INTERFACE VS ABSTRACT CLASS

Type of methods	Interface can have **only abstract methods**	Abstract class can have **abstract and non-abstract methods**
Type of variables	Interface has only **static and final variables**.	Abstract class can have **final, non-final, static and non-static variables**.
Implementation	Interface **can't** provide the **extenstion of abstract class**.	Abstract class **can** provide the **implementation of interface**
Inheritance vs Abstraction	A class **implements** an interface	A class **extends** an abstract class
Multiple implementation	Interface can **extend** another Java **interface only**	Abstract class can **extends another** Java **class** and **implements multiple** Java **interfaces**.
Accessibility of Members	**Members** of an interface are **public** by default.	Abstract class can have class members like **private, protected, etc**.
Meaning	Interface represents a **behaviour** of a class (eg. Comparable, Comparator, Serializable, Cloneable, etc…)	Abstract class represents a (partial) **entity**
Commons	Interfaces **cannot be instantiated**	Abstract classes **cannot be instantiated**

ESEMPIO SU ASTRAZIONE

Modellizzare in Java il seguente problema.

PARTE 1

Si vuole realizzare un sistema di gestione di un magazzino in cui vengono stoccate solo scarpe da ginnastica.

Ogni scarpa è caratterizzata da:

- Codice
- Marca
- Modello
- Data di stoccaggio
- Prezzo
- Misura
- Set di colori

Inoltre è necessario conoscere il volume di ingombro di ogni scatola di scarpe e le dimensioni spaziali di ingombro (altezza, larghezza e profondità) del contenitore.

Il codice di ogni paio di scarpe è univoco.

In ogni momento il magazzino deve poter:

A. Fornire la propria capienza massima
B. Stabilire quante scarpe sono presenti
C. Stabilire l'ammontare del volume occupato da tutte le scatole di scarpe
D. Stabilire il peso di tutte le scatole di scarpe presenti
E. Aggiungere una nuova scarpa
F. Rimuovere un paio di scarpe
G. Dato un codice, restituire la scarpa associata
H. Fornire la propria locazione geografica

PARTE 2

Riflettere su: chi ha il compito di generare il codice?

Applicare il principio SOLID di Singola Responsabilità.

PARTE 3

Aggiungere la possibilità di immagazzinare ogni genere di merce (si supponga sempre confezionata in scatole). Per ogni tipo di merce si deve poter conoscere:

- Il codice
- Le dimensioni di ingombro
- La data di stoccaggio
- Il peso
- La tipologia (solido, liquido, gas)
- La misura (Kg, litri, m^3,…in base al tipo di merce)

Tutte le operazioni della PARTE 2 devono essere riviste prevedendo qualunque tipo di merce.

Modificare il punto C in questo modo:

C. Stabilire l'ammontare del volume occupato da tutte le merci e gli scaffali di supporto

PARTE 4

Aggiungere la possibilità di inviare tramite una POST al servizio http://store.my.warehouse.com le principali informazioni relative al magazzino:

- Locazione geografica
- Percentuale di riempimento (scaffali compresi)
- Numero complessivo di merci presenti