

Enunciado_Auxiliar_11_Objeto_y_Clases

December 10, 2020

1 Auxiliar 11 - Objetos y Clases

1.1 P1 Amigo secreto

Creemos las siguientes clases que nos permitirán modelar el juego amigo secreto.

1.1.1 A)

Defina la clase **Persona**. Esta debe tener los campos nombre y regala_a. El primero será un string y el segundo una Persona.

1.1.2 B)

Defina la clase **Regalo**. Esta debe tener los campos nombre y destinatario. El primero será un string y el segundo una Persona.

1.1.3 C)

Defina la clase **Juego**. Esta tendrá los campos participantes y regalos. El primero será una lista de personas y el segundo una lista de regalos.

1.1.4 D)

Defina el método sorteo para la clase **Juego**. Este debe devolver una lista que represente a quién le entregará un regalo la persona que tiene la misma posición en la lista participantes.

Utilice la función **shuffle del módulo random** de python en su implementación. Además, recuerde que una persona no puede ser amiga secreta de si misma y que todos deben recibir un regalo.

1.1.5 E)

Defina el método **asignarAmigos** para la clase **Juego**. Esta función recibe una lista de personas, que representa a quienes se les hará un regalo a partir de su posición en la lista, es decir, la persona con índice *i* recibirá un regalo de la persona en la lista de participantes que tenga la posición *i*.

El método deberá setear el valor regala_a para todas las personas de la lista de participantes. Además, para el caso en que las listas tienen un tamaño diferente se debe imprimir un error.

1.1.6 F)

Defina el método **entregaRegalos** para la clase Juego. Este debe mostrar en pantalla a todos los participantes, con el regalo que hicieron y a quién se lo entregaron.

1.2 P2. Polinomios

Un polinomio de grado n se puede representar como una lista de $n+1$ coeficientes. ### A) Diseñe una clase **Pol** que permita ser utilizada de la siguiente manera:

Pol([a]) crea el polinomio a , *Pol([b, a])* crea $ax+b$, *Pol([c, b, a])* crea ax^2+bx+c y así sucesivamente.

1.2.1 B)

Defina la función **p.evaluar(x)**, la cual evalúa el argumento x en el polinomio p .

1.2.2 C)

Defina la función **p.toString()**, la que devuelve un string que representa al polinomio p .

1.2.3 D)

Defina la función **p1.sumar(p2)**, la cual recibe 2 polinomios y devuelve un polinomio correspondiente a la suma de estos.

Para ello, cree y luego utilice la función auxiliar **max_lista**, la cual reciba 2 listas y devuelva la lista que tenga el mayor largo, en caso, de que ambas tengan el mismo largo devuelva cualquiera.