En los ejercicios se valorarán aspectos como:

- **Corrección, facilidad de mantenimiento (claridad del código), fiabilidad.**
- **Eficiencia, portabilidad, reusabilidad.**

La no utilización de programación estructurada conllevará la anulación de la totalidad del ejercicio.

1.- Crea una clase llamada **Dinero** (puedes hacerla abstracta) con los siguientes atributos privates:

- color (String), ancho (float), alto (float).

Un constructor, los métodos get y set de todos los atributos y un método llamado valor(), puedes hacerlo abstract o que devuelva un cero.

Crea una clase llamada **Billete** que herede de Dinero y añada un nuevo atributo llamado valor, entero y también private. Implementa al menos 3 constructores, los set y gets e implementa al menos un método llamado valor() que devuelva el valor del billete.

Crea una clase llamada **Tarjeta** que herede de Dinero y añada tres nuevos atributos llamados num (String), saldo y crédito, float ambos y también private. Implementa al menos 3 constructores, los set y gets e implementa al menos un método llamado valor() que devuelva el saldo de la tarjeta + el crédito.

Puntos: 1 punto.

2.- Crea una clase llamada **Monedero**, tendrá al menos un atributo que consistirá en un ArrayList de objetos de tipo Dinero. Implementa al menos 3 constructores, **no tendremos set ni get del atributo** e implementa al menos los siguientes métodos:

- float **disponible**() debe devolver la cantidad disponible en mi monedero, la suma del valor de todos los billetes y los saldos (saldo + crédito) de mis tarjetas.

  Puntos: 0,5 puntos.
- float **metalico**() debe devolver la cantidad en metálico (billetes) que llevo en mi monedero.

  Puntos: 0,5 puntos.
- boolean **addDinero**(Dinero x) añade un objeto de tipo Dinero al ArrayList, devolverá true si la inserción se realiza correctamente y false en caso contrario (tarjeta repetida). **Puntos: 1 punto.**
- boolean **removeBillete**(int x) borra un objeto de tipo Billete del ArrayList, devolverá true si el borrado se realiza correctamente y false en caso contrario (no hay billetes de ese valor). **Puntos: 1 punto.**
- boolean **sacaSaldo**(String num, float x) devolverá true si se realiza correctamente (existe la tarjeta con ese num y hay (saldo+crédito) disponible) y false en caso contrario (no hay suficiente dinero en la tarjeta). **Puntos: 1 punto.**
- String **muetraMonedero()** devuelve el contenido del monedero.

  Puntos: 1 punto.

3.- Crea una clase llamada **Persona** con los atributos de dni (String), nombre (String) y Monedero. Implementa al menos 3 constructores y los set y gets. Implementa al menos el siguiente método:

- boolean **paga**(float cantidad) una persona intenta pagar una cantidad con una tarjeta, si es posible pagar, disminuirá el saldo y/0 el crédito de la tarjeta.

  Puntos: 2 puntos.

4.- Programa principal. Realiza los siguientes métodos:

- Persona **nuevaPersona**() devuelve un objeto de tipo persona. Aspectos a tener en cuenta:
  - El dni no puede estar en blanco (valorará la validación de dni correcto). El nombre no puede estar en blanco. Hay que dar la posibilidad de no tener monedero, o comenzar con un Billete.

  Puntos: 1 punto.

5.- Programa principal. Realiza los siguientes pasos:

- Crea dos personas, añade tres billetes a la primera, y dos billetes y dos tarjetas a la segunda. Muestra el disponible y el metálico de ambas personas.
- Muestra el monedero de la segunda persona.
- La segunda persona intenta pagar una cantidad de dinero, y muestra si lo consigue o no.

Puntos: 1 punto.