

Esercizio 1.

Si realizzi una applicazione per gestire il rifornimento di un magazzino merci. Si progettino:

- una interfaccia Articolo con contenga i metodi:
 - String getCodice()
 - int getGiacenza()
 - int getMin()
 - double getPrezzoUnitario()
- una classe Prodotto che implementi l'interfaccia Articolo. Per ogni prodotto vogliamo memorizzare il codice (univoco), la giacenza (numero di pezzi in magazzino), la giacenza minima che deve essere sempre presente in magazzino ed il prezzo unitario.
- una classe Servizio che implementi l'interfaccia Articolo. Per i servizi, la giacenza ed il numero minimo di pezzi sono entrambi 0.
- una classe Magazzino con una struttura dati per memorizzare sia prodotti che servizi. La classe Magazzino deve implementare i seguenti metodi:
 - vuoto(): restituisce true se il magazzino è vuoto, false altrimenti;
 - valore(Articolo a): il valore di tutti i pezzi dell'articolo "a" presenti in magazzino (dato dal prezzo unitario per il numero di pezzi in giacenza);
 - valore(): il valore di tutti gli articoli in magazzino;
 - daAcquistare(): la lista degli articoli da acquistare (per cui la giacenza è inferiore al livello minimo)
 - costo(Articolo a): il costo da sostenere per acquistare l'articolo "a" affinché la giacenza raggiunga il minimo richiesto;
 - costo(): il costo da sostenere affinché tutti gli articoli raggiungano la giacenza minima.

Esercizio 2.

Si progetti una nuova eccezione MagazzinoException e si scriva il metodo controlloMagazzino() che restituisce true se per tutti gli articoli è presente la giacenza minima, altrimenti solleva l'eccezione MagazzinoException.

Esercizio 3.

Si dica cosa stampa il seguente programma motivando la risposta e indicando, per ogni chiamata ad un metodo, la lista delle firme candidate.

```

public class A {
    public int m(A a, B b) {return 2; };
    public int m(B a, B b) {return 10; };
    public B m(B c, A b) {return c; }
}

public class B extends A {
    public int m(A a, B b) {return 3; }
    public A m(C c, B b) {return b; }
    public Object m(C c, Object o) {return o; }
}

public class C extends B {
    public A m(C c, A a) {return c; }
    public int m(B a, B b) {return 5; }
}

public class Test {
    public static void main(String[] args) {
        C gamma = new C();
        B beta = gamma;
        A alfa = gamma;
        System.out.println(alfa.m(beta, beta));
        System.out.println(gamma.m(beta.m(gamma, beta),beta));
    }
}

```