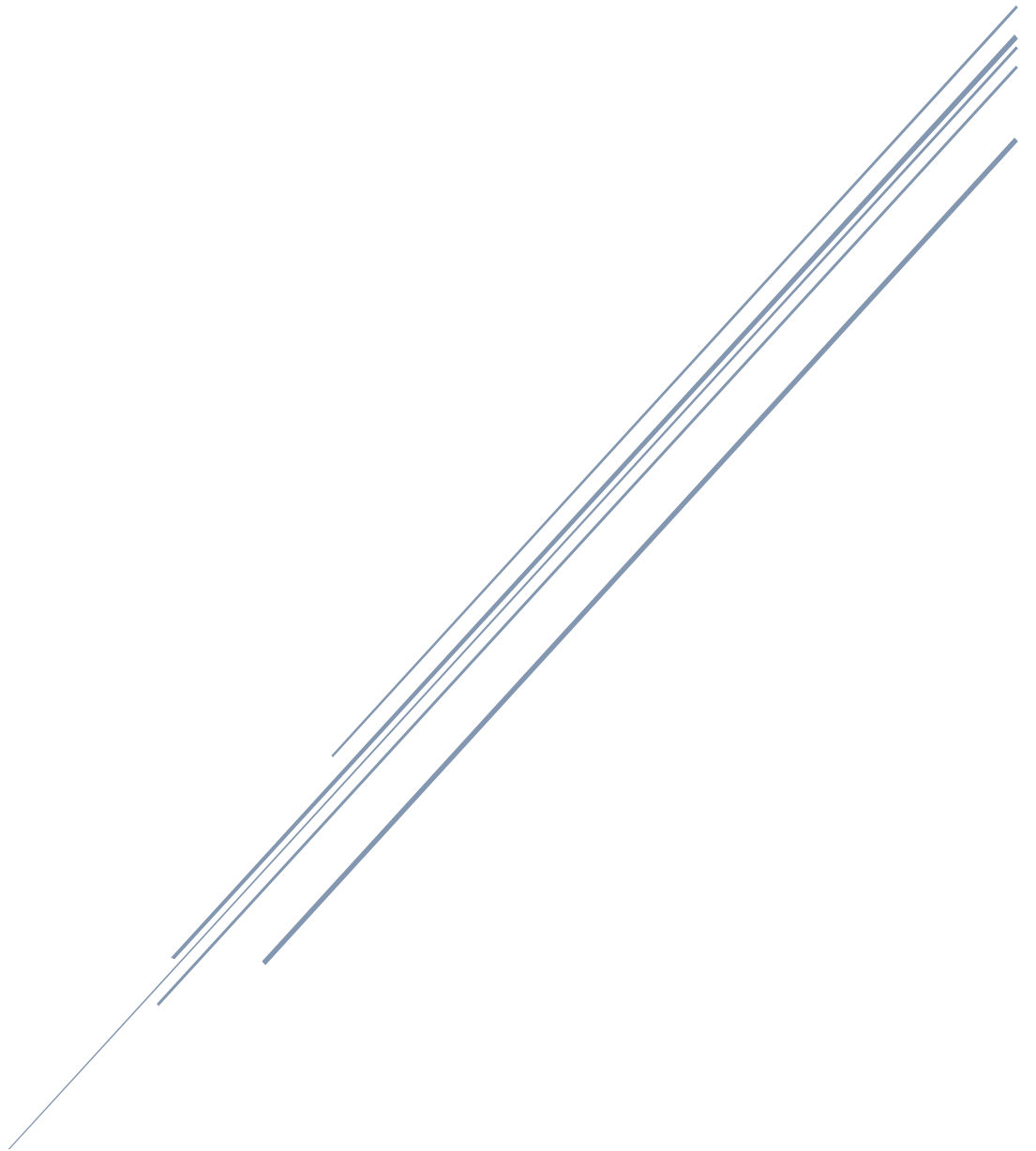


# ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ (C#)

PROGRAMMING CHALLENGE 2018-2019



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
UNIVERSITY OF PIRAEUS

Δημήτρης Ματσαγγάνης  
Π17068

# Εκφώνηση ζητούμενης εργασίας

## ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

PROGRAMMING CHALLENGE 2 ΜΟΝΑΔΩΝ 2018-2019

### Περιγραφή:

Στα «έγγραφα» θα βρείτε το αρχείο “challenge\_18-19.zip”. Μέσα σε αυτό υπάρχουν 2 εκτελέσιμα

αρχεία (.exe) από 2 διαφορετικά C# projects. Στόχος σας είναι να μελετήσετε όσο καλύτερα γίνεται τι ακριβώς κάνουν τα εκτελέσιμα αυτά (αυτό είναι μέρος της εργασίας σας) και στη συνέχεια:

- Να φτιάξετε τα δικά σας εκτελέσιμα «αντιγράφοντας» όσο πιο πιστά γίνεται ό,τι κάνει το κάθε ένα εκτελέσιμο.
- Εφόσον καταφέρετε το παραπάνω, να προχωρήσετε και σε υλοποίηση 2ου version (ξεχωριστό αρχείο) του κάθε ενός από τα 2 εκτελέσιμα με ό,τι βελτιώσεις θεωρείτε εσείς ότι θα κάνουν καλύτερο το κάθε project.

### Παραδοτέα:

- Κάθε project που θα καταφέρετε να υλοποιήσετε θα πρέπει στον κώδικά του να έχει αναλυτικά σχόλια της «λογικής» που ακολουθήσατε για την επίλυση του challenge (όχι σχολιασμός στα «προφανή» σημεία του κώδικα).
- Ένα σύντομο έγγραφο όπου να αναγράφετε τι καταφέρατε και τι όχι, καθώς και τις προτάσεις βελτίωσης που αναπτύξατε (εφόσον τα καταφέρατε). Στο έγγραφό αυτό μπορείτε να παραθέσετε και μερικά screenshots από την εκτέλεση των project σας. Τύπος αρχείου εγγράφου, .docx ή .pdf.
- Τα 2, ή 4 (στην περίπτωση των βελτιωμένων projects) projects, μαζί με το τελικό έγγραφο θα ζιπαριστούν και θα ανέβουν στο χώρο των εργασιών.

# ΠΕΡΙΕΧΟΜΕΝΑ

---

|     |   |    |
|-----|---|----|
| 1   | Εισαγωγή της εργασίας.....                      | 4  |
| 2   | Το Πρώτο εκτελέσιμο (challenge_1.exe). ....     | 4  |
| 2.1 | Περιγραφή του Προγράμματος.....                 | 4  |
| 2.2 | Βιβλιοθήκες και Αρχικοποιήσεις μεταβλητών. .... | 5  |
| 2.3 | Η Συνάρτηση button1_Click. ....                 | 6  |
| 2.4 | Οι Υπόλοιπες Συναρτήσεις.....                   | 8  |
| 3   | Το Δεύτερο εκτελέσιμο (challenge_2.exe).....    | 9  |
| 3.1 | Περιγραφή του Προγράμματος.....                 | 9  |
| 3.2 | Βιβλιοθήκες και Αρχικοποιήσεις μεταβλητών. .... | 10 |
| 3.3 | Η Συνάρτηση button1_Click. ....                 | 11 |

## 1 ΕΙΣΑΓΩΓΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.

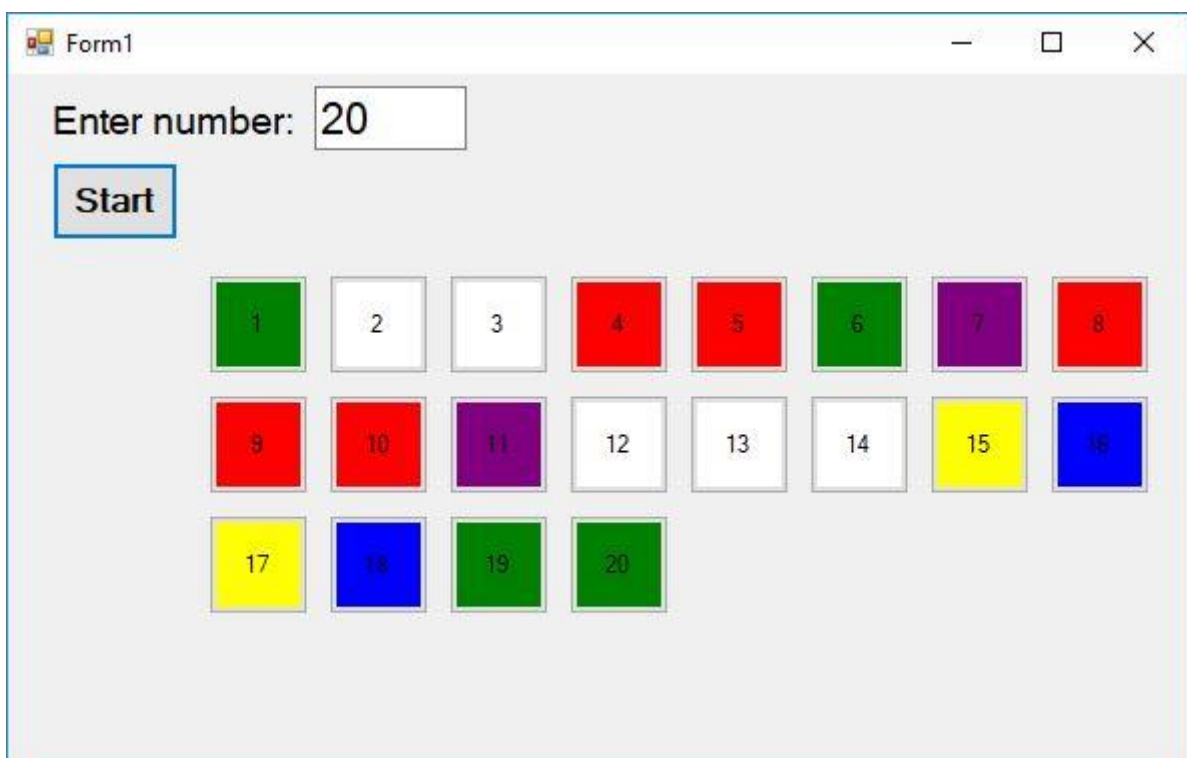
Η παρούσα εργασία, αποτελείται από δύο μέρη (δύο διαφορετικά εκτελέσιμα αρχεία ) και ως κύριο σκοπό της έχει την αποκωδικοποίηση της λειτουργίας του εκτελέσιμου αρχείου σε πρώτο στάδιο και την υλοποίησή του σε δεύτερο.

## 2 ΤΟ ΠΡΩΤΟ ΕΚΤΕΛΕΣΙΜΟ (CHALLENGE\_1.EXE).

Σε αυτήν την ενότητα θα παρουσιαστεί πλήρως το πρώτο από τα δυο δοθέντα εκτελέσιμα.

### 2.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ.

Το πρώτο εκτελέσιμο που δόθηκε είναι το πρόγραμμα (challenge\_1.exe), το οποίο δέχεται από τον χρήστη ένα θετικό ακέραιο (αν δοθεί αρνητικός ακέραιος ή το μηδέν από το χρήστη τότε δεν θα γίνει τίποτα) και στη συνέχεια εμφανίζει τόσα κουτιά, όσα και ο αριθμός που δόθηκε προηγουμένως από τον χρήστη, τα οποία λαμβάνουν τυχαία χρώματα και αναγράφουν τον αριθμό τους (από το 1 στο πρώτο κουτί, μέχρι τον αριθμό που έδωσε ο χρήστης στο τελευταίο κουτί). Ακόμη, παρατηρήθηκε ότι τα κουτιά αλλάζουν χρώμα όταν μετακινηθεί η εφαρμογή ή όταν αλλάξει το μέγεθος της φόρμας (στη περίπτωση που δοθεί ένας αριθμός και μετά δοθεί ένας μεγαλύτερος του, τότε μετά θα αλλάζουν μόνο τα κουτιά που προστέθηκαν, δηλαδή η διαφορά των δύο αριθμών).

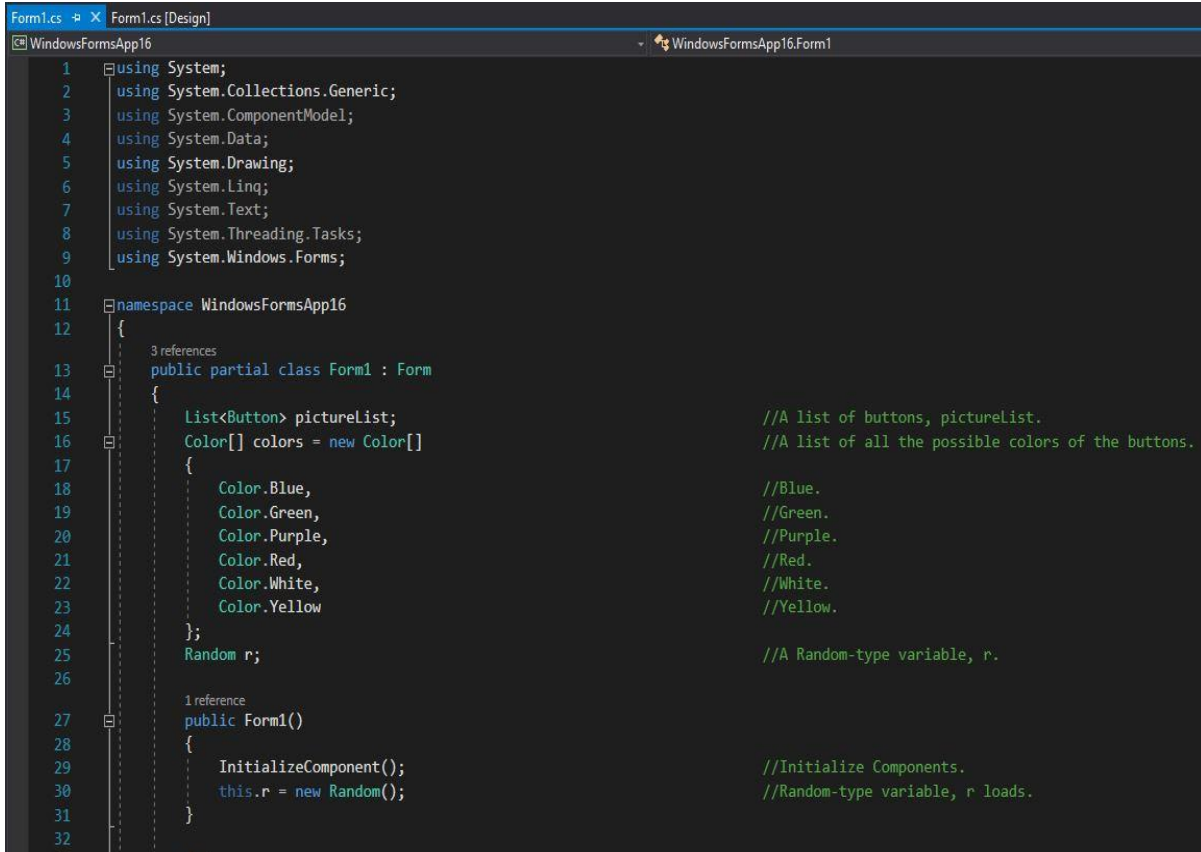


Εικόνα του 1ου εκτελέσιμου 1.

Τέλος, όταν ένα κουτί πατηθεί τότε εμφανίζει με ένα μήνυμα (MessageBox) το οποίο αναγραφεί τον αριθμό του κουτιού.

## 2.2 ΒΙΒΛΙΟΘΗΚΕΣ ΚΑΙ ΑΡΧΙΚΟΠΟΙΗΣΕΙΣ ΜΕΤΑΒΛΗΤΩΝ.

Για την σωστή λειτουργία του προγράμματος έγινε χρήση όλων των απαιτούμενων βιβλιοθηκών, όπως φαίνεται και στην παρακάτω εικόνα.



```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace WindowsFormsApp16
12 {
13     public partial class Form1 : Form
14     {
15         List<Button> picturelist;           //A list of buttons, picturelist.
16         Color[] colors = new Color[]      //A list of all the possible colors of the buttons.
17         {
18             Color.Blue,                   //Blue.
19             Color.Green,                  //Green.
20             Color.Purple,                 //Purple.
21             Color.Red,                    //Red.
22             Color.White,                  //White.
23             Color.Yellow                   //Yellow.
24         };
25         Random r;                         //A Random-type variable, r.
26
27         public Form1()
28         {
29             InitializeComponent();         //Initialize Components.
30             this.r = new Random();         //Random-type variable, r loads.
31         }
32     }
33 }
```

Εικόνα του κώδικα 1ου εκτελέσιμου 1.

Αρχικά, ορίζονται δύο λίστες διαφορετικού τύπου, μια λίστα Button, που ονομάζεται pictureList, η οποία θα περιέχει τα κουμπιά-εικόνες και μια λίστα χρωμάτων (Color), η οποία θα περιέχει όλα τα πιθανά χρώματα που μπορούν να πάρουν τα κουτιά (δηλαδή: μπλε, πράσινο, μωβ, κόκκινο, άσπρο, κίτρινο). Τέλος, ορίζεται μια μεταβλητή τύπου Random (Random-type variable) που ονομάζεται r, είναι ικανή για την τυχαία επιλογή στοιχείων και αρχικοποιείται ταυτόχρονα με την ιδιά την φόρμα.

## 2.3 Η ΣΥΝΑΡΤΗΣΗ BUTTON1\_CLICK.

Η βασική συνάρτηση του 1<sup>ου</sup> εκτελέσιμου είναι η button1\_Click, η οποία ενεργοποιείται στο πάτημα μιας εικόνας (κουμπιού) και είναι υπεύθυνη για την εμφάνιση καθώς και για την διάταξη και τοποθέτηση των εικόνων στην εφαρμογή.

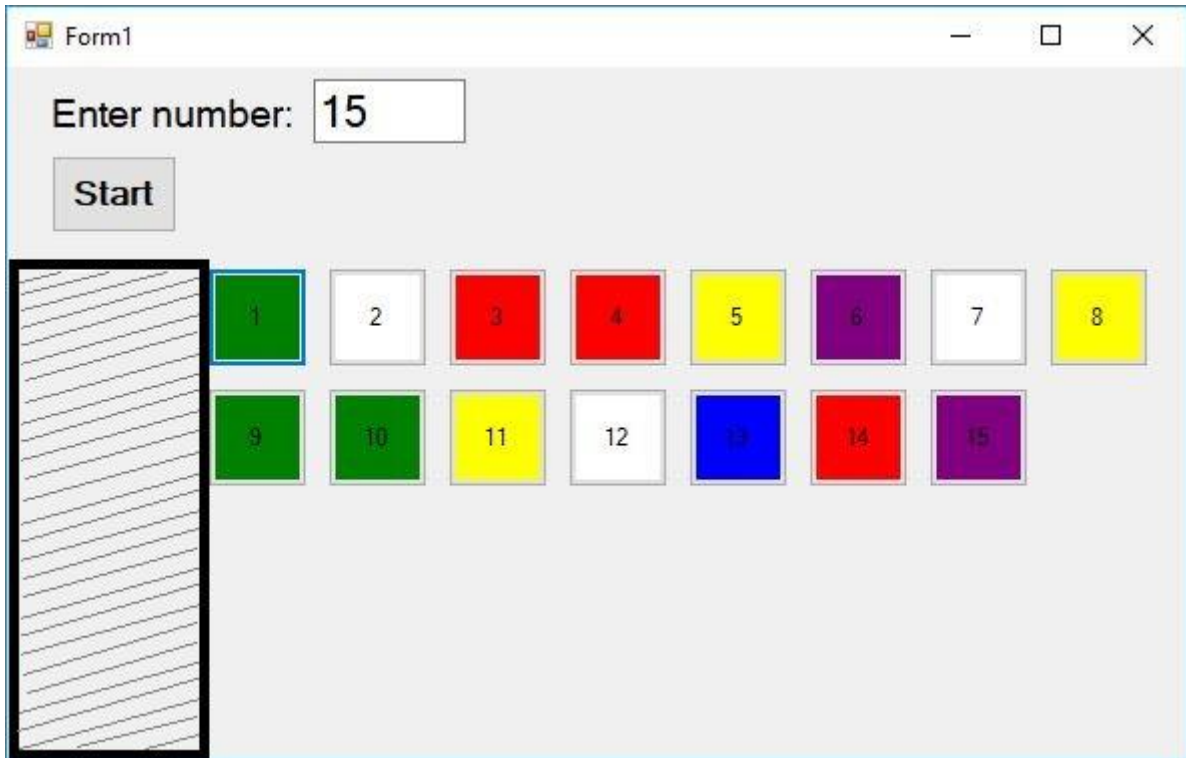
```
33 private void button1_Click(object sender, EventArgs e)
34 {
35     int input_num; //An int-type variable, input_num.
36     int button_width = 50; //An int-type variable, button_width equals to 50 "pixels".
37     int button_height = 50; //An int-type variable, button_height equals to 50 "pixels".
38     int empty_space = 100; //An int-type variable, empty_space represents the empty space down from the start button.
39     int btn_num = 60; //An int-type variable, btn_num represents the size of button and the space between two buttons(50 + 10 = 60 pixels).
40     int det_num = (this.Width - empty_space) / btn_num * btn_num; //An int-type variable, det_num determines button's location.
41     picturelist = new List<Button>(); //List of buttons, picturelist.
42
43     try
44     {
45         input_num = Convert.ToInt32(this.textBox1.Text); //Every input in textBox1 converts to 32-bit integer.
46     }
47     catch
48     {
49         //Exception.
50     }
51     return;
52
53     for (int i = 0; i < input_num; i++) //A for-loop.
54     {
55         Button button = new Button(); //Generates a new button per loop.
56         button.Size = new Size(button_width, button_height); //Every button's size is determined.
57         int x_axis_number = i * btn_num % det_num / btn_num * btn_num; //An int-type variable, x_axis_number responsible for x-axis button's location.
58         int y_axis_number = i * btn_num / det_num * btn_num; //An int-type variable, y_axis_number responsible for y-axis button's location.
59         button.Location = new Point(empty_space + x_axis_number, empty_space + y_axis_number);
60         //Button's location is determined by x_axis_number and y_axis_number after empty_space is added respectively.
61         int color_number = r.Next(0, 6); //An int-type variable, color_number is a random integer between 0 and 6.
62         button.BackColor = this.colors[color_number]; //The randomly selected number equals to a color from the colors array.
63         button.Text = (i + 1).ToString(); //Button's text is generated by the for-loop.
64         button.Click += new EventHandler(this.pictureClick); //Button's click event handler is the pictureClick void.
65         Controls.Add(button); //Form adds new buttons.
66         picturelist.Add(button); //List of buttons, picturelist adds new button.
67     }
68 }
```

Εικόνα του κώδικα 1ου εκτελέσιμου 2.

Αρχικά ορίζονται οι εξής ακέραιες μεταβλητές:

1. **input\_num**: Η οποία δέχεται ένα όρισμα από το χρήστη και το μετατρέπει σε ακέραιο αριθμό μέσα από μια μέθοδο try για τυχών σφάλματα (exceptions).
2. **button\_width**: Η οποία ορίζει το πλάτος του κουτιού στα 50 pixels.
3. **button\_height**: Η οποία ορίζει το ύψος του κουτιού στα 50 pixels.
4. **empty\_space**: Η οποία έχει αρχικοποιηθεί με την τιμή των 100 pixels, όσο είναι και το κενό που βρίσκεται κάτω από το κουμπί start (το σκιαγραφημένο πλαίσιο στην εικόνα «Εικόνα του 1ου εκτελέσιμου 2. »)
5. **btn\_num**: Η οποία έχει αρχικοποιηθεί με την τιμή των 60 pixels, όσο είναι και το μέγεθος του κουμπιού (50 pixels) αθροιστικά με το κενό αναμεσά σε δύο κουμπιά (10 pixels).
6. **det\_num**: Η οποία καθορίζει την τοποθεσία (determinant number) του κάθε κουμπιού μέσα στο for-loop, που θα περιγράφει παρακάτω. Ειδικότερα, σε πρώτο στάδιο αφού πάρει το πλάτος της φόρμας θα το περιορίζει στο χώρο που «επιτρέπεται» (μη σκιαγραφημένος χώρος στην εικόνα «Εικόνα του 1ου εκτελέσιμου 2. »), αφαιρώντας του την ακέραια μεταβλητή empty\_space και σε δεύτερο στάδιο θα διαιρεί τον «επιτρεπτό χώρο» με το εμβαδόν των κουτιών (σε συνδυασμό με τα κενά αναμεσά σε δύο κουτιά), δηλαδή την ακέραια μεταβλητή btn\_num πολλαπλασιασμένη με τον εαυτό της, με αυτόν τον τρόπο θα υπολογίζεται ο αριθμός των κουτιών που χωράνε στον «επιτρεπτό χώρο».

Έπειτα, ακολουθεί ένα for-loop μέσω του οποίου θα δημιουργούνται όσα κουμπιά θέλει ο χρήστης, δημιουργώντας ένα την φορά. Το κουτί θα παίρνει το μέγεθος του, την τοποθεσία του και το χρώμα του μέσα από αυτό το for-loop, καθώς και τον αριθμό που θα αναγραφεί επάνω του.



Εικόνα του 1ου εκτελέσιμου 2.

Αρχικά, όλα τα κουτιά θα έχουν το ίδιο μέγεθος και αυτό ορίζεται στα 50 pixels ως ύψος και ως πλάτος. Για την τοποθεσία τους ακολουθείται ένας αλγόριθμος κατά τον οποίο θα προσδιορίζεται η θέση του κουτιού τόσο στον άξονα X όσο και στον άξονα Y. Για τον άξονα X έχει ορισθεί η ακέραια μεταβλητή `x_axis_number`, η οποία θα πολλαπλασιάζει τον αριθμό του κουτιού με το μέγεθος του κουτιού, δηλαδή την ακέραια μεταβλητή `btn_num` και έπειτα θα πραγματοποιεί ακέρια διαίρεση (δηλαδή `mod`) με τον αριθμό καθορισμού του αριθμού των κουτιών που χωράνε στο επιτρεπτό πλαίσιο, δηλαδή της ακέραιας μεταβλητής `det_num` και σε δεύτερο στάδιο θα διαιρεί το παραπάνω γινόμενο με το εμβαδόν του κουτιού δηλαδή την ακέραια μεταβλητή `btn_num` πολλαπλασιασμένη με τον εαυτό της. Αντίστοιχα, για τον άξονα Y έχει ορισθεί η ακέραια μεταβλητή `y_axis_number`, η οποία θα πολλαπλασιάζει τον αριθμό του κουτιού με το μέγεθος του και μετά θα το διαιρεί με τον αριθμό καθορισμού της θέσης και τέλος το πολλαπλασιάζει με το μέγεθος του κουτιού `btn_num`. Τελικά, η θέση του κουτιού ορίζεται με την πρόσθεση της ακέραιας μεταβλητής `empty_space` τόσο στη μεταβλητή `x_axis_number` για τη θέση του στον άξονα X όσο και στη μεταβλητή `y_axis_number` για τη θέση του στον άξονα Y, για να είναι μακριά από την κορυφή και από την αριστερή άκρη όσο και η τιμή του `empty_space` (δηλαδή, τόσο στον άξονα X όσο και στον Y η ελάχιστη τιμή θα είναι το 100).

Το χρώμα του κουτιού θα καθορίζεται μέσα από μια νέα ακέραια μεταβλητή `Color_number` η οποία θα επιλεγεί τυχαία έναν αριθμό, ο οποίος σχετίζεται με το χρώμα που αντιστοιχεί



στον αριθμό στη λίστα χρωμάτων που έχει ορισθεί προηγουμένως, την colors. Ο αριθμός που αναγράφεται στο κουτί ορίζεται μέσα από το for\_loop και δημιουργείται από τον αριθμό του κουτιού μέσα στην διαδικασία επανάληψης αυξημένο κατά ένα και στο πάτημα του θα καλείται η συνάρτηση pictureClick, η οποία θα εμφανίζει τον αριθμό που αναγραφεί το κουτί μέσα σε ένα μήνυμα (MessageBox).

## 2.4 ΟΙ ΥΠΟΛΟΙΠΕΣ ΣΥΝΑΡΤΗΣΕΙΣ.

Σε αυτήν την ενότητα θα παρουσιαστούν τρεις συναρτήσεις, οι οποίες λειτουργούν συμπληρωματικά με την προαναφερθέντα συνάρτηση με σκοπό την πλήρη υλοποίηση των λειτουργιών του εκτελέσιμου.

```
69 public void pictureClick(object sender, EventArgs e)
70 {
71     MessageBox.Show(((Button)sender).Text);           //A MessageBox appears when the user clicks on a button(picture).
72 }
73
74 1 reference
75 private void Form1_ResizeBegin(object sender, EventArgs e)
76 {
77     if (pictureList != null)                           //If the pictureList is not empty.
78     {
79         foreach (Button p in this.pictureList)         //For each button in the pictureList.
80         {
81             Controls.Remove(p);                       //Removes one button from pictureList.
82         }
83         pictureList.Clear();                           //Clears the pictureList.
84     }
85 }
86 1 reference
87 private void Form1_ResizeEnd(object sender, EventArgs e)
88 {
89     if (pictureList != null)                           //If the pictureList is not empty.
90     {
91         button1.PerformClick();                       //Generates a click event for buttons.
92     }
93 }
94 }
```

Εικόνα του κώδικα 1ου εκτελέσιμου 3.

Οι συναρτήσεις αυτές είναι οι παρακάτω:

1. **pictureClick**: Η οποία είναι υπεύθυνη για την εμφάνιση ενός μηνύματος (MessageBox), το οποίο αναγραφεί τον αριθμό του κουτιού, στο οποίο θα κάνει κλικ ο χρήστης.
2. **Form1\_ResizeBegin**: Η οποία είναι υπεύθυνη για τον πλήρη καθαρισμό της λίστας pictureList όταν γίνεται μετατροπή του μεγέθους της φόρμας (resize), όταν αυτή δεν είναι άδεια.
3. **Form1\_ResizeEnd**: Η οποία είναι υπεύθυνη για την επανεμφάνιση των κουμπιών, δεδομένου ότι η λίστα pictureList δεν είναι κενή.

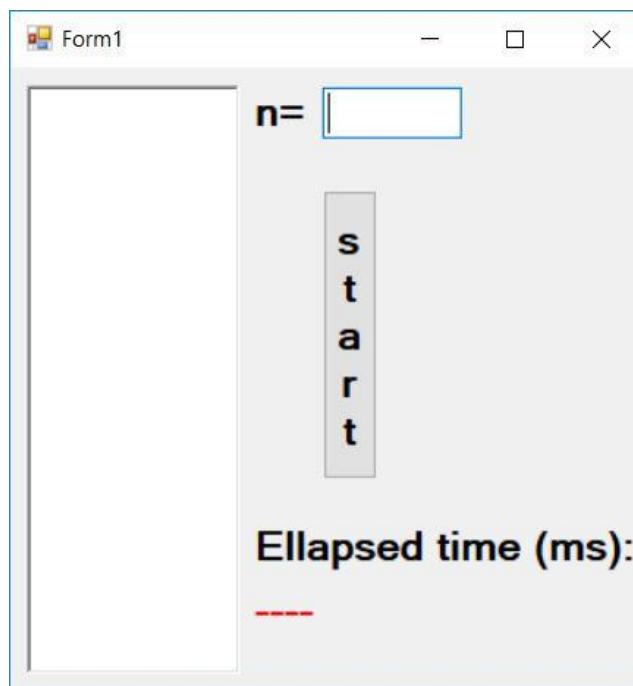


### 3 ΤΟ ΔΕΥΤΕΡΟ ΕΚΤΕΛΕΣΙΜΟ (CHALLENGE\_2.EXE).

Σε αυτήν την ενότητα θα παρουσιαστεί πλήρως το δεύτερο από τα δυο εκτελέσιμα που δοθήκαν.

#### 3.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ.

Ως δεύτερο εκτελέσιμο δόθηκε το πρόγραμμα (challenge\_2.exe), το οποίο δέχεται από τον χρήστη ένα θετικό ακέραιο αριθμό σε ένα πλαίσιο (textBox1) και μέσω ενός αλγορίθμου, ο οποίος θα περιγράφει αναλυτικά παρακάτω, εμφανίζει σε ένα δεύτερο πλαίσιο (richTextBox1) τους αριθμούς από το μηδέν (0) μέχρι τον αριθμό εισαγωγής (δηλαδή τον αριθμό που δόθηκε από τον χρήστη στο textBox1) με τυχαία σειρά.

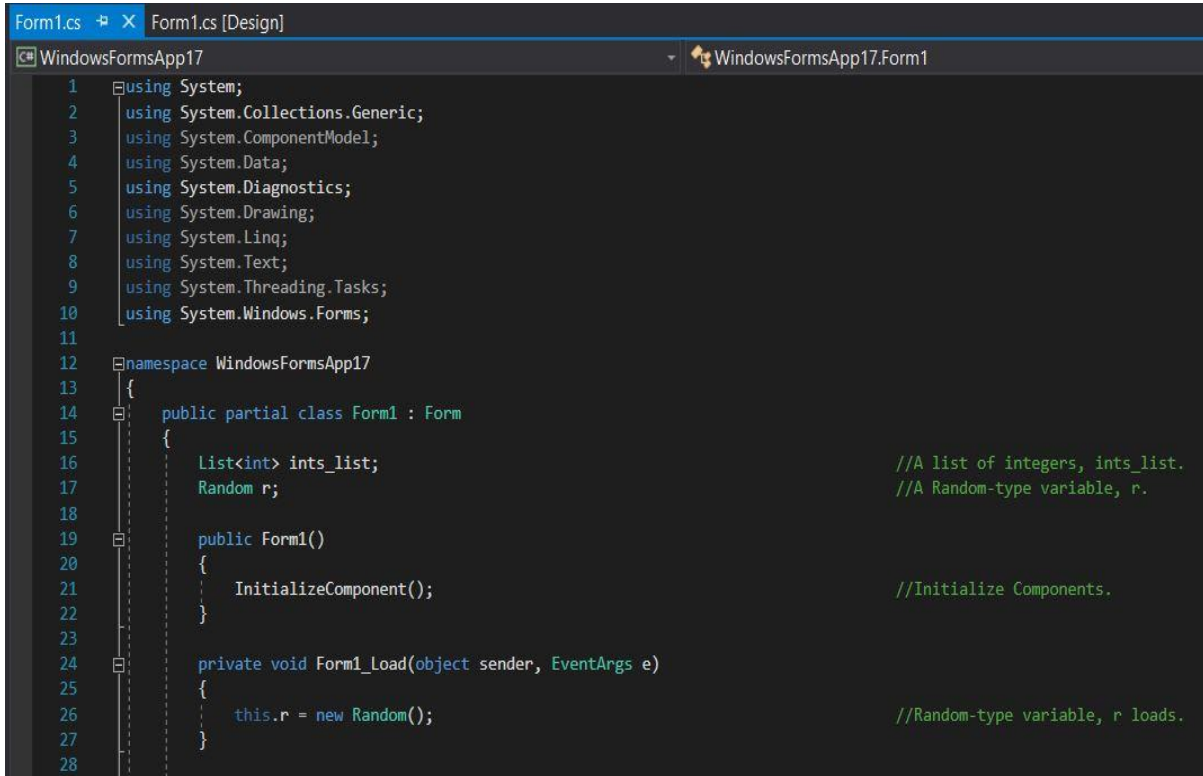


Εικόνα του 2ου εκτελέσιμου 1.

Ακόμη, το δεύτερο εκτελέσιμο κατά τη διάρκεια της παραπάνω διαδικασίας, καταμετρά τον χρόνο που χρειάστηκε το πρόγραμμα για την ολοκλήρωση της διαδικασίας που του ζητήθηκε από το χρήστη.

### 3.2 ΒΙΒΛΙΟΘΗΚΕΣ ΚΑΙ ΑΡΧΙΚΟΠΟΙΗΣΕΙΣ ΜΕΤΑΒΛΗΤΩΝ.

Για την σωστή λειτουργία του προγράμματος έγινε χρήση όλων των απαιτούμενων βιβλιοθηκών, όπως φαίνεται και στην παρακάτω εικόνα.



```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Diagnostics;
6 using System.Drawing;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 namespace WindowsFormsApp17
13 {
14     public partial class Form1 : Form
15     {
16         List<int> ints_list;           //A list of integers, ints_list.
17         Random r;                     //A Random-type variable, r.
18
19         public Form1()
20         {
21             InitializeComponent();    //Initialize Components.
22         }
23
24         private void Form1_Load(object sender, EventArgs e)
25         {
26             this.r = new Random();    //Random-type variable, r loads.
27         }
28     }
29 }
```

Εικόνα του κώδικα του 2ου εκτελέσιμου 1.

Ακόμη, όπως φαίνεται παραπάνω έγινε αρχικοποίηση μιας λίστας ακέραιων αριθμών, ints\_list και μιας τυχαίας μεταβλητής (Random-type variable), r, η οποία φορτώνει ταυτόχρονα με, την ίδια, την φόρμα.

### 3.3 Η ΣΥΝΑΡΤΗΣΗ BUTTON1\_CLICK.

Σε αυτήν την ενότητα θα περιγράψει ο αλγόριθμος που, αναφέρθηκε παραπάνω, ο οποίος ενεργοποιείται με το πάτημα του κουμπιού button1.

```

29 private void button1_Click(object sender, EventArgs e)
30 {
31     Stopwatch watch = Stopwatch.StartNew();           //A Stopwatch-type variable, watch which, works as a counter in this case.
32     int num_input;                                   //An int-type variable, num_input, which keeps the input of the textBox1.
33
34     try
35     {
36         num_input = (int)Convert.ToInt32(this.textBox1.Text); //Every input in textBox1 converts to 32-bit integer.
37     }
38     catch //Exception.
39     {
40         return;
41     }
42     this.richTextBox1.Clear(); //At every button's click, clears the richTextBox1 content.
43
44     int[] ints_array = new int[num_input]; //An int-type array, ints_array which, contains the num_input variable.
45
46     this.ints_list = new List<int>(num_input); //At every button's click, ints_list contains the num_input.
47
48     for (int i = 0; i < num_input; i++) //A for-loop.
49     {
50         this.ints_list.Add(i); //Adds every number, smaller than num_input to ints_list.
51     }
52
53     for (int j = 0; j < num_input; j++) //A for-loop.
54     {
55         int index = this.r.Next(0, num_input - j); //An int-type variable, index which, randomly selects as many index as the value of num_input.
56         int num1 = this.ints_list[index]; //An int-type variable, num1 which, equals to the index of the ints_list.
57         ints_array[j] = num1 + 1; //Int-type array, ints_array equals to num1 + 1, at every loop.
58         this.ints_list.Remove(num1); //Removes the selected number from ints_list, at every loop.
59     }
60
61     watch.Stop(); //Stopwatch-type variable, watch stops.
62     this.label3.Text = watch.ElapsedMilliseconds.ToString(); //Stopwatch-type variable, watch appears at label3 as a String.
63
64     for (int k = 0; k < ints_array.Length; k++) //A for-loop.
65     {
66         int num2 = ints_array[k]; //An int-type variable, num2 which, equals to ints_array.
67         this.richTextBox1.AppendText(num2.ToString() + Environment.NewLine); //At every button's click, num2 (i.e. ints_array) appears as a String at richTextBox1.
68     }
69 }
70
71

```

Εικόνα του κώδικα του 2ου εκτελέσιμου 2.

Αρχικά, ορίζονται μια μεταβλητή τύπου Stopwatch (Stopwatch-type variable), watch υπεύθυνη για την καταμέτρηση του χρόνου ολοκλήρωσης της διαδικασίας, μιας ακέραιας μεταβλητής num\_input η οποία, δέχεται ως όρισμα το περιεχόμενο του textBox1 από το χρήστη, το οποίο (δηλαδή το όρισμα του textBox1) μετατρέπεται μετέπειτα σε έναν 32-bit ακέραιο. Ακόμη, ορίζεται ένας πίνακας ακέραιων, ints\_array που περιέχει τον ακέραιο num\_input. Έπειτα, υπάρχουν τρεις διαδικασίες επανάληψης for (for-loops): Η πρώτη, προσθέτει κάθε αριθμό μικρότερο από τον αριθμό εισαγωγής του χρήστη (num\_input) στη λίστα ακέραιων, ints\_list, η οποία περιέχει ήδη τον αριθμό εισαγωγής του χρήστη (num\_input). Η δεύτερη, μέσω μιας νέας ακέραιου τύπου μεταβλητής index, επιλεγεί τυχαία δείκτες (δηλαδή θέσεις) μικρότερες (σε αριθμό) από τον αριθμό εισαγωγής. Οι προαναφερθέντες, τυχαία επιλεγμένες θέσεις και οι αριθμοί που βρίσκονται σε αυτές μέσα

στη λίστα ακέραιων `ints_list` ισούνται με μια νέα ακέραιου τύπου μεταβλητή `num1`, η οποία αφού αυξηθεί κατά ένα προστίθεται στον πίνακα ακέραιων `ints_array` και έπειτα αφαιρείται από την λίστα ακέραιων `ints_list` ώστε, να μην ξανά επιλεγεί ο ίδιος αριθμός. Η τρίτη και τελευταία διαδικασία επανάληψης, ορίζει μια νέα ακέραια μεταβλητή `num2` η οποία, ως όρισμα της ανά επανάληψη έχει κάθε αριθμό του πίνακα ακέραιων `ints_array`. Έπειτα, αυτή η νέα μεταβλητή (`num2`) εμφανίζεται με τη μορφή αλφαριθμητικού στο πλαίσιο του `richTextBox1`.

Τέλος, μετρά την ολοκλήρωση των 2 πρώτων διαδικασιών επανάληψης και πριν την έναρξη της τρίτης διαδικασίας επανάληψης, η `Stopwatch` τύπου μεταβλητή, `watch` σταματά και ο χρόνος που έχει μετρηθεί μέχρι εκείνη τη στιγμή, θα εμφανίζεται στο `label3` με την μορφή των μιλιδευτερόλεπτων (`ms`, `millisecond`).