

Esercizio 1

CLIENTE

```
public class Cliente implements Comparable<Cliente>{

    private String nominativo;

    private String codiceFiscale;

    private String residenza;


    Cliente(String nominativo, String codiceFiscale, String residenza){

        this.nominativo = nominativo;

        this.codiceFiscale = codiceFiscale;

        this.residenza = residenza;

    }


    Cliente(String codiceFiscale){

        this.nominativo = "Unknown";

        this.codiceFiscale = codiceFiscale;

        this.residenza = "Unknown";

    }


    public String getNominativo() {

        return nominativo;

    }


    public void setNominativo(String nominativo) {

        this.nominativo = nominativo;

    }


    public String getCodiceFiscale() {

        return codiceFiscale;

    }


    public void setCodiceFiscale(String codiceFiscale) {
```

```
    this.codiceFiscale = codiceFiscale;
}
```

```
public String getResidenza() {
    return residenza;
}
```

```
public void setResidenza(String residenza) {
    this.residenza = residenza;
}
```

```
@Override
```

```
public boolean equals(Object obj){
    if(obj == null || !(obj instanceof Cliente))
        return false;

    Cliente c = (Cliente) obj;

    return nominativo.equals(c.getNominativo()) && codiceFiscale.equals(c.getCodiceFiscale()) &&
    residenza.equals(c.getResidenza());
}
```

```
@Override
```

```
public String toString(){
    return "Nominativo: " + nominativo + "Codice Fiscale: " + codiceFiscale + "Residenza: " + residenza;
}
```

```
@Override
```

```
public int compareTo(Cliente obj){
    return this.codiceFiscale.compareTo(obj.getCodiceFiscale());
}
}
```

AUTOMOBILE

```
class Automobile implements Comparable<Automobile>{

    private String targa;

    private Cliente proprietario;

    public Automobile(String targa, Cliente proprietario) {

        this.targa = targa;

        this.proprietario = proprietario;

    }

    public String getTarga() {

        return targa;

    }

    public void setTarga(String targa) {

        this.targa = targa;

    }

    public Cliente getProprietario() {

        return proprietario;

    }

    public void setProprietario(Cliente proprietario) {

        this.proprietario = proprietario;

    }

    public boolean equals(Object obj){

        if(obj == null || !(obj instanceof Automobile))

            return false;

        Automobile a = (Automobile) obj;

        return this.targa.equals(a.getTarga()) && this.proprietario.equals(a.getProprietario());

    }

}
```

```
}
```

```
public String toString(){  
    return "Targa: " + targa + "; Proprietario: " + proprietario;  
}
```

```
public int compareTo(Automobile a){  
    return targa.compareTo(a.getTarga());  
}  
}
```

ARCHIVIO

```
public class Archivio {  
  
    private List<Automobile> lista = new ArrayList<Automobile>();  
  
    public Archivio(){  
  
    }  
  
    public void insert(String targa, Cliente cliente){  
        Automobile a = new Automobile(targa, cliente);  
        for(Automobile at : lista){  
            if(a.equals(at)){  
                at.setProprietario(cliente);  
                return;  
            }  
        }  
        lista.add(a);  
    }  
  
    public boolean deleteByTarga(String targa){  
        Iterator<Automobile> iterator = lista.iterator();
```

```

while(iterator.hasNext())
    if(iterator.next().getTarga().equals(targa)){
        iterator.remove();
        return true;
    }
return false;
}

```

```

public Archivio nuovoArchivio(String citta){
    Archivio less = new Archivio();
    for(Automobile at : lista){
        if(at.getProprietario().getResidenza().equals(citta))
            less.insert(at.getTarga(),at.getProprietario());
    }
    return less;
}

```

```

public void SortByTarga(){
    Collections.sort(lista);
}

```

```

public void SortByCF(){
    Collections.sort(lista,new Comparator<Automobile>() {
        public int compare(Automobile a1, Automobile a2){
            return a1.getProprietario().compareTo(a2.getProprietario());
        }
    });
}

```

```

public void Stampa(){
    for(Automobile at : lista){
        System.out.println(at.toString());
    }
}

```

```
}  
}
```

Esercizio2

```
public class Esercizio2 {  
    public static <T> void riordina(T[] array){  
        T temp;  
        for(int i = 0; i < array.length/2; i++){  
            temp = array[i];  
            array[i] = array[array.length - i];  
            array[array.length - i] = temp;  
        }  
    }  
}
```