**Auxiliar Herencia/excepciones/ventanas**

1. Dadas las clases siguientes:

import math

class **Figura**:

def **\_\_init\_\_(**self,x): self.x=x

class **Circulo**(Figura):

def **area**(self): return math.pi\*self.x\*\*2

def **perimetro**(self): return 2\*math.pi\*self.x

class **Cuadrado**(Figura):

def **area**(self): return self.x\*\*2

def **perimetro**(self): return 4\*self.x

Escribir un programa para una ventana que permita dibujar un círculo o un cuadrado (en un canvas de 100x100 pixeles) y permita mostrar el area y el perímetro ingresando el radio o el lado (en pixels) a través de un Entry.

##### 2. Escriba las clases necesarias para el siguiente programa:

#### r=raw\_input(“esfera o cubo?”).lower()

#### if r=="esfera":

#### cuerpo=Esfera(input("radio?"))

#### elif r=="cubo":

#### cuerpo=Cubo(input("lado?"))

#### else:

#### exit("error: esfera o cubo")

#### print "area=",cuerpo.area()

#### print "volumen=",cuerpo.volumen()

Notas. Esfera: area=4πr2 ,volumen=4/3πr3, Cubo: area=6a2,volumen=a3

**3**.La clase Bolsa está definido con las siguientes operaciones

|  |  |
| --- | --- |
| operación | significado |
| B=Bolsa() | Define objeto para guardar objetos |
| B.poner(x) | Pone objeto x en la bolsa. Excepción YaExiste si x ya estaba en la bolsa |
| B.sacar() | Saca y entrega un objeto cualquiera de la bolsa (se elige al azar entre los elementos existentes). Excepción BolsaVacia si no hay elementos |

a)Escriba la clase Bolsa suponiendo que un objeto de la clase Bolsa se representa por una lista.

b)Pruebe la clase Bolsa en un programa que genere 10 números al azar entre 1 y 10, los ponga en la bolsa y los saque y muestre todos.

c)Reescriba y pruebe la clase Bolsa suponiendo que un objeto de la clase Bolsa se representa por un diccionario.Ej:{x:x,…}.

**4 .**Escriba la clase Tabla de modo que pueda ser utilizada en el siguiente segmento de programa:

Indicación. Represente la tabla con una objeto de la clase predefinida list

T=Tabla(100) #capacidad: 100 valores

for valor in open(“archivo.txt”):

try:

T.agregar(valor)

except TablaLLena:

print “la tabla no puede tener mas de 100 valores”)

while True:

valor=input(“valor?”)

if valor==”fin”: break

try: print “esta en indice “ + T.indice(valor)

except NoExiste: print “no existe en tabla