

## Esercizio1

### Paziente

```
package Esercizio1;

public class Paziente implements Comparable<Paziente>{
    String codiceFiscale;
    int eta;
    int reparto;

    public String getCodiceFiscale() {
        return codiceFiscale;
    }
    public void setCodiceFiscale(String codiceFiscale) {
        this.codiceFiscale = codiceFiscale;
    }
    public int getEta() {
        return eta;
    }
    public void setEta(int eta) {
        this.eta = eta;
    }
    public int getReparto() {
        return reparto;
    }
    public void setReparto(int reparto) {
        this.reparto = reparto;
    }
    public Paziente(String cf, int e, int r){
        codiceFiscale = cf;
        eta = e;
        reparto = r;
    }
}
```

```

@Override
public boolean equals(Object obj){
    if(obj == null || !(obj instanceof Paziente))
        return false;

    Paziente p = (Paziente) obj;

    return this.codiceFiscale.equals(p.getCodiceFiscale()) && this.eta == p.getEta() && this.reparto ==
p.getReparto();
}

@Override
public String toString(){
    return "Codice fiscale: " + codiceFiscale + "Età: " + eta + "Reparto: " + reparto;
}

@Override
public int compareTo(Paziente obj){
    return this.codiceFiscale.compareTo(obj.getCodiceFiscale());
}
}

```

### Archivio

```

package Esercizio1;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.Iterator;
import java.util.List;

public class Archivio{

    List<Paziente> lista = new ArrayList<Paziente>();
}

```

```
public Archivio(){  
}
```

// Punto 1

```
public boolean insert(String cf, int e, int r){  
    Paziente p = new Paziente(cf, e, r);  
    for(Paziente pa : lista){  
        if(p.equals(pa))  
            return false;  
    }  
    lista.add(p);  
    return true;  
}
```

// Punto 2

```
public void deleteByCf(String cf){  
    Iterator<Paziente> it = lista.iterator();  
    while(it.hasNext()){  
        if(it.next().getCodiceFiscale().equals(cf)){  
            it.remove();  
            return;  
        }  
    }  
}
```

// Punto 3

```
public Archivio selByRepart(int r){  
    Archivio ret = new Archivio();  
  
    for(Paziente pa : lista){  
        if(pa.getReparto() == r){  
            ret.insert(pa.getCodiceFiscale(), pa.getEta(), pa.getReparto());  
        }  
    }  
    return ret;  
}
```

```
}
```

```
// Punto 4
```

```
public void sortByCf(){  
    Collections.sort(lista);  
}
```

```
// Punto 5
```

```
public void sortByEta(){  
    Collections.sort(lista,new Comparator<Paziente>() {  
        @Override  
        public int compare(Paziente p1, Paziente p2){  
            if(p1.getEta() > p2.getEta())  
                return 1;  
            if(p1.getEta() == p2.getEta())  
                return 0;  
            return -1;  
        }  
    });  
}
```

```
// Punto 6
```

```
public int[] numeroPazienti(){  
    int[] array = new int[10];  
  
    for(Paziente pa : lista){  
        array[pa.getReparto()-1]++;  
    }  
    return array;  
}  
}
```

## Esercizio 2

```
package Esercizio2;  
  
import java.util.ArrayList;  
import java.util.Collections;
```

```

public class Esercizio2 {

    //Prima versione

    public static ArrayList<String> differenzaSimmetrica(ArrayList<String> lista1, ArrayList<String> lista2){

        ArrayList<String> lista = new ArrayList<String>();

        for(String s : lista1){
            lista.add(s);
        }

        for(String s : lista2){
            lista.add(s);
        }

        Collections.sort(lista);

        for(int i=0; i<lista.size(); i++){

            if(lista.get(i).equals(lista.get(i+1))){

                lista.remove(lista.get(i));

                lista.remove(lista.get(i+1));

                i--;

            }

        }

        return lista;

    }

    //Seconda versione (metodo generico)

    public static <T extends Comparable<? super T>> ArrayList<T> differenzaSimmetrica(ArrayList<T> lista1,
    ArrayList<T> lista2){

        ArrayList<T> lista = new ArrayList<T>();

        for(T s : lista1){
            lista.add(s);
        }

        for(T s : lista2){
            lista.add(s);
        }
    }
}

```

```
}  
Collections.sort(lista);  
for(int i=0; i<lista.size(); i++){  
    if(lista.get(i).equals(lista.get(i+i))){  
        lista.remove(lista.get(i));  
        lista.remove(lista.get(i+1));  
        i--;  
    }  
}  
return lista;  
}  
}
```