

Tarea 1.01

Pidan el ingreso de un dato por teclado y guardenlo en una variable a

Muestren por pantalla el contenido de a

Tarea 1.02

Pidan el ingreso de un dato por teclado y guardenlo en una variable a

Convierta el dato a en un entero usando la función int

Muestren por pantalla el contenido de a

Tarea 1.03

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a+b$

Tarea 1.04

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a-b$

Tarea 1.05

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

*Muestren por pantalla el contenido de $a*b$*

Tarea 1.06

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de a/b

Tarea 1.07

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de a^b

Tarea 1.08

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a//b$

Tarea 1.09

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a\%b$

Tarea 1.10

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a==b$

Tarea 1.11

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a \neq b$

Tarea 1.12

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a < b$

Tarea 1.13

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a \leq b$

Tarea 1.14

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a > b$

Tarea 1.15

Pidan el ingreso de dos datos a y b por teclado.

Convierta a y b a enteros usando la función `int`

Muestren por pantalla el contenido de $a \geq b$

Tarea 1.16

Pidan el ingreso de cuatro datos a, b, c y d por teclado.

Convierta a, b, c y d a enteros usando la función `int`

*Muestren por pantalla el contenido de $a==b$
AND $c==d$*

Tarea 1.17

Pidan el ingreso de cuatro datos a, b, c y d por teclado.

Convierta a, b, c y d a enteros usando la función `int`

*Muestren por pantalla el contenido de $a==b$
OR $c==d$*

Tarea 1.18

Pidan el ingreso de cuatro datos a, b, c y d por teclado.

Convierta a, b, c y d a enteros usando la función `int`

*Muestren por pantalla el contenido de $a==b$
OR NOT $c==d$*

Tarea 1.19

Pidan el ingreso de un dato a por teclado.

Guarde en b el valor de a convertido en flotante

Guarde en c el valor de a convertido a string

Muestre por pantalla el contenido de a, b y c

Muestra por pantalla los tipos de a, b y c

Tarea 1.20

Pidan el ingreso de dos datos a por teclado.

Convierta los datos a flotantes

*Si el primer dato es mayor que el segundo
muestre por pantalla el cartel: mayor*

Tarea 1.21

Pidan el ingreso de dos datos a por teclado.

Convierta los datos a flotantes

Muestre por pantalla los datos ordenados de menor a mayor

Tarea 1.22

Ingresar por teclado una edad.

Devolver por pantalla un mensaje proporcionado a la edad:

Menor a 15: edad de crecer

De 15 a 40: edad del amor

De 40 a 65: edad de los amigos

De 65 en adelante: edad de los médicos

Tarea 1.23

Ingresa por teclado dos enteros y almacenarlos en a y b.

Crear una función de dos parámetros que devuelve la suma de esos parámetros

Usar la función creada para calcular la suma de a y b.

Mostrar por pantalla el resultado de la suma

Tarea 1.24

Ingresar por teclado una edad.

Mostrar por los años todos los cumpleaños que la persona tuvo hasta ahora:

Cumpleaños 1

Cumpleaños 2

...

Tarea 1.25

Crear una lista con los días de la semana

Mostrar el contenido de la lista con un único print

Mostrar el contenido de la lista elemento por elemento

Tarea 1.26

Crear una lista con los días hábiles de la semana

Agregar a la lista el sábado y el domingo

Mostrar el contenido de la lista con un único print

Tarea 1.27

Crear una lista con los días de la semana

Eliminar de la lista el sábado y el domingo

Mostrar el contenido de la lista con un único print

Tarea 1.28

Crear una lista con los días hábiles de la semana y otra lista con el sábado y el domingo

Extender la primera lista con la segunda

Mostrar el contenido de la lista con un único print

Tarea 1.29

Crear una lista l1 con todos los días de la semana

Crear una lista l2 con los días hábiles de la semana

Recorrer l1 e indicar para cada elemento si se encuentra o no en l2

Tarea 1.30

Crear una lista l1 con todos los impares de 1 a 5

*Insertar los pares en los lugares correspondientes
para que la lista quede ordenada*

Imprimir l1

Tarea 1.31

Crear una lista l1 con todos los días de la semana

Imprimir la longitud de l1

Extender l1 con si misma

Imprimir la longitud de l1

Tarea 1.32

Crear una tupla $t1$ con todos los días de la semana

Mostrar por pantalla el número de elemento correspondiente al miércoles

Tarea 1.33

Crear una tupla $t1$ con todos los días de la semana

Averiguar si domingo se encuentra entre todos los días de la semana

Tarea 1.34

Crear una tupla $t1$ con todos los días de la semana

Imprimir la longitud de $t1$

Tarea 1.35

Crear una tupla $t1$ con todos los días de la semana

Crear una lista $l1$ a partir de $t1$

Tarea 1.36

Crear una lista l1 con todos los días de la semana

Crear una tupla t1 a partir de l1

Tarea 1.37

Crear una lista l1 con todos los días de la semana

Recorrer l1 imprimiendo elemento por elemento

Tarea 1.38

Crear una diccionario d con todos los días de la semana poniendo como clave las siete primeras letras del alfabeto

Imprimir el diccionario

Tarea 1.39

Crear una tupla con la siete primeras letras del alfabeto

Recorrer la tupla y usarla como clave para imprimir los valores del diccionario construido en la tarea 1.38

Tarea 1.40

*Tomar el diccionario generado en el ejercicio 1.38
y cambiar cada valor letras mayúsculas*

Imprimir el resultado

Tarea 1.41

Tomar el diccionario del ejercicio 1.38

Imprimir su colección de claves

Tarea 1.42

Tomar el diccionario del ejercicio 1.38

Imprimir su colección de valores

Tarea 1.43

Tomar el diccionario del ejercicio 1.38

Imprimir su longitud

Tarea 1.44

Crear una tupla con los siete días de la semana

Recorrer los 365 días del año 2019

Para cada día del año imprimir el nombre del día de la semana

Tarea 1.45

Crear una tupla con las longitudes de los meses del año 2019

Recorrer el año mostrando por pantalla el número de día dentro del año y el número del día dentro del mes.

Tarea 1.46

Crear una tupla $t1$ con las longitudes de los meses del año 2019

Crear una tupla $t2$ con los nombres de los meses del año

Recorrer el año mostrando por pantalla el número de día dentro del año y el número del día dentro del mes.y el nombre del mes

Tarea 1.47

Crear una tupla $t1$ con las longitudes de los meses del año 2019

Crear una tupla $t2$ con los nombres de los meses del año

Crear una tupla $t3$ con los nombres de los días de la semana

Recorrer el año mostrando por pantalla el número de día dentro del año y el nombre del día dentro de la semana, el número del día dentro del mes.y el nombre del mes

Tarea 1.48

Crear una tupla $t1$ con las longitudes de los meses del año 2019

Crear una tupla $t2$ con los nombres de los meses del año

Crear una tupla $t3$ con los nombres de los días de la semana

Crear una lista con las fechas largas que hemos mostrado por pantalla en el ejercicio 1.46 (nombre del día dentro de la semana, número del día dentro del mes, nombre del mes)

Tarea 1.49

Pedir la introducción desde el teclado de una fecha en el formato largo.

Usar la lista generada en el ejercicio 1.48 para encontrar el número de día dentro del año.

Tarea 1.50

Generar una lista l1 con los números del 1 al 10

Generar una lista l2 que tenga una lista con los números del uno al 10 en cada uno de sus 10 elementos

Recorrer l2 con un for anidado dentro de otro for mostrando el producto de ambas coordenadas.

Tarea 1.51

Generar para cada provincia una tupla con el nombre de la provincia, su capital, su población y su PBI.

Armar un diccionario donde cada valor sea la tupla de una provincia y la clave sea el nombre de la provincia.

Mostrar para cada provincia su PBI per capita.

Tarea 1.52

Construya un generador que vaya recorriendo la sucesión aritmética:

1, 2, 3, 4, 5 ...

Imprima por pantalla los 10 primeros valores

Tarea 1.53

Construya un generador que vaya recorriendo la serie que va sumando la sucesión aritmética:

1, 3, 6, 10, 15, 21 ...

Imprima por pantalla los 10 primeros valores

Tarea 1.54

Construya un generador que vaya recorriendo la serie geométrica:

1, 2, 4, 8, 16, 32, 64 ...

Imprima por pantalla los 10 primeros valores

Tarea 1.55

Construya un generador que vaya recorriendo la suma de la serie geométrica:

1, 3, 7, 15, 31, 63, 127

Imprima por pantalla los 10 primeros valores

Tarea 1.56

A partir del programa anterior recorrer las letras de las tres primeras ciudades de la lista y mostrar cada letra por pantalla en un nuevo renglón.

Tarea 1.57

Investigar cuales son las excepciones para:

- *Sumar un número y un string*
- *Tomar int de un string que no es un número*
- *Tomar float de un string que no es un número*

Tarea 1.58

Desarrollar una función que obligue a ingresar un número entero y no continúe la ejecución hasta que no se ingresa un entero:

(Sugerencia: usar try, tomar float, y si no da error verificar que el numero ingresado es igual a int(número ingresado))

Proveer realimentación para guiar al usuario

Tarea 1.59

Investigue las excepciones que puede causar con `sqrt`, `log`, `pow`.

Tarea 1.60

Muestre por pantalla el directorio actual.

Cree un directorio que se llame abcd

Muestre el contenido del directorio actual

Muévase al directorio abcd

Muestre el contenido del directorio actual

Tarea 1.61

Recorra todo su disco rígido y muestre por pantalla los nombres de los archivos o directorios que comiencen con win

Tarea 1.62

Crear un directorio abc a partir del directorio actual

Mostrar por pantalla los archivos del directorio actual

Borrar el directorio abc

Mostrar por pantalla los archivos del directorio actual

Tarea 1.63

Usando scandir recorrer el disco C mostrando por pantalla los nombres de todos los directorios indentando con un espacio cada vez que entramos en un directorio.

Tarea 1.64

Recorrer todo el disco C

Imprimir cada directorio junto con la suma del espacio ocupado por los archivos de ese directorio.

Tarea 1.65

Repetir la tarea 1.64 usando walk

Tarea 1.66

Crear un archivo que contenga el árbol de directorios del disco c:

Tarea 1.67

Crear un archivo que contenga el árbol de directorios del disco c:

Tarea 1.68

Abrir el archivo que contenía todos los directorios del disco rígido

Recorrerlo buscando los directorios que empiezan con win

Mostrar esos directorios por pantalla

Tarea 1.69

Abrir el archivo que contenía todos los directorios del disco rígido

Recorrerlo buscando los directorios que contengan win

Mostrar esos directorios por pantalla

Tarea 1.70

Abrir el archivo que contenía todos los directorios del disco rígido

Recorrerlo buscando los directorios que terminen con win

Mostrar esos directorios por pantalla

Tarea 1.71

Abrir el archivo que contenía todos los directorios del disco rígido

Recorrerlo buscando los archivos que empiezan con win

Mostrar esos archivos por pantalla

Tarea 1.72

Abrir el archivo que contenía todos los directorios del disco rígido

Recorrerlo buscando los archivos que contengan numeros en el nombre

Mostrar esos archivos por pantalla

Tarea 1.73

Abrir el archivo que contenía todos los directorios del disco rígido

Recorrerlo buscando los archivos cuyo tamaño supere los 100 Mb

Mostrar esos archivos por pantalla

Tarea 1.74

*Crear un generador que devuelva la
sucesión de fibonaci*

Grabar los primeros 100 números en un archivo

*Leer el archivo con notepad y buscar un patrón de
pares e impares*

Tarea 1.75

Abrir el archivo generado en la tarea 1.74 y decidir para cada número si es par o impar y mostrarlo por pantalla

Tarea 1.76

Abrir el archivo generado en la tarea 1.74

Cargar el contenido en una variable

Escribir un nuevo archivo donde la variable se escriba dos veces

Repetir hasta obtener un error relacionado con la memoria o con el disco.

Capturar la excepción y usarla como condición de salida.

Tarea 1.77

Construya una clase que describa un modem genérico de los que los clientes tienen en sus casas.

Ejemplo de propiedades:

DNS primario

DNS secundario

Método: Usar DHCP

Tarea 1.78

Buscar en el primer manual de un modem que nos encontremos por ahí más propiedades que pueda tener y más comandos que le podamos enviar en forma remota.

Ajustar la clase Modem creada en la tarea 1.77 para que incluya las nuevas propiedades y métodos.

Tarea 1.79

Crear una clase Modem que tenga una propiedad marca y otra modelo.

Crear dos objetos de la clase Modem que se llamen modem1 y modem2.

Que la marca de modem1 sea Huawei y la de modem2 sea Motorola.

Crear una lista con los objetos del tipo modem creados y recorrerla imprimiendo las marcas

Tarea 1.80

Encapsular las propiedades del ejercicio 1.79

Crear un método que devuelva la marca

Crear un método que devuelva el modelo

Tarea 1.81

Retomar el resultado de la tarea 1.79

Crear un método que permita modificar la marca

Crear un método que permita modificar el modelo

Tarea 1.82

Retomar el resultado de la tarea 1.81

Agregar una propiedad encapsulada que lleve la cuenta de los bytes transmitidos por el modem.

Crear un constructor que inicialice en 0 la cantidad de bytes transmitidos

Tarea 1.83

Retomar el resultado de la tarea 1.82

Agregar un método que permita consultar los bytes transmitidos por un modem

Agregar un método que permita incrementar en una cantidad que pasará por parámetro la cantidad de bytes transmitidos por el modem

Tarea 1.84

Retomar el resultado de la tarea 1.83

Crear una lista con 10 objetos del tipo modem.

Reflejar una transmisión de 100 bytes por el primero, 200 por el segundo, ... y así hasta llegar al décimo.

Tarea 1.85

Retomar el resultado de la tarea 1.84

Tomar la lista producida y sumar la cantidad total transmitida.

Mostrar el total transmitido por pantalla.

Tarea 1.86

Generar una clase Dispositivo con marca, modelo, potencia

Generar una clase Switch que herede a Dispositivo

Agregarle la propiedad encapsulada bocas que se podrá setear en el constructor y consultar

Tarea 1.87

Continuando con la tarea 1.86

Generar una clase Hub que herede a Dispositivo.

Agregar una propiedad bocas, que esté encapsulada, pueda consultarse y modificarse.

Tarea 1.88

Configurar las clases de las tareas 1.86 y 1.87 para que Hub herede de dispositivo y Switch herede de Hub

Escriba como sería consultar las bocas del hub y del switch.

Tarea 1.89

Tomar los siguientes conceptos y distribuirlos en un árbol de categorías:

- *Cachorro*
- *Perro*
- *Lobo*
- *Cánido*
- *Mamífero*
- *Delfín*

Tarea 1.90

Retomando la tarea 1.89 escribir una propiedad distinta para cada clase:

Ejemplo:

Delfín: Aletas

Cachorro: Destetado

Mamífero: Peso

...

Tarea 1.91

Retomando la tarea 1.90 escribir el código de las clases y sus propiedades de manera que no estén encapsuladas.

Tarea 1.92

Retomando la tarea 1.91 modificar el código de las clases para que sus propiedades queden encapsuladas, puedan escribirse en el constructor y consultarse por un método específico.

Tarea 1.93

Retomando la tarea 1.92 modificar el código de las clases para puedan escribirse a partir de un método específico.

Tarea 1.94

Retomando la tarea 1.93 pedir el ingreso de un número entero por teclado.

Cargar en una lista esa cantidad de cachorros pidiendo por teclado que les ingresen los valores de cada propiedad

Tarea 1.95

Retomando la tarea 1.94 mostrar por pantalla todas las propiedades de todos los cachorros en un formato tabular

Tarea 1.96

Retomando la tarea 1.95 agregar una propiedad nombre a los cachorros.

Generar un diccionario que tenga como clave el nombre del cachorro y como valor la lista de propiedades de cada cachorro

Tarea 1.97

Retomando la tarea 1.95 crear una clase animal con una propiedad nombre.

Que todas las clases que venimos manejando desde la tarea 1.89 tengan la propiedad nombre

Tarea 1.98

Retomando la tarea 1.97 pedir el ingreso de un número entero por teclado. Contar desde 1 hasta el número y pedir en cada caso el tipo de animal que se va a ingresar, cargar las propiedades que corresponden a ese animal y guardarlas en una lista.

Tarea 1.99

Retomando la tarea 1.98 en vez de una lista que se trate de un diccionario que tiene por clave el nombre del animal y por valor el objeto

Tarea 1.100

Retomando la tarea 1.99 en vez de una lista que se trate de un diccionario que tiene por clave el nombre del animal y por valor un diccionario que tenga por clave el nombre de la propiedad del objeto y por valor el contenido de la propiedad

Tarea 1.101

Retomando la tarea 1.86 agregar a la jerarquía objetos de tipo Router y de tipo Bridge.

Tarea 1.102

Retomando la tarea 1.101 agregar a la clase switch una propiedad del tipo lista que permita guardar una tabla que indique que MAC ADDRESS están presentes en cada boca.

Crear un método reiniciar que borre la tabla.

Tarea 1.103

Retomando la tarea 1.102 crear un objeto del tipo paquete que tenga:

MAC origen

MAC destino

Datos

Checksum

Tarea 1.104

Retomando la tarea 1.103 crear una función que valide el checksum del paquete

Tarea 1.105

Retomando las tarea 1.102 a 1.104 agregar a la clase switch un método que le permita:

- Recibir un paquete por una de sus bocas*
- Buscar el destino en la tabla de MACs*
- Actualizar la tabla de MACs*
- Invocar un método transmitir que se aplique según corresponda a todas las bocas.*

Tarea 1.106

Generar una lista de los dispositivos que administran con sus propiedades y operaciones.

En base a sus propiedades y operaciones desplegar a los objetos en un árbol de clases

Tarea 1.107

Retomando la jerarquía de clases construida en el punto 1.106 distribuirse los objetos entre los participantes y escribir las clases.

Discutir para cada objeto que propiedades deberían ser modificables y cuales deberían estar encapsuladas.

Tarea 1.108

Genere archivo csv con notepad que contenga datos alegremente inventados con:

- Localidad,*
- Número de clientes activos,*
- Tráfico promedio diario.*

Construya código Python que lea ese archivo y lo muestre por pantalla

Tarea 1.109

Repita el ejercicio 1.108 pero generando ahora el archivo csv a leer desde Excel.

Sugerencia: revise lo generado en notepad antes de ejecutar el código y revise si debe especificar algún delimitador

Tarea 1.110

Queremos grabar la serie de los primeros 10 números naturales, sus cuadrados y sus cubos.

Genere la lista de listas que permita grabar un archivo csv con esa información.

Grabe el archivo y verifique el resultado.

Tarea 1.111

Quiere grabar la lista de nombres y cumpleaños de sus 5 mejores amigos.

Genere la lista de listas que permita grabar un archivo csv con esa información.

Grabe el archivo y verifique el resultado.

Tarea 1.112

Leer en diccionarios el archivo utilizado en la tarea 1.108

Tarea 1.113

Leer en diccionarios el archivo utilizado en la tarea 1.109

Tarea 1.114

*Escribir desde diccionarios el mismo contenido
generado para la tarea 1.110*

Tarea 1.115

*Escribir desde diccionarios el mismo contenido
generado para la tarea 1.111*

Tarea 1.116

*Tomar el archivo csv usado en la tarea 1.108
desde Excel y grabarlo como xlsx*

*Luego leerlo y mostrar el contenido por pantalla
desde Python*

Tarea 1.117

*Tomar el archivo csv usado en la tarea 1.109
desde Excel y grabarlo como xlsx*

*Luego leerlo y mostrar el contenido por pantalla
desde Python*

Tarea 1.118

Escribir en Excel el mismo contenido generado para la tarea 1.110

Tarea 1.119

Escribir en Excel el mismo contenido generado para la tarea 1.111

Tarea 1.120

Genere un programa que permita cambiar el nombre de un curso:

Pida el nombre viejo y el nuevo por teclado

Conéctese a la base de datos y ejecute una consulta que realice el update

Tarea 1.121

Investigue en internet la instrucción SQL necesaria para crear una tabla en MySQL

Construya la instrucción necesaria para crear una tabla de Alumnos con dos campos: IdAlumno y Nombre

Cree un programa Python que ejecute la creación.

Tarea 1.122

Trate de crear la tabla de vuelta para que aparezca el error de no se puede crear una tabla con un nombre que ya existe.

Agregue un bloque try except para proteger al programa del error y de aviso al usuario de la circunstancia.

Tarea 1.123

Cambiando en el connect la dirección del servidor, el puerto, el nombre de usuario, la password y la indicación del dbschema explore cuales son los errores que se generan y proteja al programa contra ellos al tiempo que informa al usuario del problema ocurrido.

Tarea 1.124

Insertar tres registros en la tabla creada sin repetir la clave primaria.

Tarea 1.125

Insertar tres registros en la tabla creada repitiendo en alguno la clave primaria.

Tomar nota del error que aparece

Incorporar un bloque try except para evitar que el programa se interrumpa e informar al usuario del error.

Tarea 1.126

Generar un programa que recupere todos los registros de la tabla que usó en la tarea 1.125 y los muestre por pantalla.

Tarea 1.127

Generar un programa que pregunte el nombre de una tabla y luego recupere todos los registros de esa tabla y los muestre por pantalla.

Proteger al programa contra el error de poner un nombre de tabla que no exista usando un bloque try except.

Tarea 1.128

Adaptar el programa de la tarea 1.127 para que lea los registros de a uno y los vaya presentando por pantalla de a uno.

Tarea 1.129

Adaptar el programa de la tarea 1.127 para que lea los registros de a uno y los vaya almacenando en una lista que tenga, en cada elemento, los datos de una fila organizada como una lista de valores.

Tarea 1.130

Adaptar el programa de la tarea 1.127 para que lea los registros de a uno y los vaya almacenando en una lista que tenga, en cada elemento, los datos de una fila organizada como un diccionario cuyas claves sean los nombres de los campos y los valores sean los datos almacenados.

Tarea 1.131

Adaptar el programa de la tarea 1.128 para que lea los registros de a uno y los vaya almacenando en un diccionario en el cual la clave es la clave primaria de la tabla y el valor es un nuevo diccionario que tiene los datos de cada registro organizados como en la tarea 1.130.

Tarea 1.132

Escribir un programa que pida el nombre de la tabla y un valor para la clave primaria.

Que muestre el valor de la descripción para esa clave primaria y pida un valor nuevo.

Que realice el update y no ponga el commit sino que pida un nuevo valor al usuario que haga que el programa quede en espera

Tarea 1.133

Que dos participantes traten de ejecutar el programa de la tarea 1.132 apuntando a la misma tabla y al mismo registro.

Tomar nota del error que da y aprender a cubrirlo con un bloque try except que reintente hasta que la operación se pueda realizar.

Tarea 1.134

Generar un programa que trate de modificar un dato en la tabla A y luego otro en la tabla B y sólo al final de commit.

Generar un segundo programa que trate de realizar la misma operación en el orden inverso.

Sincronizar la ejecución de ambos programas mediante dos usuarios que hacen el primer cambio y luego intentan el segundo. Tomar nota de lo que ocurre.

Tarea 1.135

Generar una tabla con una clave primaria y dos valores.

Generar un programa que pida una clave primaria, muestre el registro y luego pida un valor nuevo para cada uno de los dos datos.

Ejecutar a la vez el programa sobre el mismo registro en forma sincronizada. Uno de los usuarios modifica el primer valor y el otro el segundo. ¿Cómo queda la base de datos?

Tarea 1.136

Generar las 6 clases precedentes con sus propiedades no públicas y crear sólo el método para consultarlas al tiempo que los datos se cargan sólo en el constructor.

Crear un programa que permita al usuario cargar anfibios indicando el tipo que cargará (Dukw o LVT-7) y pedirle los atributos correspondientes ir creando los objetos y almacenarlos en una lista.

Tarea 1.137

Diseñar las tablas necesarias para grabar los objetos que hemos manejado en la tarea 5.17:

Dukw:

IdDukw

IdMotor

Ruedas

Calado

Carga

LVT7:

IdLVT7

IdMotor

Ruedas

Calado

HMGs

Motores:

IdMotor

Marca

Modelo

Tarea 1.138

Crear un método grabar para los Dukws, LVT7s y Motores.

Ese método deberá conectarse a la base de datos, grabar los datos en las tablas necesarias distinguiendo por si mismo si graba un objeto nuevo o si esta actualizando uno ya existente.

Tarea 1.139

Perfeccionar la tarea 1.138 para atrapar los errores y guiar al usuario para corregir:

Grabar un vehículo con motor inexistente.

Grabar un vehículo que está siendo modificado por otro usuario.

Tarea 1.140

Retomar la tarea 1.136 y, tras la carga de 5 elementos pasar a grabar la lista de objetos aprovechando el polimorfismo que me ofrece el método grabar

Tarea 1.141

Crear un programa que pueda leer desde las tablas los objetos existentes y almacenarlos en los objetos correspondientes recreando la lista de objetos.

Si todos trabajamos sobre la misma tabla nos vamos a encontrar con muchos más objetos de los que cada uno creó...

Tarea 1.142

Obtener dos strings desde el teclado

Ordenarlos alfabéticamente.

Insertarlos dentro del string que diga:

La cadena & string 1 & está antes de & string 2.

Tarea 1.143

Ingresa en cuatro variables los datos de un partido de fútbol:

Local, goles_local, visitante, goles visitante.

Luego insertelos en un string que brinde el resultado nombrando primero al ganador o al local si hubo empate.

Tarea 1.144

Retomando la tarea 1.96 encargarse de imprimir en un solo renglón las propiedades de cada cachorro usando %

Tarea 1.145

Obtener dos strings desde el teclado

Ordenarlos alfabéticamente.

Insertarlos dentro del string que diga:

La cadena & string 1 & está antes de & string 2.

Usar str.import()

Tarea 1.146

Ingresa en cuatro variables los datos de un partido de fútbol:

Local, goles_local, visitante, goles visitante.

Luego insertelos en un string que brinde el resultado nombrando primero al ganador o al local si hubo empate.

Usar str.import()

Tarea 1.147

Retomando la tarea 1.96 encargarse de imprimir en un solo renglón las propiedades de cada cachorro usando %

Usar str.import()

Tarea 1.148

Obtener dos strings desde el teclado

Ordenarlos alfabéticamente.

Insertarlos dentro del string que diga:

La cadena & string 1 & está antes de & string 2.

Usar f-strings

Tarea 1.149

Ingresa en cuatro variables los datos de un partido de fútbol:

Local, goles_local, visitante, goles visitante.

Luego insertelos en un string que brinde el resultado nombrando primero al ganador o al local si hubo empate.

Usar f-strings

Tarea 1.150

Retomando la tarea 1.96 encargarse de imprimir en un solo renglón las propiedades de cada cachorro usando %

Usar f-strings

Tarea 1.151

Escribir una función lambda que ayude en la conversión de unidades.

La idea es generar las 6 funciones que permitan convertir entre cm, m y km.

Definir y probar las 6 funciones usando una función lambda

Tarea 1.152

Generar los primeros 5 términos de la serie aritmética.

Graficarlos como barras verticales

Tarea 1.153

Generar los primeros 5 términos de la serie aritmética.

Graficarlos como barras horizontales

Tarea 1.154

Generar los primeros 5 términos de la serie aritmética.

Graficarlos como una línea de ancho 5 y color rojo

Tarea 1.155

Generar los primeros 5 términos de la serie aritmética.

Graficarlos como un scatterplot usando X como marcadores.

Tarea 1.156

Generar un programa que cargue la edad y el peso de todos los participantes del curso

Graficarlos como un scatterplot usando o como marcadores.

Tarea 1.157

Tomar los pesos del ejercicio 1.156 y graficarlo como un conjunto de barras verticales.

Tarea 1.158

Tomar las edades del ejercicio 1.156 y graficarlo como un conjunto de barras horizontales.

Tarea 1.159

Juntar en una figura los dos gráficos de las tareas 1.157 y 1.158 utilizando subplot

Tarea 1.160

Ingresa desde el teclado un número de lados para un polígono

Ingresa luego las coordenadas de cada vértice del polígono.

Extraiga el máximo y el mínimo para fijar la escala.

Dibuje el polígono y rellénelo de azul.

Tarea 1.161

A partir del ejercicio 1.160 guarde en un archivo de texto plano los datos del polígono.

Adapte ahora el programa para en vez de ingresar los datos por teclado se tomen del archivo de texto plano.

Tarea 1.162

Cree una clase vértice cuyas propiedades públicas sean X,Y

Cree una clase polígono cuyas propiedades públicas sean una lista de vértices y un color de relleno.

Genere un método graficar para la clase polígono que lo muestre usando matplotlib y fill

Tome el polígono desde el archivo plano usado en 1.161

Tarea 1.163

Considere dos funciones:

- $F1(x) = 3x+2$
- $F2(x) = x*x + 5$

Para los valores de x entre 1 y 10 calcule los valores de Y1 e Y2 usando F1 y F2

Grafique ambas funciones y pinte de verde la zona entre ellas.

Tarea 1.164

*Definir la función $f3 = x*x*x+3$.*

Incluirla en el gráfico del ejercicio 1.163

Pintar de amarillo el area entre $f2$ y $f3$.

Tarea 1.165

Una persona quiere obtener un aumento de sueldo. Para eso prepara para su jefe la siguiente información:

- a. Sus últimos 6 meses de sueldo*
- b. Los últimos 6 alquileres*
- c. El total de supermercado de los últimos 6 meses*
- d. El total de gastos escolares de sus hijos de los últimos 6 meses*

Con todo eso construye un gráfico en Python donde muestra superpuestos sus gastos y una línea con sus ingresos.

Tarea 1.166

El jefe le quiere responder al empleado con similares argumentos. Para eso prepara:

- a. Las ventas totales de los últimos 6 meses.*
- b. Los costos de operación de los últimos 6 meses.*
- c. Los sueldos totales de los últimos 6 meses*

Con todo eso construye un gráfico en Python donde muestra superpuestos los gastos de la empresa y una línea que muestra las ganancias que quedan.

Tarea 1.167

Dibujar con flechas que giren en sentido horario los cuatro lados del cuadrado que va de $(0,0)$ a $(0,1)$ a $(1,1)$ a $(1,0)$

Tarea 1.168

Dibuje con quiver un campo vectorial plano que represente un fluido girando en sentido horario con una velocidad creciente con el radio.

Sugerencia:

Prueba con $q = ax.\text{quiver}(X, Y, V, -U)$

Tarea 1.169

*Crear una lista con los sueldos de todos los conocidos.
Generar un histograma con esos sueldos.*

Tarea 1.170

Generar un histograma con las alturas aproximadas de todas las personas que conoce.

Tarea 1.171

Los jefes quieren hacerse una idea de cómo consumen internet los clientes.

*Para eso contamos con los 10 que mas consumen.
Tenemos para cada uno el máximo, el mínimo, la mediana
y el desvío estándar.*

Genere un boxplot mostrando esa información