

## Αριστείδης Παλιούρας

Διδασκαλία αντικειμενοστραφούς προγραμματισμού με αξιοποίηση του BBC micro:bit και της γλώσσας προγραμματισμού Python (Φύλλο εργασίας 2)

Σε αυτή τη δραστηριότητα, θα σχεδιάσουμε ένα πρόγραμμα για να διαχειριστούμε ένα **φωτεινό σηματοδότη** με τρία χρώματα (κόκκινο, πορτοκαλί, πράσινο). Χρησιμοποιώντας τον αντικειμενοστραφή προγραμματισμό, θα δημιουργήσουμε μία κλάση για τον **φωτεινό σηματοδότη** και θα ελέγξουμε τη λειτουργία του χρησιμοποιώντας το **micro:bit**.

### Βήμα 1

Θα χρησιμοποιήσουμε το **micro:bit** για να προσομοιώσουμε τη λειτουργία του φωτεινού σηματοδότη. Θα δημιουργήσουμε μια κλάση **TrafficLight** με τα παρακάτω χαρακτηριστικά και συμπεριφορά.

Σχεδιασμός της κλάσης TrafficLight - Χαρακτηριστικά		
Χαρακτηριστικό	Όνομα μεταβλητής	Περιγραφή μεταβλητής
Κατάσταση	status	Η κατάσταση του φωτεινού σηματοδότη. Η μεταβλητή έχει μία από τις παρακάτω τιμές:  1: Πράσινο φως 2: Πορτοκαλί φως 3: Κόκκινο φως

Σχεδιασμός της κλάσης TrafficLight – Συμπεριφορά		
Ενέργεια	Όνομα μεθόδου	Περιγραφή μεθόδου
Ανάβει το πράσινο φως	green_light()	Στον φωτεινό σηματοδότη ανάβει το πράσινο φως για 3 δευτερόλεπτα. Στην οθόνη του <b>micro:bit</b> θα εμφανίζεται ο <b>χαρακτήρας Π</b> για 3 δευτερόλεπτα.
Ανάβει το πορτοκαλί φως	orange_light()	Στον φωτεινό σηματοδότη ανάβει το πορτοκαλί φως για 1 δευτερόλεπτο. Στην οθόνη του <b>micro:bit</b> θα εμφανίζεται ο <b>χαρακτήρας Ο</b> για 1 δευτερόλεπτο.
Ανάβει το κόκκινο φως	red_light()	Στον φωτεινό σηματοδότη ανάβει το κόκκινο φως για 2 δευτερόλεπτα. Στην οθόνη του <b>micro:bit</b> θα εμφανίζεται ο <b>χαρακτήρας Κ</b> για 2 δευτερόλεπτα.

## Αριστείδης Παλιούρας

Διδασκαλία αντικειμενοστραφούς προγραμματισμού με αξιοποίηση του BBC micro:bit και της γλώσσας προγραμματισμού Python (Φύλλο εργασίας 2)

### Βήμα 2

Στο περιβάλλον προγραμματισμού <https://python.microbit.org/v/3> ορίζουμε την κλάση TrafficLight ως εξής:

```
from microbit import *
import time

class TrafficLight:
    def __init__(self):
        self.status = 1 # Ξεκινά με πράσινο φως

    def green_light(self):
        self.status = 1
        display.show(Image('99999:'
                            '90009:'
                            '90009:'
                            '90009:'
                            '90009'))

    def orange_light(self):
        self.status = 2
        display.show(Image('99999:'
                            '90009:'
                            '90009:'
                            '90009:'
                            '99999'))

    def red_light(self):
        self.status = 3
        display.show(Image('90090:'
                            '90900:'
                            '99000:'
                            '90900:'
                            '90090'))
```

Η μέθοδος `__init__` είναι μια **ειδική** μέθοδος στην Python που ονομάζεται **κατασκευαστής (constructor)**. Χρησιμοποιείται για να αρχικοποιήσει τα χαρακτηριστικά ενός αντικειμένου όταν δημιουργείται.

Η μέθοδος `__init__` καλείται αυτόματα όταν δημιουργείται ένα νέο αντικείμενο από μια κλάση.

## Αριστείδης Παλιούρας

Διδασκαλία αντικειμενοστραφούς προγραμματισμού με αξιοποίηση του *BBC micro:bit* και της γλώσσας προγραμματισμού *Python* (**Φύλλο εργασίας 2**)

Η μέθοδος `__init__` (όπως και όλες οι μέθοδοι) λαμβάνει πάντα την παράμετρο `self` ως το πρώτο όρισμα. Η παράμετρος `self` αναφέρεται στο ίδιο το αντικείμενο, επιτρέποντας στην μέθοδο να έχει πρόσβαση και να τροποποιεί τα χαρακτηριστικά του.

### Βήμα 3

Δημιουργία αντικειμένου.

Ο κώδικας για τη δημιουργία ενός **αντικειμένου** της κλάσης **TrafficLight** που ορίσαμε προηγουμένως είναι:

```
# Δημιουργία αντικειμένου  
traffic_light = TrafficLight()
```

### Βήμα 4

Κυρίως πρόγραμμα.

```
while True:  
    traffic_light.green_light()  
    time.sleep(3)  
    traffic_light.orange_light()  
    time.sleep(1)  
    traffic_light.red_light()  
    time.sleep(2)
```

Στο παραπάνω πρόγραμμα δεν έχουμε αξιοποιήσει τη μεταβλητή **status**.

Μια καλή ιδέα για την αξιοποίηση της μεταβλητής **status** είναι να τη χρησιμοποιήσουμε για να ελέγχουμε και να αλλάζουμε την κατάσταση του φωτεινού σηματοδότη δυναμικά, αντί να καλούμε απευθείας τις μεθόδους `green_light()`, `orange_light()` και `red_light()`.

Για παράδειγμα, μπορούμε να δημιουργήσουμε μια μέθοδο **next\_state()** που θα αλλάζει την κατάσταση του σηματοδότη κυκλικά (πράσινο → πορτοκαλί → κόκκινο → πράσινο).

Οπότε ο κώδικας γίνεται ως εξής:

## Αριστείδης Παλιούρας

Διδασκαλία αντικειμενοστραφούς προγραμματισμού με αξιοποίηση του BBC micro:bit και της γλώσσας προγραμματισμού Python (Φύλλο εργασίας 2)

```
from microbit import *

class TrafficLight:
    def __init__(self):
        self.status = 1 # Ξεκινά με πράσινο φως

    def update_display(self):
        #Ενημερώνει την οθόνη του micro:bit με το σωστό σύμβολο
        if self.status == 1:
            display.show(Image('99999:'
                               '90009:'
                               '90009:'
                               '90009:'
                               '90009'))
        elif self.status == 2:
            display.show(Image('99999:'
                               '90009:'
                               '90009:'
                               '90009:'
                               '99999'))
        elif self.status == 3:
            display.show(Image('90090:'
                               '90900:'
                               '99000:'
                               '90900:'
                               '90090'))

    def next_state(self):
        #Αλλάζει την κατάσταση του φωτεινού σηματοδότη κυκλικά
        if self.status == 1:
            self.status = 2 # Μετάβαση σε πορτοκαλί
            self.update_display()
            sleep(1000)
        elif self.status == 2:
            self.status = 3 # Μετάβαση σε κόκκινο
            self.update_display()
            sleep(2000)
        elif self.status == 3:
            self.status = 1 # Μετάβαση σε πράσινο
            self.update_display()
            sleep(3000)
```

## Αριστείδης Παλιούρας

Διδασκαλία αντικειμενοστραφούς προγραμματισμού με αξιοποίηση του BBC micro:bit και της γλώσσας προγραμματισμού Python (**Φύλλο εργασίας 2**)

```
# Δημιουργία αντικειμένου TrafficLight
traffic_light = TrafficLight()

# Κυκλική εναλλαγή σηματοδότη
while True:
    traffic_light.next_state()
```