



Instituto Politécnico Nacional

INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo



Paradigmas de programación

Prof.: Roberto Tecla Parra

Alumno: Arturo Monroy Mora

Grupo: 3CM3

Guía 3er Parcial



1. Denota las características esenciales que distinguen a un objeto de otra clase de objetos y provee así una frontera conceptual definida, relativa a la perspectiva del observador

R. D) Abstracción

2. Engloba todas las (usualmente estáticas) propiedades del objeto más los valores actuales (usualmente dinámicos) de cada una de sus propiedades.

R. A) Estado de un objeto

3. Es cómo actúa y reacciona un objeto en términos de un cambio de estado.

R. B) Comportamiento de un objeto.

4. Es la propiedad de un objeto que lo distingue de todos los otros objetos.

R. C) Identidad de un objeto.

5. Es una colección de objetos que comparten estructura y comportamiento.

R. B) Clase.

6. Es la utilización de interfaces idénticas con diferentes implementaciones.

R. A) Método.

7. En POO los datos deben ir junto al código que los manipula.

R. Verdadero.

8. En POO los datos no deben ir junto al código que los manipula.

R. Falso.

9. La representación de un TAD solo debe ser manipulada por las operaciones definidas para ese TAD.

R. Verdadero.

10. El encapsulamiento sirve para ocultar la complejidad.

R. Falso.

11. El encapsulamiento sirve para proteger los datos.

R. Verdadero.



12. ¿Cuál es el tipo de `>=>`?

R. IV) `m a -> (a -> m b) -> m b.`

13. ¿Cuál es el tipo de `fail`?

R. String -> m a

14. `Just x >=> f = f x` es la función `>=>` para...

R. B) los Maybe.

15. `Fail _ = Nothing` es la función `fail` para...

R. B) los Maybe.

16. Para no usar el Lambda de abajo

`Just 1000 >=> (\a -> Just (a < 1500))`

escriba una función que use la notación `do`.

R:

```
menor :: Maybe Bool
menor = do
  a <- Just 1000
  Just (a<1500)
```

17. Para no usar el Lambda de abajo

`Just 433 >=> (\a -> Just "?" >=> (\b -> Just (show a ++ "_" ++ b)))`

escriba una función que use la notación `do`.

R:

```
hacer :: Maybe String
hacer = do
  a <- Just 433
  b <- Just "?"
  Just (show a ++ "_" ++ b)
```

18. `Xs >=> f = concat (map f xs)` es la función `>=>` para

R. A) las listas.

19. Para no usar los lambdas de abajo

`[1,2] >>= \n -> ['a','b'] >>= \ch -> return (n,ch)`
escriba una función que use la notación do

R:

```
hacer :: [(Int,Char)]
hacer = do
  n <- [1,2]
  ch <- ['a','b']
  return (n,ch)
```

20. Codificar una función que use la notación do para calcular el producto cartesiano de 3 conjuntos

R:

```
hacer :: [(Int,Int,Int)]
hacer = do
  a <- [1,2,3]
  b <- [1,3,5]
  c <- [2,4,6]
  return (a,b,c)
```