TAREA 6: Fecha de entrega: 10 de Febrero del 2025

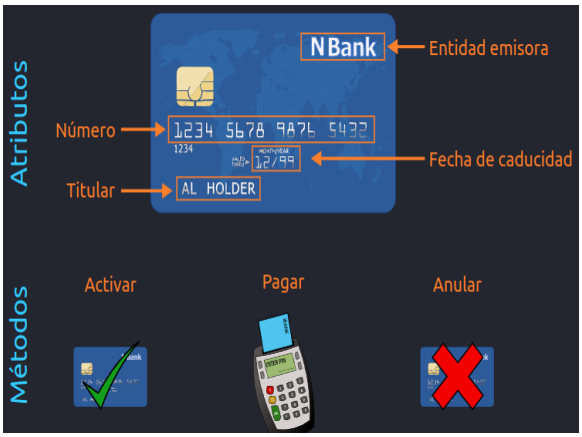
Instrucciones

- Resuelve los siguientes problemas de manera clara y completa. Asegúrate de incluir todos los pasos necesarios para llegar a la solución. Aplica los conceptos aprendidos en clase y justifica tus respuestas cuando sea necesario.

Problemas :

Programación Orientada a Objetos (POO)

1. Una tarjeta de crédito puede representarse como un objeto:



   - Atributos: Número de la tarjeta, titular, balance, fecha de caducidad, pin, entidad emisora, estado (activa o no), etc.
   - Métodos: Activar, pagar, renovar, anular.

   Acceso a los atributos y métodos de un objeto

   - dir(objeto): Devuelve una lista con los nombres de los atributos y métodos del objeto objeto.

   Para ver si un objeto tiene un determinado atributo o método se utiliza la siguiente función:

   - hasattr(objeto, elemento): Devuelve True si elemento es un atributo o un método del objeto objeto y False en caso contrario.

   Para acceder a los atributos y métodos de un objeto se pone el nombre del objeto seguido del operador punto y el nombre del atributo o el método.

   - objeto.atributo: Accede al atributo atributo del objeto objeto.
   - objeto.método(parámetros): Ejecuta el método método del objeto objeto con los parámetros que se le pasen.

   En Python los tipos de datos primitivos son también objetos que tienen asociados atributos y métodos.

2. Vamos a crear una clase llamada **Persona**. Sus atributos son: nombre, edad y DNI. Construye los siguientes métodos para la clase:

   - Un constructor, donde los datos pueden estar vacíos.
   - Los setters y getters para cada uno de los atributos. Hay que validar las entradas de datos.
   - **mostrar():** Muestra los datos de la persona.
   - **esMayorDeEdad():** Devuelve un valor lógico indicando si es mayor de edad.

3. Crea una clase llamada Cuenta que tendrá los siguientes atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional. Construye los siguientes métodos para la clase:

   - Un constructor, donde los datos pueden estar vacíos.
   - Los setters y getters para cada uno de los atributos. El atributo no se puede modificar directamente, sólo ingresando o retirando dinero.
   - **mostrar()**: Muestra los datos de la cuenta.
   - **ingresar(cantidad):** se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
   - **retirar(cantidad):** se retira una cantidad a la cuenta. La cuenta puede estar en números rojos.

4. Vamos a definir ahora una **Cuenta Joven**, para ello vamos a crear una nueva clase **CuantaJoven** que deriva de la anterior. Cuando se crea esta nueva clase, además del titular y la cantidad se debe guardar una bonificación que estará expresada en tanto por ciento.Construye los siguientes métodos para la clase:

   - Un constructor.
   - Los setters y getters para el nuevo atributo.
   - En esta ocasión los titulares de este tipo de cuenta tienen que ser mayor de edad., por lo tanto hay que crear un método **esTitularValido()** que devuelve verdadero si el titular es mayor de edad pero menor de 25 años y falso en caso contrario.
   - Además la retirada de dinero sólo se podrá hacer si el titular es válido.
   - El método **mostrar()** debe devolver el mensaje de **Cuenta Joven** y la bonificación de la cuenta.

   Piensa los métodos heredados de la clase madre que hay que reescribir.

5. Realizar un programa que conste de una clase llamada **Alumno** que tenga como atributos el nombre y la nota del alumno. Definir los métodos para inicializar sus atributos, imprimirlos y mostrar un mensaje con el resultado de la nota y si ha aprobado o no.

6. Realizar un programa que tenga una clase **Persona** con las siguientes características. La clase tendrá como atributos el nombre y la edad de una persona. Implementar los métodos necesarios para inicializar los atributos, mostrar los datos e indicar si la persona es mayor de edad o no.

7. Desarrollar un programa que cargue los datos de un triángulo. Implementar una clase con los métodos para inicializar los atributos, imprimir el valor del lado con un tamaño mayor y el tipo de triángulo que es (equilátero, isósceles o escaleno).

8. Realizar un programa en el cual se declaren dos valores enteros por teclado utilizando el método `__init__`. Calcular después la suma, resta, multiplicación y división. Utilizar un método para cada una e imprimir los resultados obtenidos. Llamar a la clase Calculadora.

9. Realizar una clase que administre una agenda. Se debe almacenar para cada contacto el nombre, el teléfono y el email. Además deberá mostrar un menú con las siguientes opciones.

Bucles For y While

1. Pedir al usuario dos números m y n. Los mismos deben ser mostrados por pantalla, teniendo en cuenta que deben incluirse tanto m como n. Es decir, el usuario no debe observar que se ejecuta $n - 1$.

2. Pedir al usuario que ingrese 2 numeros, luego, debe mostrarse el rango de esos 2 números, pero, solo imprimiendo aquellos que sean impares. Dado que la fórmula para saber los número impares es: $2 \times n + 1$

3. Escriba un programa que pregunte una y otra vez si desea continuar con el programa, siempre que se conteste exactamente sí (en minúsculas y con tilde).

4. Escriba un programa que simule una Alcancia. El programa solicitará primero una cantidad, que será la cantidad de dinero que queremos ahorrar. A continuación, el programa solicitará una y otra vez las cantidades que se irán ahorrando, hasta que el total ahorrado iguale o supere al objetivo. El programa no comprobará que las cantidades sean positivas.

5. Escriba un programa que solicite una contraseña (el texto de la contraseña no es importante) y la vuelva a solicitar hasta que las dos contraseñas coincidan, con un límite de tres peticiones.

6. Escribir un programa que pida al usuario una palabra y la muestre por pantalla 10 veces.

7. Escribir un programa que pregunte al usuario su edad y muestre por pantalla todos los años que ha cumplido (desde 1 hasta su edad).

8. Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla todos los números impares desde 1 hasta ese número separados por comas.

9. Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla la cuenta atrás desde ese número hasta cero separados por comas.

10. Escribir un programa que pregunte al usuario una cantidad a invertir, el interés anual y el número de años, y muestre por pantalla el capital obtenido en la inversión cada año que dura la inversión.

- Añadir contacto
- Lista de contactos
- Buscar contacto
- Editar contacto
- Cerrar agenda

10. En un banco tienen clientes que pueden hacer depósitos y extracciones de dinero. El banco requiere también al final del día calcular la cantidad de dinero que se ha depositado. Se deberán crear dos clases, la clase cliente y la clase banco. La clase cliente tendrá los atributos nombre y cantidad y los métodos `__init__`, depositar, extraer, **mostrar_total**. La clase banco tendrá como atributos 3 objetos de la clase cliente y los métodos `__init__`, operar y **deposito_total**.

    (a) Respuestas completas a cada pregunta.

    (b) Conclusiones.

    (c) Bibliografía.

2. Programas (50 %)

    (a) Programas correspondientes a cada pregunta.

    (b) La respuestas deben estar editadas para ser mostradas en el momento de la ejecución del programa.

Rubrica (100 %)

1. Reporte (50 %) El reporte debe ser presentado por escrito, redactado con sus propias palabras y editado utilizando la herramienta LaTeX.