

## PRÁCTICA\_8: POO

### Programación Orientada a Objetos

1. Programa en python que gestiona una agenda. Tendremos la clase agenda que será una lista de objetos de tipo contacto. Cada contacto tendrá nombre, teléfono, email. Mostrar un menú que permita:
  - Añadir un contacto: si no está ya en la agenda (los contactos se identifican por su teléfono)
  - Buscar un contacto: permitirá buscar por nombre(puede haber varios contactos con el mismo nombre), teléfono o email.
  - Listar contactos
  - Eliminar contactos: dado un teléfono, eliminar ese contacto

Se tendrán en cuenta a la hora de implementar:

- a) Las clases agenda y contacto irán en un paquete llamado “modelos”
- b) La clase agenda implementa los métodos para añadir, buscar, eliminar y el método especial `__str__` para mostrar la lista de contactos
- c) La función buscar se implementa para los tres criterios de búsqueda, recibirá un parámetro que nos indica cuál es el elegido en cada momento. Utiliza este parámetro por defecto para buscar por nombre
- d) La función eliminar se implementa utilizando la función filter
- e) La clase contacto implementa sólo getter para el nombre, para teléfono y email implementa getter y setter
- f) La clase contacto implementa el método especial `__eq__`
- g) Implementa en un módulo las funciones que pintan los dos menús (el principal y el de búsqueda)
- h) Implementa en otro módulo una función para validar la opción introducida para cada menú (usarás la misma función para ambos). Esta función lanzará una excepción `OutOfRange` con el mensaje “Opción fuera del intervalo”. Esta excepción será tuya e irá implementada en un módulo
- i) Debes controlar para las opciones introducidas en el menú que el formato es correcto(introducen un número entero).

- j) En este mismo módulo también implementará una función que valida que el teléfono sea correcto (empieza por 6 y tiene longitud 9). Si no lo es lanza una excepción tuya InvalidFormat. Esta excepción también irá en un módulo (distinto a la otra).
- k) Ambas excepciones deben ir en un paquete

2. Desarrollar un programa que conste de una clase padre Cuenta y dos subclases PlazoFijo y CuentaAhorro. Todas las cuentas llevan nombre y apellidos del titular, saldo e identificador. Dicho identificador se genera automáticamente concatenando las dos primeras letras del nombre del titular + las dos últimas letras de su último apellido+ FJ O AH (dependiendo del tipo de cuenta)+ número de orden de dicha cuenta (deberás añadir variable de clase para controlar el número de cuentas independientemente del tipo)

Titular: Rosa Rodríguez García

Cuenta Ahorro

Tercera cuenta que voy a crear

identificador = ROIAAH3

Todas las clases tendrán un método para mostrar la información

La clase PlazoFijo tendrá dos atributos propios, plazo e interés. Tendrá un método para obtener el importe del interés (cantidad\*interés/100)

Aplicación que gestione una lista de cuentas y permita:

- a) Añadir una nueva cuenta
- b) Mostrar información de todas las cuentas
- c) Mostrar información de todas las cuentas de un tipo (plazo fijo o ahorro)
- d) Mostrar información de una cuenta concreta, dado su titular (si tiene varias se muestran todas)
- e) Calcular el interés dado el id de una cuenta

3. Añadir a la clase agenda anterior un método que transforma la lista de contactos en un diccionario siendo los nombres las claves

4. Desarrollar un programa con una clase llamada Lista. Tendrá como atributo una lista. Redefinir los operadores de suma, resta, multiplicación y división respecto a un valor entero.

5. El ministerio ha implementado un proceso de adjudicación de becas. Hay dos tipos de becas: Las becas para fp y las de libros. Una beca sólo puede ser de uno de estos dos tipos. La solicitud de una beca de un alumno tiene como datos:

fp: nif , localidad, renta per cápita y fecha de nacimiento.

libros: nif, nota, renta per cápita y fecha de nacimiento.

Nuestro programa en primer lugar dará de alta en una lista las solicitudes(controlar todas las excepciones al introducir datos), teniendo en cuenta que para cada solicitud se genera un código. Cada tipo de solicitud lleva su propia numeración, además en las de fp se añade FP y en las de libros LI, p. ej. el código de la segunda solicitud de libros será "LI2" y el de la segunda de FP será "FP2".

Una vez generada la lista, hacer un menú que permita:

- a) Dado un nif buscar la solicitud y dar la posibilidad de modificar su renta. Si el nif no pertenece a ninguna solicitud lanzar una excepción propia "SolicitudNoExisteError"
- b) Hacer un método polimórfico puntuarBecas que calcula los puntos que se dan a cada solicitud de la siguiente forma:
  - Becas FP: Si es de Madrid 1 pto. Además si la renta es menor de 6000 se le añaden 2 ptos y si está entre 6000 y 10000 , ambos incluidos se le añade 1 pto. Si la edad es mayor de 21 años se le dan 2 ptos más.
  - Becas Libros: Si la nota es 7 u 8 se le da un pto, si es 9 ó 10 dos puntos. Además si la renta es menor de 6000 euros 2 ptos más. Si la edad es menor o igual a 20 años 2 ptos más.
- c) Eliminar las becas con 0 puntos.
- d) Sumar un punto a las becas de fp de Madrid.
- e) Mostrad la información completa de todas las becas.