

# Università Degli Studi dell'Aquila

Il appello di Laboratorio di Algoritmi e Strutture Dati - A.A. 2022/2023

Martedì 7 febbraio 2023 – Dott.ssa Giovanna Melideo (Durata: 1:30 h)

Svolgere i seguenti esercizi avendo come riferimento il linguaggio JAVA.

## ESERCIZIO 1

Si vuole gestire un archivio elettronico per raccogliere i dati relativi a tutti i pazienti ricoverati in alcune strutture ospedaliere. Ogni Ospedale è univocamente identificato da un codice alfanumerico. Si consideri la seguente classe **Paziente** (**cartella “Il appello” del Team di classe**) che rappresenta un paziente ricoverato. Di ogni paziente vengono archiviate le seguenti informazioni: nome, cognome, CF, età.

Realizzare una classe **Ricoveri** come una mappa che associa ad ogni Ospedale (codice di tipo **String**, chiave univoca) l'insieme di pazienti (**Set<Paziente>**) in esso ricoverati, fornendo le seguenti funzionalità:

- a) Ricerca dell'Ospedale in cui è ricoverato un dato paziente (**null** se assente):

```
public String cerca (Paziente p);
```

- b) Ricovero: inserimento di un paziente nell'insieme associato al dato Ospedale (se il paziente è già ricoverato in qualche Ospedale l'operazione fallisce; se l'Ospedale specificato in input non esiste il metodo lo inserisce):

```
public boolean ricovera(Paziente paziente, String Codice_Ospedale);
```

- c) Dimissione: cancellazione di un paziente dall'insieme associato all'Ospedale di ricovero (se il paziente è ricoverato il metodo restituisce il codice ospedaliero di dimissione, altrimenti **null**):

```
public String dimetti(Paziente paziente);
```

- d) Elenco dei ricoverati: creazione di un nuovo insieme contenente tutti i pazienti ricoverati, in ordine crescente di età:

```
public Set<Paziente> elencoRicoverati ();
```

- e) Selezione per età: estrazione di tutti i pazienti ricoverati aventi l'età specificata in input, in ordine alfabetico crescente per cognome-nome:

```
public Set<Paziente> elencoByEta (int eta);
```

## ESERCIZIO 2

Un albero di interi si definisce “fair” se tutti i figli destri dei nodi sono pari e tutti i figli sinistri sono dispari. La radice dell'albero può contenere qualsiasi valore in quanto non è né figlio destro né sinistro. Realizzare il metodo

```
static boolean fairBTree(LinkedBinaryTree<Integer> t)
```

che dato in input un albero binario di interi verifica se l'albero è “fair”.

## ESERCIZIO 3

Aggiungere alla classe **Network<Vertex>** un nuovo metodo costruttore

```
public Network (Set<Edge<Vertex>> graph)
```

che prende in input un grafo orientato pesato rappresentato come insieme di oggetti di tipo **Edge** e costruisce il medesimo grafo rappresentato come un oggetto di tipo **Network** (Si consideri la classe semplificata **Edge** contenuta nella **cartella “Il appello” del Team di classe**).