

## Esercizio 2

(a) Si scrivano in Java le definizioni di una classe `tupleJ`, estesa in una classe `coppiaJ` a sua volta estesa in una classe `ternaJ`. Le tre classi hanno un comportamento analogo ai tipi astratti `coppiaOfT`, `ternaOfT`, `tupleOfT`, visti in esercizio 1, in particolare devono nascondere lo stato di implementazione ed esportare le sole operazioni necessarie per calcolare la funzione `fun_cOfT`. In più hanno un'operazione `size` che vale il numero di componenti degli oggetti della specifica classe.

(b) Si completino le seguenti definizioni per due oggetti dell'interfaccia funzioni `fun2T`, so

```
public interface fun2T <T>{
    public T apply(T x, T y);
}
public static class main{
    public <T> T fun_cOfT (fun2T<T> f, tupleJ<T> r)throws IllegalArgumentException{...};
    public static void main(){
        fun2T<Integer> somma = new fun2T<Integer> (){
            public integer apply(integer x, integer y){return x+y;}
        }
        fun2T<Integer> minimo = new fun2T<Integer> (){
            public integer apply(integer x, integer y){...}
        }
        CoppiaJ<Integer> c = ...;
        TernaJ<Integer> t = ...;
        Integer r1 = ... ;
        Integer r2 = ... ;
    }
}
```