

21. Esercitazione

IS 2024-2025



Laura Semini, Jacopo Soldani

Corso di Laurea in Informatica

Dipartimento di Informatica, Università of Pisa

SU COSA CI ESERCITIAMO?

Il metodo `finePasto()` aggiorna lo stato dei punti come segue:

- ogni commensale che ha mangiato senza lavare i piatti perde un punto
- chi ha lavato i piatti guadagna un numero di punti pari a $(\text{numero commensali} - \text{numero lavapiatti}) / \text{numero lavapiatti}$

```
public void finePasto(String[] commensali, String[]  
lavapiatti, StatoPunti stPunti) {  
1. int numeroLavapiatti = lavapiatti.length;  
2. int numeroCommensali = commensali.length;  
3. if (numeroCommensali == 0 || numeroLavapiatti == 0)  
4. throw new IllegalArgumentException("array vuoto");  
5. int puntiGuadagnati = (numeroCommensali -  
    numeroLavapiatti) / numeroLavapiatti;
```

```
6. for (String commensale : commensali) {  
7.     boolean haLavatoPiatti = false;  
8.     for (String sguattero: lavapiatti)  
9.         if (sguattero.equals(commensale)){  
10.             haLavatoPiatti = true;  
11.             break;}  
12.     if (haLavatoPiatti)  
13.         stPunti.aggiorna(commensale,  
            puntiGuadagnati);  
14.     else  
14.         stPunti.aggiorna(commensale,-1);  
    }  
}
```

Esercizi

1. Fornire una test obligation che si basa su criteri black box.
2. Definire 5 casi di test che soddisfano la test obligation fornita al punto 1
3. Disegnare il diagramma di flusso di finePasto() usando un opportuno diagramma UML.
4. Definire dei casi di test che garantiscano:
 1. La copertura dei comandi
 2. La copertura delle condizioni semplici
 3. La copertura dei cammini
5. Quali criteri di riduzione del numero di combinazioni ha senso applicare in questo caso?

Ex 3: soluzione



