



Seminario de Lenguajes
Opción Python
Práctica 4

Parte I

Excepciones

- 1.- ¿Qué es una excepción? Ejemplifique. ¿Por qué es útil que un lenguaje de programación tenga soporte para manejo de excepciones?
 - 2.- ¿Qué acción realiza Python una vez que levanta y maneja una excepción? Si no encuentra un manejador en su bloque try except, ¿dónde lo busca? ¿Y si no existe, qué sucede? Escriba un código que lo demuestre.
 - 3.- ¿En qué se diferencian las cláusulas else y finally del bloque try except? ¿Son obligatorias? De un ejemplo de uso.
 - 4.- Sea una lista de N elementos realice una función que dada una posición ingresada por el teclado, imprima su elemento correspondiente en la lista. Maneje con excepciones si la posición no existe, informándolo y requiriendo que ingrese nuevamente el valor.
 - 5.- Maneje con excepciones el ejercicio 2 de la Práctica 3 para el caso en que los archivos pasados como argumento no existan.
 - 6.- Reescriba el módulo de colas FIFO de la Práctica 2 con manejo de excepciones para los casos que considere necesario (por ejemplo que levante una excepción cuando se desee quitar un elemento de la cola cuando esté vacía).
-

Parte II

Programación Orientada a Objetos (POO)

- 7.- ¿Qué son los objetos? ¿Cómo se comunican entre ellos?

8.- ¿Qué son las clases? ¿Cómo se definen en Python? ¿Cómo se inicializan los objetos de una clase?

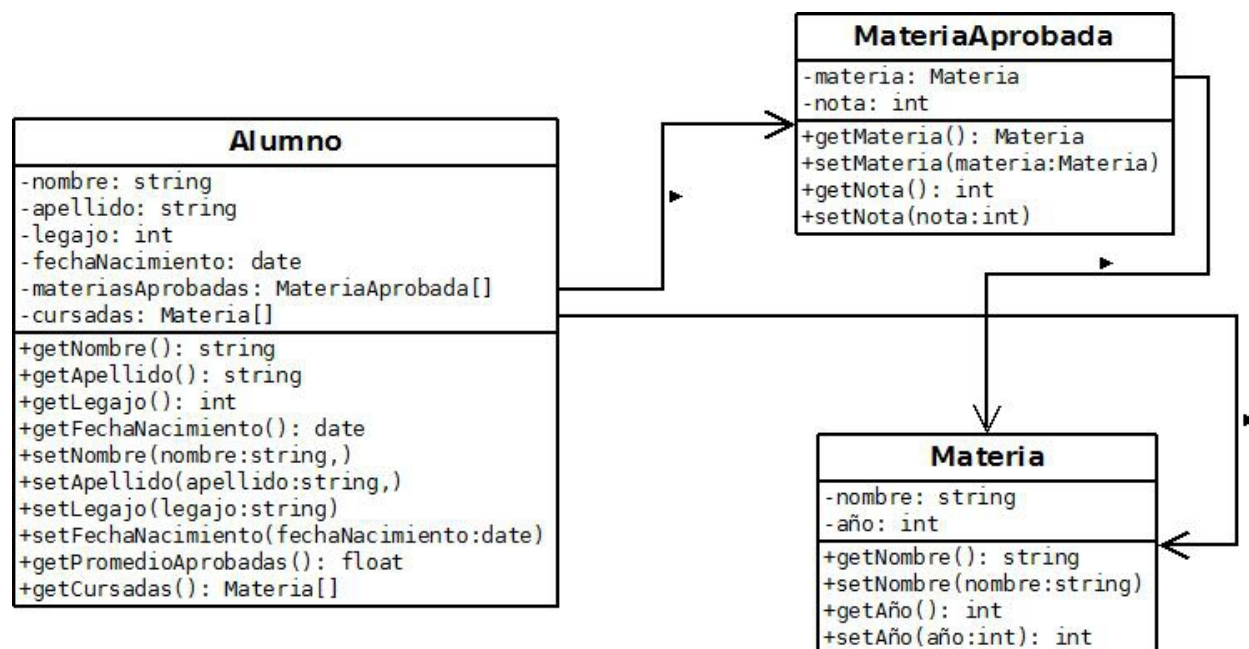
9.- ¿Cuáles son los tres conceptos básicos de la POO? ¿Qué tipo de herencia utiliza Python? ¿Soporta “sobrecarga” de operadores? ¿Cómo se definen los métodos y variables públicos y privados?

10.- Implemente una clase Punto que represente un punto cartesiano (x, y) y que posea un método distancia() que reciba como parámetro otro punto y calcule la distancia entre ambos. (Sean $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$ la distancia entre ambos es igual a $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$)

11.- A la clase anterior agréguele un método que verifique la igualdad entre dos puntos. ¿Es lo mismo preguntar si dos objetos son iguales con el signo == que preguntándolo con el método implementado? ¿Por qué?

12.- Modelice con objetos a los alumnos y materias de la facultad. Un alumno debe tener como información mínima el nombre y apellido, número de legajo y fecha de nacimiento. Implemente un método para asignarle materias que esté cursando el alumno y aquellas que aprobó junto con su nota. Implemente un método que calcule el promedio de todas las cursadas que aprobó un alumno y luego realice un listado de aquellos alumnos que estén cursando al menos 2 materias.

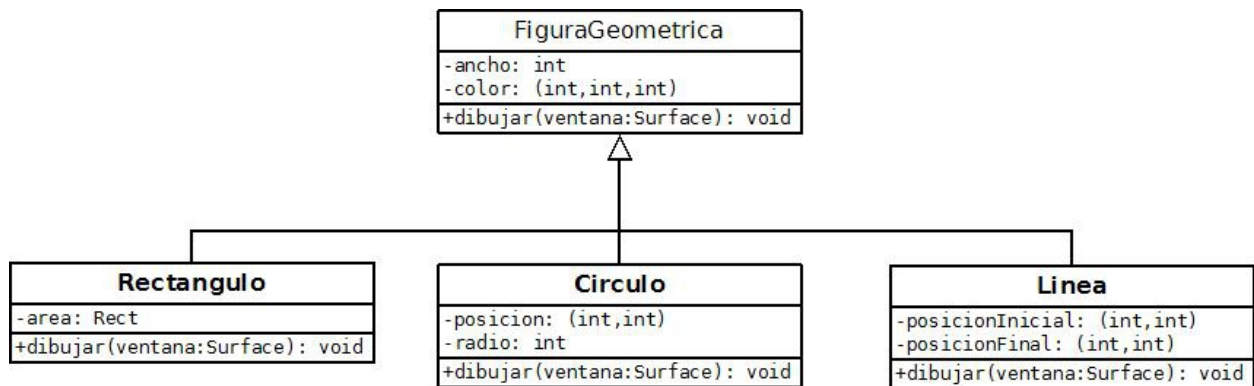
A continuación se verá el diagrama UML asociado:



Parte III

Introduccion a Pygame:

13.- Modelice en base al siguiente diagrama de clases con herencia:



La clase **FiguraGeometrica** tiene un método `dibujar()` que debe ser implementado por cada subclase, es decir, que cada figura en particular sabe cómo dibujarse en la pantalla que recibe como parámetro. Para el desarrollo de dicho método, utilice las funciones del módulo `Draw` de `Pygame`.

Para manejar los objetos de estas clases haga uso de polimorfismo, guardando varias figuras en una estructura para luego dibujarlas.