

## Христина Секулоска

211236

```
/**
 * The method should act as follows:
 *     if ukVisaApplications or usaVisaApplications is null throw NullPointerException
 *     else, return a (non-null) list of applicants (applicant ids) that have applied both for UK and USA Visa
 *     else, if there are no such applicants return null
 */
1 usage
public static List<String> applicantsForBothVisas(List<String> ukVisaApplications,
                                                List<String> usaVisaApplications) {
```

За да изведување на тестирањето треба да се дефинираат карактеристики кои го опишуваат влезното множество податоци т.е влезниот домен. Овие карактеристики може да се дефинираат согласно влезните параметри неинтересирајќи не што прави самата функција и каков и е излезот, додека пак кога станува збор за тоа што прави функцијата, каков и е излезот и како се поврзани двата параметри меѓусебе ќе ги опишам подоле во **functionality based**.

**-Interface-based карактеристики:** Овие карактеристики треба да се однесуваат на влезните параметри на функцијата. Функцијата има 2 влезни параметри ukVisaApplications I usaVisaApplications па според тоа следните карактеристики би биле:

C1: ukVisaApplications не е null или usaVisaApplications не е null

C2: ukVisaApplications не е празна низа или usaVisaApplications не е празна низа

**- Functionality-based карактеристики:**

Како што кажав погоре ваквиот тип на карактеристики се однесуваат на поврзаноста помеѓу нив и како влијаат на крајниот излез на функцијата. Една таква карактеристика може да биде:

C3: листа од ukVisaApplications и usaVisaApplications коишто има апликанти кои што аплицирале за двете визи

Согласно сето ова, карактеристиките C1 и C2 ќе вратат точно/неточно (true/false) доколку постои барем една низа што е null .Нивното партиционирањето ќе биде на 2 блока - true (T) и false (F). Имено, можам да кажам дека овие изрази го

задоволуваат својството на дисјунктност т.е. еден услов не може истовремено да биде и точен и неточен, воедно го задоволуваат и својството на комплетност т.е. или е точно или е неточно.

Карактеристиката C3 има поделба на следните блокови:

C3.1: ниту еден елемент од ukVisaApplications и usaVisaApplications не е ист т.е. листата е празна

C3.2 : барем 1 елемент од ukVisaApplications и usaVisaApplications е ист т.е. листата не е празна

И со оваа карактеристика ги задоволуваме својството на дисјунктност и комплетност бидејќи или ќе немаме ни 1 елемент, или барем 1 елемент.

Со користењето на Base Choice Coverage (BCC) тестот ќе го покријам влезното множество на податоци, прво треба да изберам base choice. Согласно тестовите C1 C2 C3 base choice е сите карактеристики да се TRUE и happy path би бил T T C3.2

Имено, другите тестови ги добив на тој начин што секоја карактеристика ќе се менува во сите други можни блокови (вредности), а другите ќе бидат фиксирани. Според основниот случај што го избрав погоре, другите тестови би биле:

F T C3.2 - тест 2

T F C3.2 - тест 3

T T C3.1 - тест 4

Според горенаведените тестови треба да видиме дали има некои што се невозможни (infeasible):

- Прв невозможен тест е првиот F T C3.2 бидејќи не може истовремено некоја од низите да е null и да не е празна некоја од низите. Поради тоа треба да го замениме тестот F T C3.1 со карактеристиката C3.1 каде што кажува дека ниту еден елемент од ukVisaApplications и usaVisaApplications не е ист т.е. листата е празна. Но, со оваа промена се уште тестот е невозможен бидејќи не може една од низите да е null и истовремено ниту една од низите да не е празна. Оттука, повторно го менуваме тестот во F F C3.1 којшто е возможен.

-Втор невозможен тест е вториот T F C3.2 бидејќи не може истовремено да имаме една празна низа, а при нивна споредба да имаме барем 1 елемент што им е ист.

Според тоа овој тест го менуваме во T F C3.1 којшто ќе биде возможен, поточно имаме една празна низа и ниту еден елемент од низите не е ист.

Наредно треба уште да дефинираме вредности што одговараат за секој од тестовите:

● BC - тест 1 (T T C3.2):

ukVisaApplications = {1, 2, 3, 4},

usaVisaApplications = {3,4,5,6}

очекуван излез: {3, 4}

● Тест 2 (F F C3.1):

ukVisaApplications = null,

usaVisaApplications = null

очекуван излез: NullPointerException

● Тест 3 (T F C3.1):

ukVisaApplications = {},

usaVisaApplications = {1, 2, 3}

очекуван излез: {}

● Тест 4 (T T C3.1):

ukVisaApplications = {1, 2, 3, 4},

usaVisaApplications = {5, 6, 7, 8}

очекуван излез: {}

Согласно ова, имаме 4 теста коишто треба да ги бидат исполнети.