

# СКИТ домашна 1

```
/**
 * Method that finds the employees that belong to only one of the
 * teams given as input parameters.
 *
 * @param team1 represents a set of all employee IDs
 *               for the employees that belong to team1
 * @param team2 represents a set of all employee IDs
 *               for the employees that belong to team2
 * @return the IDs of the employees that belong to only one of the teams
 */
public static Set<String> notInBothTeams(Set<String> team1,
                                         Set<String> team2) {
```

## Interface based

Влез во функцијата се два стринг сетови team1 и team2.

Ќе дефинираме по една карактеристика за секој сет која ќе се однесува на бројот на елементи во сетот. Па така добиваме две карактеристики K1: број на елементи во првиот сет односно team1. И K2: број на елементи во вториот сет односно team2. Понатаму карактеристиките ќе ги поделиме во 3 блокови и тоа празен сет (тука ќе спаќа и неиницијализиран односно null), сет со 1 елемент и сет со повеќе од 1 елементи.

Карактеристика	Блок 1	Блок 2	Блок 3
K1	team1==null or team1.isEmpty	team1.size==1	team1.size > 1
K2	team2==null or team2.isEmpty	team2.size==1	team2.size > 1

- A) Да партиципањето го исполува условот за дисјунктност затоа што еден сет во исто време не може да има повеќе елементи од 1 и точно 1 елемент и да нема елементи или да е null.
- B) Да својството на комплетност е исполнето затоа што еден сет единствено може да не биде иницијализиран или да има 0 елементи или да има 1 елемент или да има повеќе од 1 елемент.
- C) Основен тест би бил K1: блок 3 и K2: блок 3 односно  
 $K1K2 = (team1.size > 1, team2.size > 1)$   
BCC test requirements:
1.  $K1K2 = (team1.size > 1, team2.size > 1)$
  2.  $K1K2 = (team1.size > 1, team2.size == 1)$
  3.  $K1K2 = (team1.size > 1, team2 == null \text{ or } team2.isEmpty)$
  4.  $K1K2 = (team1.size == 1, team2.size > 1)$

$$5. \quad K1K2 = (team1 == null \text{ or } team1.isEmpty, team2.size > 1)$$

Infeasible TRs: нема

Вкупен број на тестови 5.

D) .

## functionality based

Функцијата ги враќа сите вработени кои што припаѓаат само во еден од двата тимови пратени како аргументи од ова може да се заклучи дека функцијата ќе зависи дали сетовите се преклопуваат односно дали имаат пресек или не исто така зависи и дали еден од сетовите е подмножество на другиот. Па од тука може да добиеме 3 карактеристики дополнително ќе ги дефинираме карактеристиките дали некој од тимовите е null или празен затоа што и таа карактеристика е значајна за функцијата. Па вкупно добиваме 5 карактеристики кои може да се поделат по два блока true и false односно:

Карактеристика	Блок 1	Блок 2
K1: $team1 == null \text{ or } team1.isEmpty$	True	False
K2: $team2 == null \text{ or } team2.isEmpty$	True	False
K3: $team1 \cap team2 == \emptyset$	True	False
K4: $team1 \subseteq team2$	True	False
K5: $team2 \subseteq team1$	True	False

- A) Да партиципирањето го исполува условот за дисјунктност затоа што во исто време не може еден сет да нема елементи и да има, или двата сетови да имаат пресек и да немаат, или еден сет да е подмножество на другиот и да не е.
- B) Да својството на комплетност е исполнето затоа што еден сет единствено може да нема елементи или да има, двата сетови единствено може да имаат пресек или да немаат, еден сет единствено може да биде подмножество на другиот сет или да не биде подмножество .
- C) Основен тест би бил тимовите да не се null тимовите да имаат пресек и еден тим да не е подмножество на друг тим односно  
Основниот тест би бил:  $K1K2K3K4K5 = FFFFF$ .

BCC test requirements:

- $K1K2K3K4K5 = FFFFF$ ;
- $K1K2K3K4K5 = TFFFF$ ;
- $K1K2K3K4K5 = FTFFF$ ;
- $K1K2K3K4K5 = FFTFF$ ;
- $K1K2K3K4K5 = FFFTF$ ;
- $K1K2K3K4K5 = FFFFT$ ;

Infeasible TRs: 2 (TFFFF) и 3 (FTFFF) не може истовремено еден сет да е null и двата сета да имаат пресек и да не е подмножество на другиот сет. Па така TR 2 и 3 може да ги замениме со FTTF и FTTF.

Конечни BCC test requirements:

1.  $K_1K_2K_3K_4K_5 = FFFFF$ ;
2.  $K_1K_2K_3K_4K_5 = TFTTF$ ;
3.  $K_1K_2K_3K_4K_5 = FTTFT$ ;
4.  $K_1K_2K_3K_4K_5 = FFTFF$ ;
5.  $K_1K_2K_3K_4K_5 = FFFTF$ ;
6.  $K_1K_2K_3K_4K_5 = FFFFT$ ;

Вкупен број на тестови 6.

D) .