

NOME: _____

COGNOME: _____

MATRICOLA: _____

Un programmatore sta ristrutturando una applicazione per la gestione di consorzi di imprese. Per ogni consorzio ed ogni impresa devono essere gestite le seguenti informazioni: nome, nazione (della sede legale), un insieme di dipendenti; i consorzi inoltre gestiscono, in una opportuna collezione, l'insieme delle imprese consorziate. Le classi **Consorzio** e **Impresa** (riportate nel codice allegato) rappresentano oggetti di questo tipo. Si noti che nella classe **Consorzio** la collezione delle imprese consorziate è rappresentata dalla mappa **imprese**, che ha per chiave il nome di una impresa consorziata, per valore un riferimento all'oggetto che rappresenta l'impresa consorziata. Si suppone che il nome di un consorzio e di una impresa sia univoco (non possono esistere consorzi e/o imprese con lo stesso nome).

DOMANDA 1 (5%)

Scrivere il codice del metodo **public void aggiungiConsorziata(Impresa impresa)** della classe **Consorzio**.

DOMANDA 2

Un progettista esperto fa notare al programmatore che le singole imprese che appartengono ad un consorzio potrebbero a loro volta essere consorzi. Per tenere in considerazione questa possibilità, il progettista suggerisce di introdurre una classe astratta **Società**, come astrazione dei concetti di consorzio e impresa, e di modificare le classi **Consorzio** e **Impresa** affinché estendano correttamente tale classe astratta.

- (15%) Scrivere il codice della classe astratta **Società** e riscrivere completamente il codice delle classi **Consorzio** e **Impresa** affinché estendano correttamente la classe astratta **Società**
- (20%) nella scrittura del metodo **getNumeroDipendenti()** si consideri che il numero di dipendenti di un consorzio deve essere calcolato come la somma del numero dei propri dipendenti più il numero dei dipendenti di tutte le società consorziate.

DOMANDA 3 (20%)

Introdurre in maniera opportuna nelle classi **Società**, **Consorzio**, **Impresa** definite nel punto precedente il metodo **public Map<Integer, List<Società>> nazione2consorziate()**: questo metodo restituisce una mappa che ha per chiave una nazione, e per valore una lista contenente le società consorziate con sede nella stessa nazione espressa della chiave; tale lista deve essere inoltre ordinata per anno di costituzione.

Ad esempio, supponendo che il consorzio **PIPPO**, costituito nel 2000, con sede legale in **ITA**, sia composto dalle seguenti società:

NOME IMPRESA	ANNO COSTITUZIONE	NAZIONE
Minni	2000	FRA
Paperino	2002	SPA
QuiQuoQua	2002	FRA

Dove **QuiQuoQua** è a sua volta un consorzio costituito dalle società:

Ciccio	2005	ITA
Gastone	2005	USA

il metodo **nazione2consorziate()**, invocato sull'oggetto che rappresenta il consorzio **PIPPO** restituirebbe la mappa (per chiarezza, indichiamo il nome delle imprese al posto dei riferimenti ai corrispondenti oggetti)

```
ITA → {Pippo, Ciccio}
FRA → {Minni, QuiQuoQua}
SPA → {Paperino}
USA → {Gastone}
```

NB: le liste contenute nella mappa restituita sono ordinate per anno di costituzione.

CONTINUA SUL RETRO

DOMANDA 4 (20%)

Scrivere una classe di test per la classe **Consorzio**. In particolare, scrivere almeno 4 test case per testare la correttezza del metodo **getNumeroDipendenti()** ed almeno 4 per testare la correttezza del metodo **nazione2consorziate()**. In particolare, la correttezza dei due metodi va verificata nel caso di:

- (i) consorzio vuoto,
- (ii) consorzio composto da una singola impresa,
- (iii) consorzio composto da due imprese con sede in nazioni diverse,
- (iv) consorzio composto da due imprese con sede nella stessa nazione.

DOMANDA 5 (20%)

Data la seguente porzione di codice, disegnare un diagramma che rappresenti lo stato degli oggetti referenziati dalle variabili **elencoSocietà**, **s0**, **s1**, **s2** nel punto marcato con il commento **// ******

Non è necessario rappresentare gli oggetti referenziati dalle variabili di istanza **imprese** e **dipendenti**.

```
public static void main(String[] args) {
    List<Società> elencoSocietà = new ArrayList<ElencoSocietà>();

    Società s0 = new Impresa(2000, "ITA", "qui");
    Società s1 = new Consorzio(2001, "FRA", "paperopoli");
    elencoSocietà.add(s0);
    elencoSocietà.add(s1);
    elencoSocietà.add(new Consorzio(2002, "USA", "topolonia"));
    s0 = s1;
    s1 = new Impresa(1903, "SPA", "paperone");
    elencoSocietà.add(s0);
    elencoSocietà.add(s1);
    // ****
}
```

```

import java.util.*;
public class Consorzio implements Comparable<Consorzio> {
    private int annoDiCostituzione;
    private String nazione;
    private String nome;
    private Set<Dipendente> dipendenti;
    private Map<String, Impresa> imprese;

    public Consorzio(int annoCostituzione, String nazione, String nome) {
        this.annoDiCostituzione = annoCostituzione;
        this.nazione = nazione;
        this.nome = nome;
        this.dipendenti = new HashSet<Dipendente>();
        this.imprese = new HashMap<String, Impresa>();
    }

    public int getAnnoDiCostituzione(){ return this.annoDiCostituzione;}

    public String getNazione() { return this.nazione; }

    public String getNome() { return this.nome; }

    public void aggiungiConsorziata(Impresa impresa) {
        /* scrivere il codice di questo metodo (vedi DOMANDA 1) */
    }

    public void aggiungiDipendente(Dipendente dipendente) {
        this.dipendenti.add(dipendente);
    }

    public int getNumeroDipendenti () {
        // scrivere il codice di questo metodo (vedi DOMANDA 2)
    }

    public int compareTo(Consorzio that) {
        return this.getNome().compareTo(that.getNome());
    }
}

```

```

import java.util.*;
public class Impresa implements Comparable<Impresa> {
    private int annoDiCostituzione;
    private String nazione;
    private String nome;
    private Set<Dipendente> dipendenti;

    public Impresa(int annoDiCostituzione, String nazione, String nome) {
        this.annoDiCostituzione = annoDiCostituzione;
        this.nazione = nazione;
        this.nome = nome;
        this.dipendenti = new HashSet<Dipendente>();
    }

    public int getAnnoDiCostituzione() { return this.annoDiCostituzione; }

    public String getNazione() { return this.nazione; }

    public void aggiungiDipendente(Dipendente dipendente) {
        this.dipendenti.add(dipendente);
    }

    public String getNome() { return this.nome; }

    public int compareTo(Impresa that) {
        return this.getNome().compareTo(that.getNome());
    }

    public int getNumeroDipendenti() {
        // scrivere il codice di questo metodo (vedi DOMANDA 2)
    }
}

```

```

public class Dipendente {
    // codice omissso perché irrilevante ai fini del compito
}

```