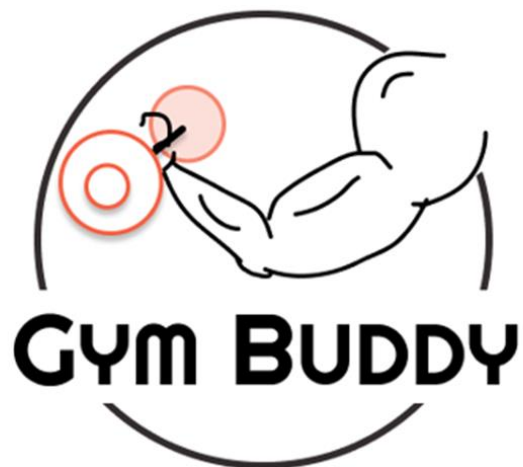


# Test Cases v0.1

GymBuddy



### *Σύνθεση ομάδας*

*Παρακάτω αναφέρονται τα στοιχεία των μελών της ομάδας μας:*

- *ΓΙΑΝΝΑΚΗΣ ΕΜΜΑΝΟΥΗΛ ΔΗΜΗΤΡΙΟΣ, 1067491, Δ' ΕΤΟΣ*
- *ΚΑΒΟΥΛΑΣ ΑΛΕΞΑΝΔΡΟΣ, 1067498, Δ' ΕΤΟΣ*
- *ΜΗΛΙΩΝΗΣ ΒΑΣΙΛΕΙΟΣ, 1067415, Δ' ΕΤΟΣ*
- *ΜΠΟΛΙΑΣ ΑΡΙΣΤΕΙΔΗΣ, 1069910, Δ' ΕΤΟΣ*
- *ΠΑΠΑΘΑΝΑΣΙΟΥ ΑΛΕΞΙΟΣ, 1067501, Δ' ΕΤΟΣ*

*Το project μας αναπτύσσεται στο Github , στον παρακάτω σύνδεσμο:*

*<https://github.com/vasMil/GymBuddy>*

*Επιλέξαμε να το χρησιμοποιήσουμε και για τα αρχεία word. Επειδή όμως το git σαν εργαλείο προσφέρει δυνατότητες merge των αρχείων μόνο για αρχεία κώδικα, είμαστε ιδιαίτερα προσεκτικοί, ώστε δύο μέλη της ομάδας να μην επεμβαίνουν στο ίδιο αρχείο τις ίδιες χρονικές περιόδους. Το πρόβλημα αυτό θα μπορούσε να λυθεί με τη χρήση εργαλείων, όπως τα google docs. Επιλέξαμε το git, ώστε να εξοικειωθούμε με το εργαλείο που διδαχτήκαμε.*

## Admin Checks Report

Μεταβλητή Εισόδου	Κλάσεις ισοδυναμίας	Περιπτώσεις Ελέγχου
Reports (Πεδίο Τιμών)	Reports <0 (άκυρες τιμές) Reports >=0	Reports = -1  Reports = 0 Reports = 1
Violates (Σύνολο Τιμών)	Violates in {True, False} Violates not in {True, False}	Violates = true Violates = 55
Violations (Πεδίο Τιμών)	Violations <0 (άκυρες τιμές) Violations >=0 Violations <=3	Violations = -1  Violations = 0 Violations = 1 Violations = 3
ViolationCategory (Πεδίο Τιμών)	ViolationCategory <0(άκυρες τιμές) ViolationCategory >=0	ViolationCategory = -1  ViolationCategory =0 ViolationCategory =1
NotifyAccuser (Σύνολο Τιμών)	NotifyAccuser in {True, False} NotifyAccuser not in {True, False}	NotifyAccuser = true NotifyAccuser = 3

## Trainer Accepts Request

Μεταβλητή Εισόδου	Κλάσεις ισοδυναμίας	Περιπτώσεις Ελέγχου
Requests (Πεδίο Τιμών)	Requests <0 (άκυρες τιμές) Requests >=0	Requests = -1  Requests = 0 Requests = 1
acceptRequest (Σύνολο Τιμών)	acceptRequest in {True, False} acceptRequest not in {True, False}	acceptRequest = True acceptRequest = 33
addReasonOfDenial (Πεδίο Τιμών)	addReasonOfDenial<30(άκυρες τιμές) addReasonOfDenial>=30	addReasonOfDenial = 29  addReasonOfDenial = 30 addReasonOfDenial = 31

## Είσοδος στην κατάταξη διαγωνισμού

Μεταβλητές	Κλάσεις Ισοδυναμίας	Περιπτώσεις Ελέγχου
Kgs (πεδίο τιμών)	Kg<0 (άκυρες τιμές) 0<=kgs<=1000 Kgs>1000 (άκυρες τιμές)	Kg= ο μικρότερος δυνατός ακέραιος αριθμός Kg= -1 Kg= 0 Kgs=1000 Kgs=1001 Kgs= ο μεγαλύτερος δυνατός ακέραιος αριθμός
Exercise category.name (σύνολο τιμών)	Categories in {legs, chest, shoulders, arms, back, cardio} Categories not in {legs, chest, shoulders, arms, back, cardio}	
Exercise.name (σύνολο τιμών)	If category = 'legs' Exercises in {front squad, walking lunges, step up, glute bridge, Olympic lift, deadlift}  If category = 'chest' Exercises in {bench press, push up, band chest fly, decline dumbbell bench press, cable fly}  If category = 'arms' Exercises in {standing dumbbell curl, standing barbell curl, ez-bar Preacher Curl, Crucifix curl, hammer curl, triceps pressdown, ez bar skull crushers}  If category = 'shoulders' Exercises in {military presses, dumbbell presses,	

	lat raises, overhead press, standing cable fly}  If category = 'back' Exercises in {lat pulldown, cross row, seated row, deadlift, pullups}  If category = 'cardio' Exercises in {treadmill, bicycle}	
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

### Διαδικασία πιστοποίησης γυμναστή

Μεταβλητή Εισόδου	Κλάσεις Ισοδυναμίας	Περιπτώσεις Ελέγχου
sendDocs (Σύνολο Τιμών)	sendDocs{TRUE, FALSE}	sendDocs= TRUE

### Εύρεση γυμναστή

Μεταβλητές	Κλάσεις Ισοδυναμίας	Περιπτώσεις Ελέγχου
gymHours (σύνολο τιμών)	gymHours in {08:00, 09:00, 10:00, 11:00.....22:00} gymHours not in {23:00....07:00}	gymHours = 09:00  gymHours = 24:00

## Create Program

Μεταβλητή Εισόδου (είδος)	Κλάσεις Ισοδυναμίας	Περιπτώσεις Ελέγχου
numberOfDays (πεδίο τιμών)	<p>numberOfDays &lt; 1 ( άκυρες τιμές)</p> <p>1 &lt;= numberOfDays &lt;= 7</p> <p>numberOfDays &gt; 7 ( άκυρες τιμές)</p>	<p>numberOfDays = 0 μικρότερος δυνατός αρνητικός ακέραιος. numberOfDays = 0</p> <p>numberOfDays = 1 numberOfDays = 7</p> <p>numberOfDays = 8 numberOfDays = 0 μεγαλύτερος δυνατός θετικός ακέραιος.</p>
idealKg (πεδίο τιμών)	<p>idealKg &lt;= 0 ( άκυρες τιμές)</p> <p>0 &lt; idealKg &lt;= 550</p> <p>idealKg &gt; 550 ( άκυρες τιμές)</p>	<p>idealKg = 0 μικρότερος δυνατός αρνητικός δεκαδικός. idealKg = 0</p> <p>idealKg = 1 idealKg = 550</p> <p>idealKg = 551 idealKg = 0 μεγαλύτερος δυνατός θετικός ακέραιος.</p>
idealHeight (πεδίο τιμών)	<p>idealHeight &lt;= 0 ( άκυρες τιμές)</p> <p>0 &lt; idealHeight &lt; 300</p> <p>idealHeight &gt;= 300 ( άκυρες τιμές)</p>	<p>idealHeight = 0 μικρότερος δυνατός αρνητικός δεκαδικός. idealHeight = 0</p> <p>idealHeight = 1 idealHeight = 299</p> <p>idealHeight = 300 idealHeight = 0 μεγαλύτερος δυνατός θετικός ακέραιος.</p>
workoutDay.name (σύνολο τιμών)	<p>workoutDay.name in { Έγκυρα workoutDay.name }</p> <p>workoutDay.name not in { Έγκυρα workoutDay.name }</p> <p>(* έγκυρο -&gt; 0 &lt; workoutDay.name &lt; 10 χαρακτήρες)</p>	<p>workoutDay.name = ' Πόδια '</p> <p>workoutDay.name = ' '</p>
ExerciseCategory.name (σύνολο τιμών)	<p>ExerciseCategory.name in {Legs, Chest, Shoulders, Arms, Back, Cardio}</p>	

Exercise.name (σύνολο τιμών)	<p>If ExerciseCategory.name = 'legs' Exercise.name in {front squad, walking lunges, step up, glute bridge, Olympic lift, deadlift}</p> <p>If ExerciseCategory.name = 'chest' Exercise.name in {bench press, push up, band chest fly, decline dumbbell bench press, cable fly}</p> <p>If ExerciseCategory.name = 'arms' Exercise.name in {standing dumbbell curl, standing barbell curl, ez-bar Preacher Curl, Crucifix curl, hammer curl, triceps pressdown, ez bar skull crushers}</p> <p>If ExerciseCategory.name = 'shoulders' Exercise.name in {military presses, dumbbell presses, lat raises, overhead press, standing cable fly}</p> <p>If ExerciseCategory.name = 'back' Exercise.name in {lat pulldown, cross row, seated row, deadlift, pullups}</p> <p>If ExerciseCategory.name = 'cardio' Exercise.name in {treadmill, bicycle}</p>	

## Upload Video

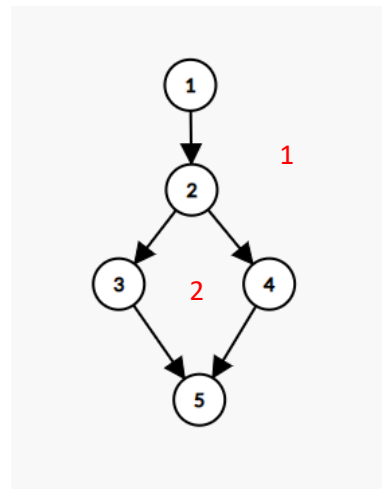
Στη κατηγορία τεχνικής white box, επιλέξαμε τη μέθοδο path coverage που περιγράφεται στα [1] και [2]. Παρακάτω περιγράφουμε το test case για το sequence «Ανάρτηση Βίντεο» ή στα αγγλικά “Upload Video”. Αρχικά παραθέτουμε τον κώδικα, σημειώνοντας πάνω σε αυτός τα κομμάτια εκείνα που θεωρούμε σαν κόμβους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

1) Test Case 1<sup>ο</sup>: Στην μέθοδο onCreate της κλάσης UserProfile

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.setup_prof)
    supportActionBar?.apply { this: ActionBar
        title = "User Profile"
        setBackgroundDrawable(ColorDrawable(resources.getColor(R.color.main_color)))
    }
1 // Setup Views
    val usernameView = findViewById<TextView>(R.id.username_textView)
    val changeGymButton = findViewById<Button>(R.id.changeGym_button)
    val gymView = findViewById<TextView>(R.id.gym_textView)

    usernameView.text = user.name
2 if(user.currentGym == null) { ←
3     changeGymButton.text = "Add Gym"
    }
    else {
4     gymView.text = user.currentGym!!.name
    }

    // Add event listener on changeGym button so it redirects to the MapBoundary
    changeGymButton.setOnClickListener { it: View!
5        val intentMap = Intent( packageContext: this, MapBoundary::class.java)
        startActivity(intentMap)
    }
}
```



Υπολογισμός κυκλωματικής πολυπλοκότητας:

$$V(g) = |E| - |V| + 2 \Rightarrow V(g) = 5 - 5 + 2 \Rightarrow V(g) = 2$$

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
N1	null	Έχει τη τιμή null	Το κουμπί του προφίλ έχει το κείμενο Add Gym
N2	Υπάρχει valid Gym object	Δεν έχει τιμή null	Το κουμπί του προφίλ έχει το κείμενο Change Gym

Με τα μονοπάτια να είναι:

N1: 1-2-3-5

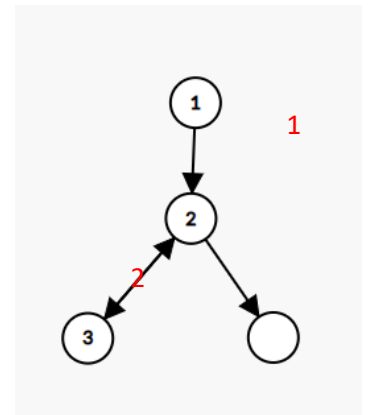
N2: 1-2-4-5

Μονοπάτι	Σ2 (User.currentGym == null)
N1	T
N2	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.

2) Test case 2<sup>ο</sup>: Στη μέθοδο displayExercises της κλάσης ChooseExerciseBoundary

```
private fun displayExercises() {
    // fetch the exercises
    category.fetchExercises()
    exercises = category.exercises.toTypedArray()
    // display buttons
    val exercisesView = findViewById<LinearLayout>(R.id.exerciseScrollView)
    exercises.forEach { it -> <←
        val btn = createExerciseButton(exercisesView, it)
        btn.setOnClickListener { itBtn ->
            // TODO: Pass control to "upload a file" use case
            // Update confirmation button for the upload and display it
            displayConfirmButton(itBtn.tag.toString().split(" ")[1].toInt())
            confirmButton.setOnClickListener { it: View!
                handleConfirmation()
            }
        }
    }
}
```



Σημειώνουμε ότι ο κενός κόμβος υποδηλώνει τη λήξη της συνάρτησης.

Υπολογισμός κυκλωματικής πολυπλοκότητας:

$$V(g) = 4 - 4 + 2 \Rightarrow V(g) = 2$$

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
N1	0	Το array exercises είναι άδειο	Το κουμπί του προφίλ έχει το κείμενο Add Gym
N2	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:

N1: 1-[2-3]<sup>exercises.size</sup>-2-(κενός κόμβος)

N2: 1-2-(κενός κόμβος)

Κάλυψη συνθηκών:

Μονοπάτι	Σ2 (i < exercises.size)
N1	F(exercises.size), T
N2	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 N2 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 N, N-1.

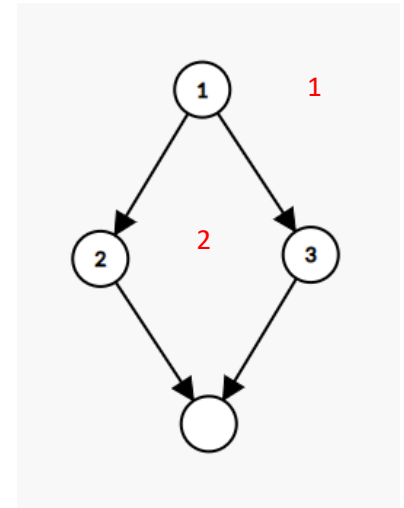
Οπότε:

Μονοπάτι	exercises.forEach{ ... }
N1	10
N2	0
N3	1

Επομένως θα εξετάσουμε και τα 3 μονοπάτια N1, N2, N3, εφόσον θέλουμε να κάνουμε έλεγχο βρόγχου.

Test case 3º: Στη μέθοδο handleConfirmation της κλάσης ChooseExerciseBoundary

```
private fun handleConfirmation() {
1  if(acceptSwitch.isChecked) {
    Toast.makeText( context: this@ChooseExerciseBoundary, text: "Success!",
2      Toast.LENGTH_SHORT)
      .show()
    val intent = Intent( packageContext: this, UserProfile::class.java)
    startActivity(intent)
  }
  else {
3    Toast.makeText( context: this@ChooseExerciseBoundary, text: "Failed to upload video!",
      Toast.LENGTH_SHORT)
      .show()
  }
}
```



Υπολογισμός κυκλωματικής πολυπλοκότητας:

$$V(g) = |E| - |V| + 2 \Rightarrow V(g) = 4 - 4 + 2 \Rightarrow V(g) = 2$$

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
N1	true	Έχει τη τιμή true	Το σύστημα δέχεται τα αρχεία που ανέβασε ο Αθλητής
N2	false	Έχει τη τιμή false	Το σύστημα απορρίπτει τα αρχεία που ανέβασε ο Αθλητής

Με τα μονοπάτια να είναι:

N1: 1-2-(κενός κόμβος)

N2: 1-3-(κενός κόμβος)

Μονοπάτι	Σ1 (acceptSwitch.isChecked)
N1	T
N2	F

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

## Αναφορές

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