

## Esercitazione 3 – Cloneable, Contenitori & Generics

- Modificare la classe *Example* come di seguito riportato:

```
public class Example implements Iterable<Double>{
    private List<Double> example;
    Example ()
    {
        example=new LinkedList<>();
    }
    public Iterator<Double> iterator() {
        return example.iterator();
    }
    // si noti che il metodo "set" è stato rimpiazzato dal metodo "add"
    void add(Double v) {
        example.add(v);
    }

    Double get(int index) {
        return example.get(index);
    }...
}
```

Modificare di conseguenza tutti metodi che fanno riferimento all'attributo *example*.

Usare l'iteratore per scandire l'attributo *example* nel metodo *distance*

Modificare la classe *Data* per risolvere gli errori

- Modificare la classe *Data* come di seguito riportato:

```
public class Data {
    private List<Example> data=new ArrayList<>();
    ...
}
```

Modificare la classe di conseguenza e riscrivere il metodo `public String toString(){...}` usando l'iteratore per scandire l'attributo *data*.

- Modificare la classe *Cluster* come di seguito riportato:

```
public class Cluster implements Iterable<Integer>, Cloneable{
    private Set<Integer> clusteredData=new TreeSet<>();
    ...
}
```

Modificare la classe *Cluster* di conseguenza

Rimuovere `public int getElement(int i){...}`

Rimpiazzare il metodo

`Cluster createACopy() { .. }`

Con il metodo

`public Object clone() {}`

che deve essere opportunamente usato per realizzare la clonazione di un oggetto istanza di cluster. Il metodo clone deve essere usato nel progetto laddove era usato il metodo createCopy

Modificare l'implementazione di `Cluster mergeCluster (Cluster c){...}` e `String toString(Data data){...}` usando gli iteratori

Modificare le implementazioni di *SingleLinkDistance* e *AverageLinkdistance* usando gli iteratori per scandire i cluster.

Eseguendo il main l'output del progetto non è cambiato!