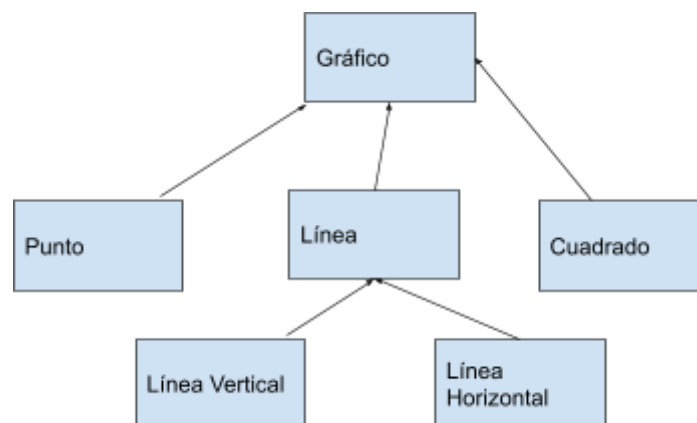


## TAREA 2 TÉCNICAS DE PROGRAMACIÓN

1.- Cree 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.

2.- Se quiere crear una jerarquía de elementos gráficos como en la siguiente figura:



Crear las clases necesarias para poder pintar los elementos gráficos de la jerarquía. Para completar la funcionalidad proporcionada por esta jerarquía, crear un array de objetos de la clase Gráfico y haga que se muestre por pantalla.

3.- Crea una clase llamada 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.

Toda la interfaz de usuario debe ser hecha por consola como los ejemplos que realizamos en clase.

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

5.- Haz una clase llamada Password que siga las siguientes condiciones:

Que tenga los atributos longitud y contraseña . Por defecto, la longitud sera de 16.

Los constructores serán los siguiente:

Un constructor por defecto.

Un constructor con la longitud que nosotros le pasemos. Generará una contraseña aleatoria con esa longitud.

Los métodos que implementa serán:

- `esFuerte()`: devuelve un booleano si es fuerte o no, para que sea fuerte debe tener más de 2 mayúsculas, más de 1 minúscula y más de 5 números.
- `generarPassword()`: genera la contraseña del objeto con la longitud que tenga.
- Método `get` para contraseña y longitud.
- Método `set` para longitud.

Ahora, crea una clase ejecutable:

1. Crea un array de Passwords con el tamaño que tu le indiques por teclado.
2. Crea un bucle que cree un objeto para cada posición del array.
3. Indica también por teclado la longitud de los Passwords (antes de bucle).
4. Crea otro array de booleanos donde se almacene si el password del array de Password es o no fuerte (usa el bucle anterior).
5. Al final, muestra la contraseña y si es o no fuerte (usa el bucle anterior). Usa este simple formato:

contraseña1 valor\_booleano1

contraseña2 valor\_bololeano2

6.- realizar una jerarquía de de clases utilizando herencia que modele los seres vivos capacidad de habla. Las clases deben modelar al menos a los loros, los profesores y los alumnos. Todas las clases en la jerarquía elaboradas deben disponer de un método habla sin argumentos que proporcione una salida por pantalla similar a la siguiente:

Hola, me llamo Juan y se hablar.  
Soy racional.  
Tengo 40 años.  
Nací el 1 de enero de 1965  
Soy profesor.

Para que el ejercicio sea interesante es necesario que todos los objetos habladores tengan un conjunto de características que les diferencian de los demás, por ejemplo, que los loros no sean conscientes de su edad o su fecha de nacimiento.







7.-














Crear un conjunto de clases usando los conocimientos de programación orientada a objetos para modelar estos personajes de pokemon.

Entender la jerarquía en herencia por decir *Bulbasaur*, es la clase padre de *Ivysaur*, y son de tipo planta.

saque la información necesario de

[https://www.wikidex.net/wiki/Lista\\_de\\_Pok%C3%A9mon\\_de\\_la\\_primera\\_generaci%C3%B3n](https://www.wikidex.net/wiki/Lista_de_Pok%C3%A9mon_de_la_primera_generaci%C3%B3n)

#	Mini sprite	Nombre	Tipo 1	Tipo 2	Japonés
001		Bulbasaur			フシギダネ
002		Ivysaur			フシギソウ

003		Venusaur			フシギバナ
004		Charmander			ヒトカゲ
005		Charmeleon			リザード
006		Charizard			リザードン
007		Squirtle			ゼニガメ
008		Wartortle			カメール
009		Blastoise			カメックス

Son libres para analizar su representación de la mejor forma posible, incluidos sus habilidades, y otras características que vean convenientes.