

Auxiliar 12: Precontrol 3

Todos los problemas deben ser resueltos en *Python*, utilizando estrictamente la Receta de Diseño entregada a lo largo del curso. Use nombres apropiados para funciones y variables, y testee cada vez que sea posible.

1. Listas

La clase Dado contiene los métodos usados en las siguientes instrucciones:

- `D = Dado(L)` # crea dado D de L lados con un valor inicial 0 (que indica que el dado aún no ha sido lanzado)
- `D.lanzar()` # lanza dado D de manera que el dado toma un valor al azar entre 1 y L
- `D.valor()` # entrega valor del dado D

Al respecto:

1. (1 pto) Escriba una función que se invoque en la forma `lanzarDados(N,L)` que entregue una lista de N dados de L lados, cada uno de los cuales debe ser creado y lanzado una vez. Por ejemplo, `lanzarDados(2,6)` entrega una lista con dos objetos de la clase Dado (cada uno con un valor al azar entre 1 y 6).
2. B)(1pto) Escriba una función de nombre `valoresDados` que reciba una lista de dados (lista de objetos de la clase Dado) y entregue una lista de números con los valores de los dados (ordenados de mayor a menor). Por ejemplo, si la lista que se recibe como parámetro contiene dados con valores 3, 2 y 5 entonces la función debe entregar la lista `[5, 3, 2]`.
3. C)(1.5 ptos) Escriba un programa que use las funciones `lanzarDados` y `valoresDados` para simular el juego entre dos jugadores A y B cada uno de los cuales lanza 3 dados de 6 lados y escriba las líneas indicadas en el siguiente ejemplo: valores de los dados del jugador A: `[5, 3, 2]` valores de los dados del jugador B: `[5, 4, 1]` gana B Notas. -El ganador es B porque la lista `[5,4,1]` es mayor que la lista `[5,3,2]`. En Python `[5,4,1]>[5,3,2]` entrega True. -El resultado puede ser un empate.

2. Preugnta 2

La clase Bolsa está definida con las siguientes operaciones

- `B=Bolsa()` # Crear Bolsa inicialmente sin valores
- `B.poner(x)` # Poner objeto x en la bolsa (si no estaba)
- `X = B.sacar()` # Sacar y entregar un objeto cualquiera de la bolsa (se elige al azar entre los elementos existentes). Entrega None si no hay elementos.

1. Siguiendo la receta de diseño, escriba la clase Bolsa suponiendo que un objeto de la clase Bolsa se representa por una lista.
2. Pruebe la clase Bolsa en un programa que genere 20 números al azar entre 1 y 90, los ponga en la bolsa y los saque y muestre todos.