

Si intende realizzare una gerarchia di classi per rappresentare e gestire i seguenti tipi di oggetti:

- PC fissi, suddivisi in:
 - Desktop
 - Server
- PC portatili, suddivisi in:
 - Notebook
 - Palmari

Le caratteristiche generali di interesse, comuni a tutti i PC, sono:

- tipo di processore
- dimensione della memoria RAM
- dimensione della memoria di massa
- marca
- modello
- sistema operativo

1. PC fissi

I PC fissi sono caratterizzati, oltre agli attributi comuni, da:

- tipo di case (grande, medio, piccolo)

Per i PC fissi di tipo Desktop è necessario registrare:

- tipo di scheda video
- tipo di scheda audio

Per i PC fissi di tipo Server è necessario registrare:

- numero dei processori
- presenza o meno di dischi di tipo RAID

2. PC portatili

I PC portatili sono caratterizzati, oltre agli attributi comuni, da:

- peso

- dimensioni fisiche (altezza, larghezza, profondità)
- dimensione del video (in pollici)
- presenza o meno dell'interfaccia di rete wireless

Per i PC portatili di tipo Notebook è necessario indicare:

- presenza o meno di webcam
- in caso affermativo, la relativa risoluzione (ad es. in megapixel)

Per i PC portatili di tipo Palmare è necessario indicare:

- presenza o meno dell'interfaccia Bluetooth
- tipo di espansione della memoria di massa (es. SD, mini-SD, micro-SD, ...)

Parte A – Progettazione ad oggetti

1. Implementazione delle classi in Java

Implementare in linguaggio Java le classi progettate, specificando:

- costruttori (almeno uno completo per ogni classe concreta);
- metodi di accesso agli attributi (getter e, se opportuno, setter);
- il metodo `toString()` sovrascritto, in modo da fornire una rappresentazione testuale significativa dell'oggetto;
- il metodo `equals(Object o)` sovrascritto, in modo da confrontare due PC sulla base di un criterio ragionevole (ad es. marca, modello e tipo di PC, oppure altri attributi ritenuti appropriati).

Parte B – Gestione di un inventario di PC

Classe InventarioPC

Definire una classe `InventarioPC` che gestisca un insieme di oggetti di tipo PC.

La classe deve utilizzare una struttura dati adeguata (ad esempio `ArrayList<PC>`) e deve fornire almeno le seguenti funzionalità:

3.1. Inserimento e rimozione

- `void aggiungiPC(PC pc)`: aggiunge un PC all'inventario.

- boolean rimuoviPC(PC pc): rimuove il PC indicato, se presente, restituendo true in caso di rimozione avvenuta con successo, false altrimenti.

3.2. Ricerca semplice

- List<PC> cercaPerMarca(String marca): restituisce l'elenco di tutti i PC della marca specificata.
- List<PC> cercaPerSistemaOperativo(String sistemaOperativo): restituisce l'elenco di tutti i PC che eseguono il sistema operativo specificato.

3.3. Statistiche e calcoli

Implementare i seguenti metodi, che richiedono l'uso di cicli e condizioni:

- PC trovaPCConPiuRAM(): restituisce il PC (di qualunque tipo) con la quantità di RAM massima; in caso di parità è sufficiente restituirne uno qualsiasi tra quelli a RAM massima.
- double calcolaMediaRAM(): calcola e restituisce la quantità media di RAM dei PC presenti nell'inventario (se l'inventario è vuoto, gestire il caso in modo opportuno, ad esempio restituendo 0.0).
- int contaPortatiliConWifi(): restituisce il numero di PC portatili (notebook o palmari) che dispongono di interfaccia di rete wireless.
- int contaServerConRaid(): restituisce il numero di server dotati di dischi RAID.
- Notebook trovaNotebookPiuLeggero(): restituisce il notebook con peso minimo; se nell'inventario non sono presenti notebook, il metodo restituisce null.

Ricerca con più criteri

Ampliare la classe InventarioPC introducendo il seguente metodo:

- List<PC> cercaPortatili(double pesoMassimo, boolean richiedeWifi)

Il metodo deve restituire una lista contenente tutti i PC portatili (notebook e palmari) presenti nell'inventario che soddisfano contemporaneamente le seguenti condizioni:

- peso minore o uguale a pesoMassimo;
- se richiedeWifi == true, devono essere inclusi solo i portatili che dispongono di interfaccia di rete wireless;

- se richiedeWifi == false, il criterio relativo al Wi-Fi non deve essere considerato.

L'implementazione richiede di:

- scorrere la collezione di PC con un ciclo;
 - distinguere i portatili dagli altri PC (utilizzando ad esempio l'operatore instanceof o una gerarchia di classi ben progettata);
 - applicare condizioni multiple per verificare i vari vincoli.
-

Parte C – Classe di test

Classe TestPC

Codificare una classe TestPC contenente un metodo main che:

- istanzi almeno un oggetto per ciascuna delle tipologie di PC concrete previste (Desktop, Server, Notebook, Palmare);
- inserisca tali oggetti in un'istanza di InventarioPC;
- invochi tutti i metodi definiti nella classe InventarioPC, verificandone il funzionamento.

In particolare, il programma dovrà:

- stampare a video il PC con più RAM;
- stampare la media della RAM dei PC presenti;
- stampare il numero di portatili dotati di Wi-Fi;
- stampare il numero di server con dischi RAID;
- stampare il notebook più leggero, se presente;
- mostrare il risultato di alcune ricerche (ad esempio: tutti i PC di una certa marca, tutti i PC con un certo sistema operativo, tutti i portatili sotto un dato peso con/ senza Wi-Fi).