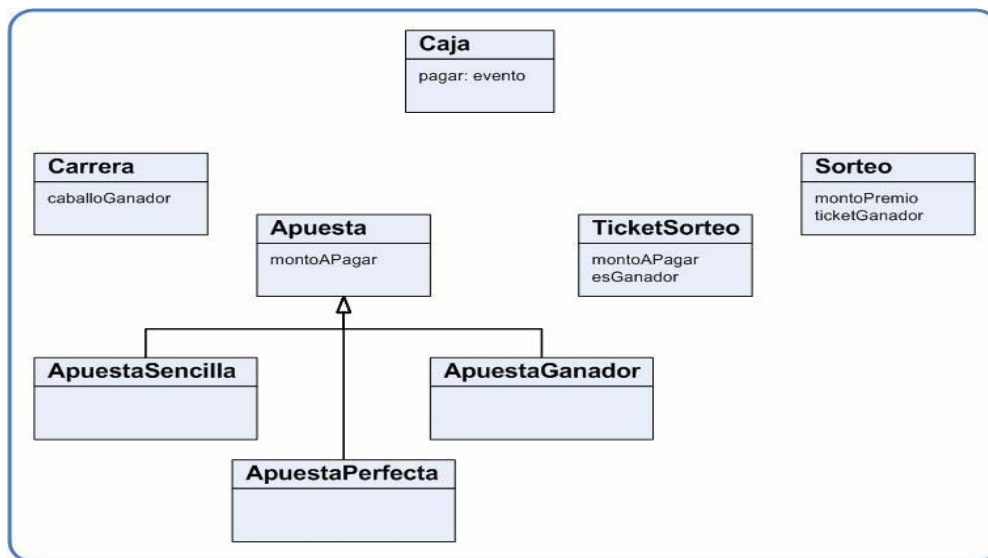
Hay tres tipos de apuesta:

- A ganador.
- Perfecta, en la que el apostador indica qué caballo saldrá primero, qué caballo saldrá segundo y qué caballo saldrá tercero.
- Sencilla, en la que el apostador indica un solo caballo, y gana si el caballo sale primero, segundo o tercero.

También se realizan sorteos, a las personas que realizan consumos en los buffets del hipódromo se les entregan tickets para estos sorteos.

Por la caja se pagan apuestas como tickets.

Se construyen las clases indicadas en este Diagrama Jerárquico de Clases



El código del método pagar: en la clase Caja es el siguiente

```
pagar: evento
"desembolsa el pago correspondiente al evento"
(evento esGanador)
if True: [self desembolsar: evento montoAPagar]
```

El monto a pagar de una apuesta es resultado de un cálculo que tiene que ver con la proporción entre total de apuestas y apuestas ganadoras en una carrera, la distribución entre tipos de apuesta, y el porcentaje que el hipódromo decida quedarse como ganancia. Este cálculo es el mismo para todos los tipos de apuesta.

**A partir de este enunciado:**

- Indicar en el diagrama en qué clase/s falta implementar el método *esGanador*.
- Con este agregado, indicar la presencia de polimorfismo. Explicar el lugar dónde se lo aprovecha y los beneficios que implica.

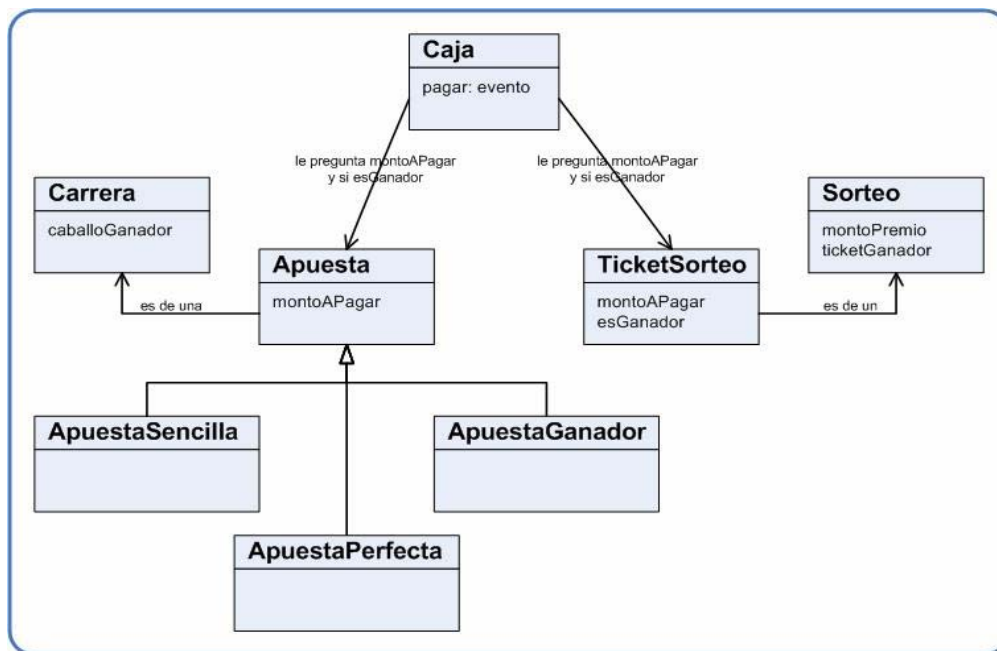
## FINAL DE PARADIGMAS DE PROGRAMACION

- c. Supongamos que el hipódromo quiere estimular las apuestas perfectas, y lo hace aumentando en un 25% el monto a pagar por ellas. Implementar el método que hace falta para lograr este aumento, indicando qué concepto/s de orientación a objetos se está/n utilizando.
- d. Ahora se nos pide saber el monto total pagado en premios. Indicar a cuál de los objetos del diagrama se asignaría esta responsabilidad.

### Aclaración

*Solo a efectos de la mejor comprensión del modelo jerárquico representado, se muestra el siguiente gráfico que establece todas las relaciones para comprender mejor las relaciones que se producen entre las diferentes clases.*

*Por ello solo debe tenerse en cuenta en caso de que simplifique la comprensión del modelo planteado.*



---

### Ejercicio 4

Describir brevemente el concepto de variable en los paradigmas lógico, de objetos y estructurado imperativo. Indicar una diferencia fundamental en el manejo de variables entre lógico y los otros, ¿con qué concepto está relacionada esta diferencia?

---

Respuestas correctas:

Ejercicio 1

- a. map, porque recibe una función como argumento.
- b. porque la expresión incluye la lista infinita [1..].
- c. por la evaluación diferida / evaluación tardía / lazy evaluation, que permite obtener el primer elemento sin necesidad de calcular el map para toda la lista.
- d. map :: (a-> b) -> [a] -> [b]

Ejercicio 2

- a. sí, porque es posible hacer cualquiera de estas consultas:
  - hermano(bart,lisa). % respuesta sí / yes
  - hermano(X,lisa). % respuesta X = bart
  - hermano(lisa,X). % respuesta X = bart
  - hermano(X,Y). % respuestas X = bart, Y = lisa y al revés
- b. la respuesta va a ser no.  
El concepto relacionado es mundo cerrado / (hipótesis de) universo cerrado / (suposición de) mundo acotado.
- c. Lo que me interesa es que entiendan la diferencia entre “y” y “o”.  
O sea que digan “son las mismas condiciones, en hermano es un ‘y’ mientras que en hermanastro es un ‘o’”.  
... o si quieren, que lo implementen, pero que describan en sus palabras similitudes y diferencias entre los predicados implementados.

Ejercicio 3

Respuestas correctas:

- a. ApuestaSencilla, ApuestaPerfecta y ApuestaGanador
- b. los objetos polimórficos son las instancias de ApuestaSencilla, ApuestaPerfecta, ApuestaGanador y TicketSorteo; son polimórficos para / el que usa este polimorfismo es la Caja.
- c. el método es
  - #ApuestaPerfecta
  - montoAPagar
  - ^super montoAPagar \* 1.25el concepto es redefinición
- d. si quiero saber el monto total pagado lo más razonable es preguntarle a la Caja, que maneja los pagos.

Respuesta correcta:

lógico: lo que se liga en la unificación / lo que se reemplaza por los valores / algo similar al concepto matemático de variable.

objetos: referencia a un objeto

estructurado: nombre simbólico de una dirección de memoria.

Diferencia: en lógico no puedo asignar/asignar explícitamente/reasignar una variable, en los otros sí. Está relacionada con el concepto de efecto de lado.

## FINAL DE PARADIGMAS DE PROGRAMACION

Criterio de aprobación

-----

Propongo que apruebe un alumno que responde al menos:

- dos de los tres ítems de funcional
  - uno de los dos ítems de lógico
  - dos de los ítems de objetos
  - algo razonable en la última pregunta, que demuestre idea de efecto de lado y/o del concepto de variable en objetos y estructurado.
- entonces está para aprobar.

También permitiría aprobar a un alumno al que le falta totalmente un tema, si uno de los otros está cerca de la perfección, o dos bastante bien.

Ya si faltan dos de las cuatro preguntas creo que no está para aprobar, lo evaluaría si las otras dos están muuuuy cerca de la perfección.

Tal vez sí podría aprobar si una de las que está perfecta es la 3, dado que el tema de objetos se lleva más peso en la materia.