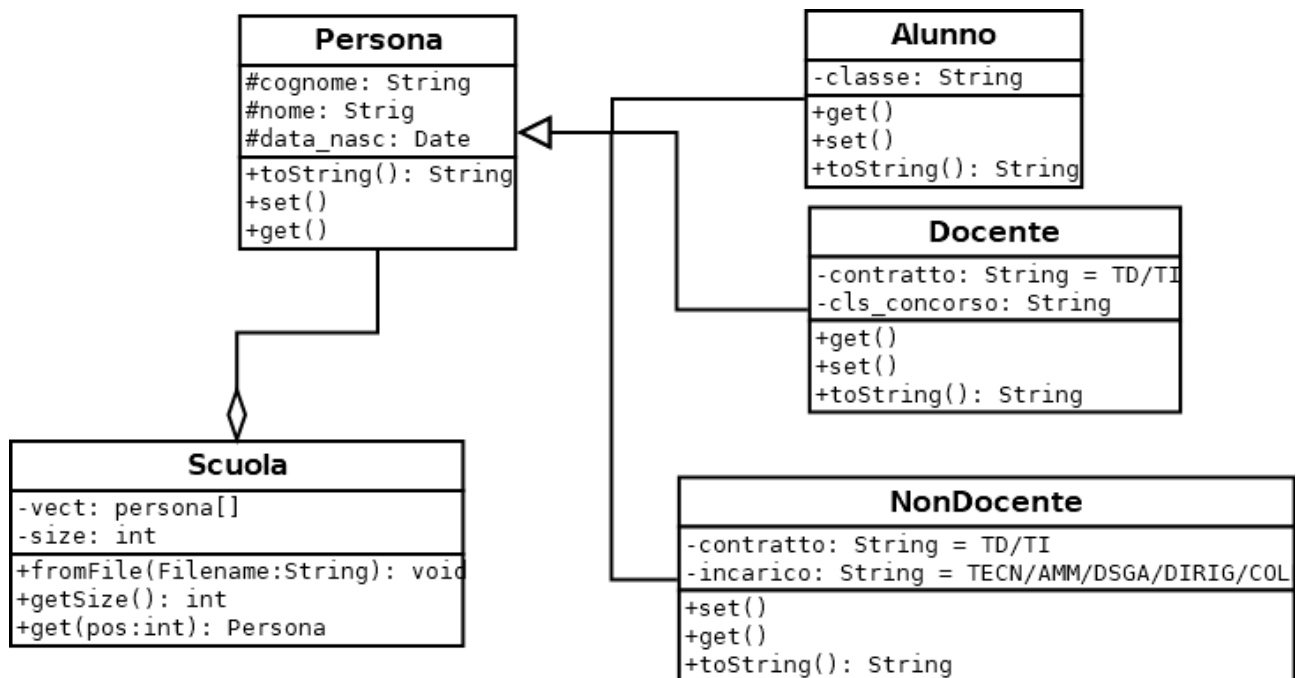


Scuola
Alunni Docenti Non Docenti



```

public class Persona {
    protected String cognome,    nome;
    protected Date data_nasc;

    public Persona(String cognome, String nome, Date d_nasc) {
        this.cognome = cognome;
        this.nome = nome;
        this.data_nasc = d_nasc;
    }
    public Persona(String cognome, String nome, String d_nasc) {
        this.cognome = cognome;
        this.nome = nome;
        try {
            this.data_nasc = new
SimpleDateFormat("dd/MM/yyyy").parse(d_nasc);
        } catch (ParseException ex) {
            this.data_nasc= new Date(0); //1 1 1970
        }
    }
    get e set ....
    @Override
    public String toString() {
        DateFormat df = new SimpleDateFormat("MM/dd/yyyy");
        return cognome+","+ nome+","+ df.format(data_nasc);
    }
}
  
```

```

public class Alunno extends Persona{

    private String classe;
  
```

```

        public Alunno(String cognome, String nome, Date d_nasc, String
classe) {
            super(cognome,nome,d_nasc);
            this.classe = classe;
        }
        public Alunno(String cognome, String nome, String d_nasc,
String classe) {
            super(cognome,nome,d_nasc);
            this.classe = classe;
        }

        get e set ...
        @Override
        public String toString() {
            return super.toString()+"," + classe ;
        }
    }
}

```

1> Completare le classi Persona e Alunno con i metodi set e get, Scrivere nelle classi stesse un main per provare il metodo toString

2> Aggiungere le classi Docente e NonDocente con gli attributi corretti.

```

public class Scuola {

    private Persona[] vect;
    private int size; // numero elementi effettivamente inseriti

    public int getSize() {
        return size;
    }

    public boolean add(Persona p) {
        if (vect==null)
            vect=new Persona[20]; // strategia di allocazione
arbitraria
        if (size < vect.length) {
            vect[size] = p;
            size++;
            return true;
        } else {
            return false;
        }
    }

    Persona get(int pos) {
        if (pos >= 0 && pos < size) {
            return vect[pos];
        } else {
            return null;
        }
    }
}

```

La classe scuola rappresenta una aggregazione di persone, si può aggiungere una persona ottenere l'i-esima persona e sapere quante persone sono effettivamente presenti
Con il codice fornito l'inserimento fallisce quando la struttura è piena.

3> Aggiungere il metodo toString che rappresenta l'intera struttura

4> Aggiungere il metodo int indexOf(String s) che permette di ottenere la posizione di una persona che nel nome o nel cognome contengono s, -1 se non esiste.

5> (facoltativo) Cambiare add in modo che la struttura si adatti alle richieste di inserimento.

6> Aggiungere il metodo remove(int pos) rimuove dalla struttura la persona di posizione pos

il seguente metodo :

```
void fromFile(String fn) {
    vect = new Persona[2000]; // scelta un po' arbitraria
    size = 0;
    BufferedReader bw;
    try {
        bw = new BufferedReader(new FileReader(fn));
        String linea;
        linea = bw.readLine();
        int rig = 0;
        while ((linea = bw.readLine()) != null) {
            String campi[] = linea.split(",");
            if (campi[0].equalsIgnoreCase("A")) { // è un
alunno
                Alunno a = new Alunno(campi[1], campi[2],
campi[3], campi[4]);
                add(a);

            }
        }

        bw.close();
    } catch (FileNotFoundException ex) {

Logger.getLogger(Scuola.class.getName()).log(Level.SEVERE, null,
ex);

    } catch (IOException ex) {

Logger.getLogger(Scuola.class.getName()).log(Level.SEVERE, null,
ex);

    }
}
}
```

Legge dal un csv righe e usando il costruttore di default aggiunge un alunno,

```
Alunno a = new Alunno(bla,bla,bla,bla);
add(a);
```

7> Usando i costruttori di Docente e NonDocente modificare il metodo in modo che tutte le righe siano lette e aggiunte.

8> Aggiungere i metodo numAlunni(), numDocenti(), numNonDocenti()

9> Aggiungere i metodi getAlunni(), getDocenti() getNnDocenti() che restituiscuo vettori di Alunni Docenti etc.

.