# ¿Qué significa que un programa cumple con su contrato?

Que un programa cumple con su contrato significa que el programa implementado cumple con la especificación del problema a solucionar.

# ¿Qué relación existe entre un problema, una especificación, un algoritmo y un programa?

La relación que existe es que la especificación es la descripción del problema, el algoritmo es la descripción de la solución al problema y el programa es la implementación de la solución al problema.

# ¿Qué es la transparencia referencial? ¿Por qué existe en el Paradigma de Programación Funcional, pero no en el Paradigma de Programación Imperativo?

La trasparencia referencial es la propiedad que garantiza que el valor de una expresión depende exclusivamente de sus subexpresiones, por lo tanto, al ejecutar una función, su valor prevalece en cualquier parte del programa.

Esto no es así en la programación imperativa porque los valores de los argumentos están asociados a un espacio de memoria. Por ejemplo si definimos

Def doble (x:int) -> int:

x =x + x

return x

Si (x = 1) y ejecutamos la función al salir (x = 2), al ejecutar una vez mas (x = 4). En haskell esto no ocurre por la transparencia referencial, doble(1) = 2 para todas las veces que lo ejecutemos.

# ¿Puede existir una función recursiva sin un caso base? ¿Por qué?

No porque si no esta definido el caso base pueden suceder dos cosas, o la función se indefine o el paso recursivo se repite hasta el infinito, lo cual no es producente.

# ¿Qué tipos de instrucciones de control encargadas de repetir un bloque de código vimos en Imperativo?

Las estructuras de control vistas fueron, while, do while y for.

While y do while funcionan de manera similar, dada una condición de finalización hasta que esta no se cumpla un bloque de código se ejecuta. La diferencia está en que en while primero se define la condición y luego el bloque de ejecución, mientras que do while es al revés.

For funciona de manera similar al while, en la condición de finalización se declara de manera local una variable que puede tomar valores una lista, cuando esta recorre toda la lista el ciclo de ejecución finaliza.

# ¿Cuál es la diferencia entre el pasaje de parámetros por copia y por referencia?

La diferencia es que al pasar un parámetro como copia se crea una copia local (la información de la variable se almacena en otro espacio de memoria) del parámetro pasado para que no sea modificada por la función, mientras que por referencia se utiliza directamente la variable (se usa la misma dirección de memoria).

# Defina un Test Suite que tenga el cubrimiento de sentencias de al menos el 70%. Y otro que cubra el 100% de las ramas.

Como la función consta de verificar que no hay repetidos en las filas o columnas tenemos los siguientes casos.

|  |  |
| --- | --- |
| Filas | Columnas |
| Correcta | Correcta |
| Correcta | Incorrecta |
| Incorrecta | Correcta |
| Incorrecta | Incorrecta |

Formar casos de test con estos bastaría para cubrir el 100% de las sentencias y las ramas.

