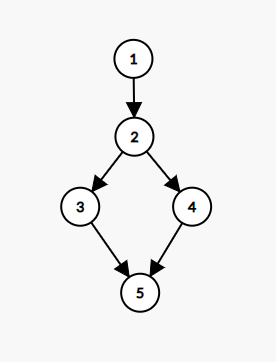
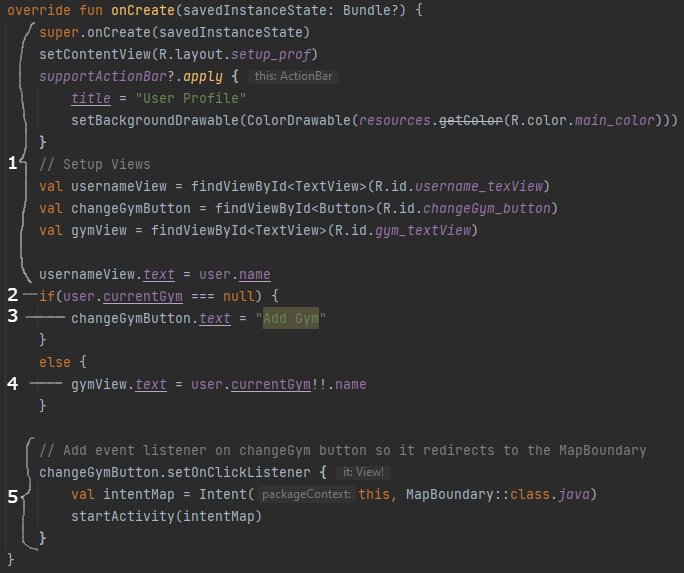
Στη κατηγορία τεχνικής white box, επιλέξαμε τη μέθοδο path coverage που περιγράφεται στα [1] και [2]. Παρακάτω περιγράφουμε το test case για το sequence «Ανάρτηση Βίντεο» ή στα αγγλικά “Upload Video”. Αρχικά παραθέτουμε τον κώδικα, σημειώνοντας πάνω σε αυτός τα κομμάτια εκείνα που θεωρούμε σαν κόμβους και τα σημειώνουμε με έναν αριθμό. Στα δεξιά παρατίθενται τα γραφήματα που προκύπτουν.

1. Test Case 1ο: Στην μέθοδο onCreate της κλάσης UserProfile



1

2

Υπολογισμός κυκλωματικής πολυπλοκότητας:

V(g) = |E| - |V| + 2 ⇒ V(g) = 5 – 5 + 2 ⇒ V(g) = 2

Υπολογισμός μέσω των περιοχών του G, φαίνονται με κόκκινη αρίθμηση:

V(g) = 2

Υπολογισμός μέσω των απλών συνθηκών στον κώδικα, φαίνονται με μπλε βέλη στον κώδικα:

V(g) = 1 + 1 = 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Μονοπάτι** | **Περίπτωση Ελέγχου**  **Τιμή Εισόδου User.currentGym** | **Περιγραφή** | **Αναμενόμενο αποτέλεσμα** |
| Ν1 | null | Έχει τη τιμή null | Το κουμπί του προφίλ έχει το κείμενο Add Gym |
| Ν2 | Υπάρχει valid Gym object | Δεν έχει τιμή null | Το κουμπί του προφίλ έχει το κείμενο Change Gym |

Με τα μονοπάτια να είναι:

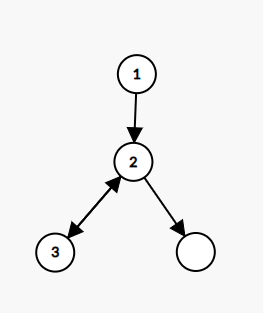
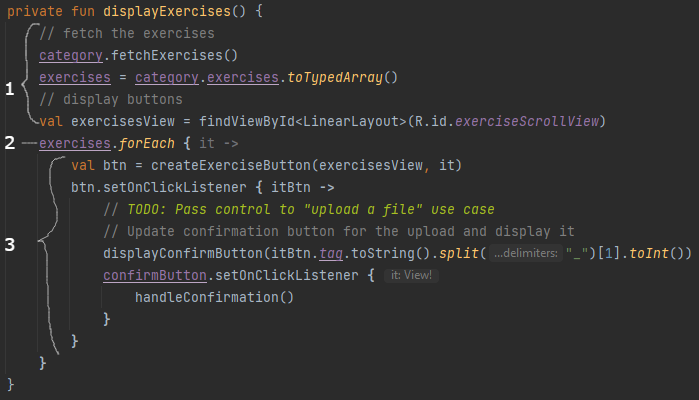
Ν1: 1-2-3-5

Ν2: 1-2-4-5

|  |  |
| --- | --- |
| **Μονοπάτι** | **Σ2 (User.currentGym == null)** |
| Ν1 | T |
| Ν2 | F |

Επομένως είναι προφανές ότι για να δοκιμαστεί η συνθήκη Σ2 πρέπει να βάλουμε στη μεταβλητή User.currentGym τη τιμή null και τη τιμή ενός valid Gym Object. Επομένως χρειαζόμαστε 2 περιπτώσεις ελέγχου.

1. Test case 2ο: Στη μέθοδο displayExercises της κλάσης ChooseExerciseBoundary



1

2

*Σημειώνουμε ότι ο κενός κόμβος υποδηλώνει τη λήξη της συνάρτησης*.

Υπολογισμός κυκλωματικής πολυπλοκότητας:

V(g) = 4 – 4 + 2 ⇒ V(g) = 2

Υπολογισμός μέσω περιοχών του γραφήματος:

V(g) = 2

Υπολογισμός μέσω των συνθηκών στον κώδικα:

V(g) = 2, το foreach υποθέτω λειτουργεί είτε με κάποιου είδους iterator, οπότε γίνεται ο έλεγχος it != it\_end, είτε μέσω ελέγχου ενός ακεραίου, είτε συγκρίνοντας κάποια integer μεταβλητή με το size του Array.

|  |  |  |  |
| --- | --- | --- | --- |
| **Μονοπάτι** | **Περίπτωση Ελέγχου**  **Τιμή Εισόδου exercises.size** | **Περιγραφή** | **Αναμενόμενο αποτέλεσμα** |
| Ν1 | 0 | To array exercises είναι άδειο | Το κουμπί του προφίλ έχει το κείμενο Add Gym |
| Ν2 | 10 | Το array exercises έχει θετικό αριθμό στοιχείων | Το κουμπί του προφίλ έχει το κείμενο Change Gym |

\*Σημειώνουμε επίσης ότι αρνητική τιμή δεν μπορεί ποτέ να εμφανιστεί αφού δεν μπορεί ποτέ να υπάρξει αρνητικό size ενός array. Όμοια, έχουμε ορίσει το attribute exercises ώστε να μην είναι nullable και το αρχικοποιούμε με ένα κενό array.

Όπου τα μονοπάτια είναι:

Ν1: 1-[2-3]exercises.size-2-(κενός κόμβος)

Ν2: 1-2-(κενός κόμβος)

Κάλυψη συνθηκών:

|  |  |
| --- | --- |
| **Μονοπάτι** | **Σ2 (i < exercises.size)** |
| Ν1 | F(exercises.size), T |
| Ν2 | T |

Διαπιστώνουμε λοιπόν ότι ελέγχοντας στο μονοπάτι N1, έχοντας δηλαδή μη κενό array ελέγχουμε και τις δύο περιπτώσεις.

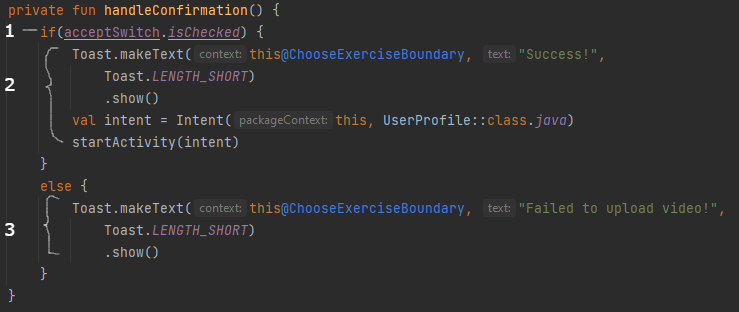
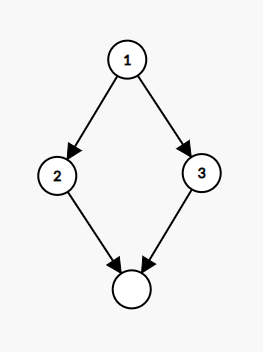
Κάλυψη βρόγχου: (θεωρούμε ότι το foreach είναι ένα for loop με απλούστερο συντακτικό)

Σύμφωνα με αυτή, βλέπε [1] διαφάνεια 75, πρέπει να επιλέξουμε τις περιπτώσεις ελέγχου που θα οδηγήσουν το loop να εκτελεστεί 0 φορές, το μονοπάτι Ν2 δηλαδή, μία φορά, οπότε exercises.size == 1, για μια μόνο άσκηση δηλαδή και περισσότερες από μια ασκήσεις. Αφού δεν γνωρίζουμε ποιος είναι ο μέγιστος αριθμός ασκήσεων που θα υπάρχουν σε μια κατηγορία, δεν μπορούμε να ελέγξουμε τις ακραίες περιπτώσεις Ν, Ν-1.

Οπότε:

|  |  |
| --- | --- |
| **Μονοπάτι** | **exercises.forEach{ … }** |
| Ν1 | 10 |
| Ν2 | 0 |
| Ν3 | 1 |

Επομένως θα εξετάσουμε και τα 3 μονοπάτια Ν1, Ν2, Ν3, εφόσον θέλουμε να κάνουμε έλεγχο βρόγχου.

1. Test case 3ο: Στη μέθοδο handleConfirmation της κλάσης ChooseExerciseBoundary

1

2

Υπολογισμός κυκλωματικής πολυπλοκότητας:

V(g) = |E| - |V| + 2 ⇒ V(g) = 4 – 4 + 2 ⇒ V(g) = 2

Υπολογισμός μέσω των περιοχών του G:

V(g) = 2

Υπολογισμός μέσω των απλών συνθηκών στον κώδικα:

V(g) = 1 + 1 = 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Μονοπάτι** | **Περίπτωση Ελέγχου**  **Τιμή Εισόδου**  **acceptSwitch.isChecked** | **Περιγραφή** | **Αναμενόμενο αποτέλεσμα** |
| Ν1 | true | Έχει τη τιμή true | Το σύστημα δέχεται τα αρχεία που ανέβασε ο Αθλητής |
| Ν2 | false | Έχει τη τιμή false | Το σύστημα απορρίπτει τα αρχεία που ανέβασε ο Αθλητής |

Με τα μονοπάτια να είναι:

Ν1: 1-2-(κενός κόμβος)

Ν2: 1-3-(κενός κόμβος)

|  |  |
| --- | --- |
| **Μονοπάτι** | **Σ1 (acceptSwitch.isChecked)** |
| Ν1 | T |
| Ν2 | F |

Στη συνθήκη Σ1 πρέπει να ελέγξουμε τόσο τη τιμή true όσο και τη τιμή ενός false. Επομένως χρειαζόμαστε 2 περιπτώσεις ελέγχου.

# Αναφορές

|  |  |
| --- | --- |
| [1] | M. Ξένος. (2022). Διάλεξη 12 [PowerPoint slides]. |
| [2] | Α. Ηλίας (2022). Φροντιστήριο 11 [PowerPoint slides]. |