

Esame di Programmazione II, 2 febbraio 2015

La ditta Paranoid produce l'omonimo sistema operativo per una linea di smartphone su cui è possibile installare applicazioni (*app*). Esistono degli *store* che accumulano app e le rendono disponibili ai clienti. Nel corso degli anni, il sistema operativo Paranoid ha subito delle evoluzioni, per cui esistono già quattro versioni successive, in ordine cronologico:

```
public enum OS {
    LITTLE_LION, // prima versione
    RUNNING_TIGER,
    ACKWARD_BEAR,
    FLOATING_HIPPO // ultima versione
}
```

In Java, gli `enum` implementano `java.lang.Comparable`, per cui ad esempio gli elementi di `OS` si possono comparare con il metodo `int compareTo(OS other)`.

Esercizio 1 [5 punti] Si completi la seguente classe, che implementa un'app per il sistema Paranoid:

```
public class App implements Comparable<App> {
    ... aggiungete campi private
    public final OS MIN_OS;

    // minOS e' la versione minima di sistema operativo richiesta da questa app
    public App(String name, OS minOS, int version) {
        this.MIN_OS = minOS;
        ...
    }

    public String getName() { ...restituisce il nome dell'app }
    public int getVersion() { ...restituisce la versione dell'app }
    public String toString() { ...restituisce il nome dell'app seguito dalla sua versione }
    public boolean equals(Object other) { ...due app sono uguali se hanno uguale nome e versione }
    public int hashCode() { ...deve essere non banale }
    public int compareTo(App other) { ...due app vengono prima ordinate per nome, poi per versione }
}
```

Esercizio 2 [7 punti] Un dispositivo (smartphone) permette di installare un numero arbitrario di app, purché esse siano compatibili con la sua versione installata del sistema operativo Paranoid. È possibile reinstallare un'app già installata, ma solo se si passa a una versione successiva dell'app. In tal caso, la nuova versione sostituisce la vecchia versione della app. È sempre possibile deinstallare un'app.

Si completi la seguente classe, che implementa un dispositivo:

```
public class Device {
    public final OS os;
    ... eventuali campi privati

    // il dispositivo implementa il sistema operativo indicato e all'inizio non ha app installate
    public Device(OS os) {
        this.os = os;
    }

    // aggiunge l'app a quelle gia' installate, se possibile:
    // * se il sistema operativo del dispositivo non supporta l'app, lancia una AppNotSupportedException
    // * se l'app e' gia' installata in una versione non precedente, lancia una AppAlreadyInstalledException
    // * se l'app e' gia' installata in una versione precedente, la sostituisce con la nuova versione
    public void install(App app) { ... }

    // rimuove l'app da quelle gia' installate. Se non era installata, non fa nulla
    public void deInstall(App app) { ... }

    // restituisce le app installate su questo dispositivo
    public SortedSet<App> getInstalledApps() { ... }
}
```

Suggerimento: può tornare utile la classe `java.util.TreeSet<E>`, che implementa un insieme ordinato di `E` (che deve essere un `Comparable`) e che implementa l'interfaccia `java.util.SortedSet<E>`.

Esercizio 3 [2 punti] Sia `AppNotSupportedException` che `AppAlreadyInstalledException` sono sottoclassi concrete dell'eccezione astratta `InstallationException`, sottoclasse di `java.lang.RuntimeException`. Si scrivano queste tre classi di eccezione.

Esercizio 4 [8 punti] Uno store contiene app e ha l'intelligenza per selezionare quelle compatibili con un dato dispositivo. Si completi la seguente classe che implementa uno store:

```
public class Store {
    private final String name;
    ... eventuali campi private

    public Store(String name) {
        this.name = name;
    }

    public void add(App app) { ... aggiunge un'app a questo store }

    public String toString() {
        ... restituisce il nome dello store seguito dal toString() delle app contenute in questo store
    }

    public SortedSet<App> getAppsFor(Device device) {
        ... restituisce le app di questo store compatibili con il sistema operativo del dispositivo indicato.
        Se piu' versioni sono compatibili col dispositivo, restituisce solo la versione piu' recente
    }
}
```