

Drukuj

Zamknij

Generator liczb Dużego Lotka w C#

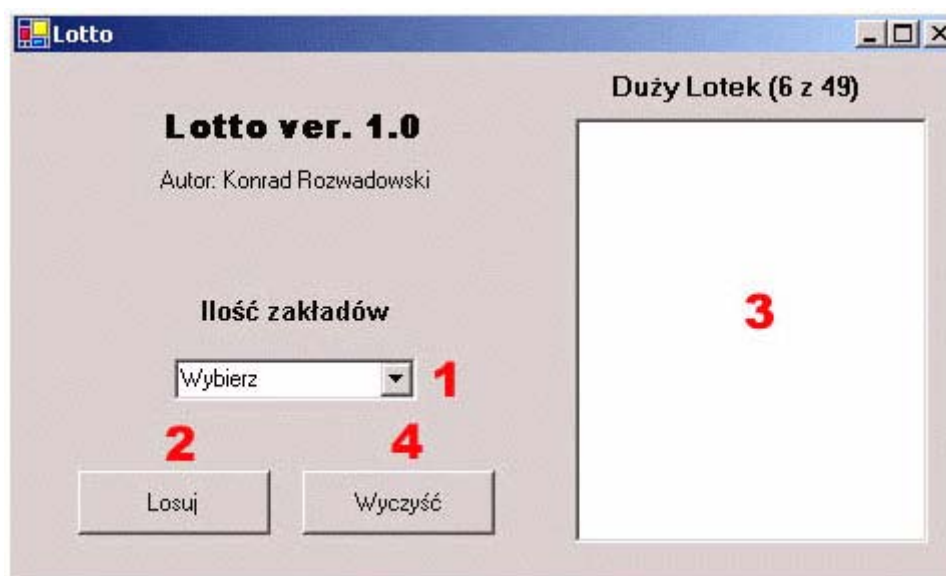
2003-12-09 21:59:34

Konrad Rozwadowski - Katedra Systemów Równoległych i Sieci Komputerowych, Polsko-Japońska Wyższa Szkoła Technik Komputerowych w Warszawie

Aplikacja ma pomóc grającym w Dużego Lotka powinna więc być prosta w swojej budowie oraz losować wygrywające numerki :). Program będzie działał na zasadzie Chybił Trafił, ustawiamy tylko ile zakładów chcemy wylosować, a resztą zajmuje się już program.

Krótki instruktarz działania:

- wybieramy ilość zakładów (od 1 do 10) (nr. 1)
- naciskamy na przycisk Losuj (nr. 2)
- a w okienku listBoxa wyświetlą się numerki (nr. 3)
- wygenerowane liczby możemy usunąć za pomocą przycisku Wyczyść (nr. 4)



Krok po kroku jak napisać program w C#

Potrzebne będą nam dwa przyciski, jeden listBox oraz comboBox. Ustawiamy na formatce i odpowiednio podpisujemy. Pierwszą rzeczą jaką trzeba wykonać to umieszczenie w comboBoxie ilości zakładów które będą się wyświetlać po naciśnięciu. Najlepiej jest to zrobić wtedy gdy formatka się ładuje, naciskamy na formatkę dwa razy i wpisujemy:

```
private void Form1_Load(object sender, System.EventArgs e)
{
    //wstawiamy
    System.Object[] ItemObject = new System.Object[10];
    for (int i = 0; i <= 9; i++)
    {
        ItemObject[i] = i+1; // +1 ponieważ zakłady są od 1 do 10
    }
    comboBox1.Items.AddRange(ItemObject);
}

```

W napisanym kodzie ustawiamy tablicę 10 elementową która będzie zawierała numery zakładów od 1 do 10, gdzie następnie zostanie wyświetlona w comboBoxie za pomocą metody AddRange().

Następnie wygenerujemy zdarzenie dla przycisku Losuj, klikamy dwa razy i piszemy:

```
private void button1_Click(object sender, System.EventArgs e)
{
    //wstawiamy
    int selectedIndex = comboBox1.SelectedIndex;
}

```

```

        for(int i=0;i<selectedIndex+1;i++)
        {
            Dodaj(i);
        }
    }

```

W zmiennej selectedIndex będzie przechowywany numer indeksu wybranej ilości zakładów. Następnie w pętli for zostanie wywołana funkcja Dodaj() tyle razy ile mamy wybranych zakładów, oraz przekazujemy jej parametr „i” który będzie nam potrzebny w następnej funkcji.

Funkcja Dodaj() wygląda następująco:

```

//wstawiamy przed funkcją Main()
public void Dodaj(int i)
{
    string x = ""; //omówienie w dalszej części
    int temp; //omówienie w dalszej części

int[,] myIntArray = new int[i+1,6];
for(int j=0;j<6;j++)
{
    myIntArray[i,j]=Losuj();
}
//c.d.n.

```

Inicjujemy tablicę gdzie będą przechowywane nasze szczęśliwe numerki, nasz przekazywany parametr „i” pokazuje do którego zakładu numerki będzie należał. Wywołujemy funkcję losującą, która przekaże numerki.

Funkcja losująca wygląda następująco:

```

Random fixRand = new Random();
public int Losuj()
{
    int liczba;
    liczba = fixRand.Next()%49 + 1;
    return liczba;
}

```

Przed funkcją deklarujemy obiekt losujący, do którego będziemy się odwoływać. Funkcja musi zwrócić numerki od 1 do 49 dlatego do %49 dodajemy 1, w przeciwnym wypadku byłyby losowane liczby od 0 do 48.

Powracamy do funkcji Dodaj(). Dodajemy kod, który będzie odpowiedzialny za sprawdzenie czy numerki nie powtórzyły się w danym zakładzie.

```

//c.d.
powtorz:    for(int j=0;j<6;j++)
            {
                for(int k=0;k<6;k++)
                {
                    if(j==k)
                    {
                    }
                    else
                    {
                        if(myIntArray[i,j]==myIntArray[i,k])
                        {
                            myIntArray[i,j]=Losuj();
                            goto powtorz;
                        }
                    }
                }
            }
//c.d.n.

```

Instrukcja goto, posłużyła mi do ponownego sprawdzenia numerków w zakładzie, jeżeli numerki się powtórzyły. W dalszej części zastosowałem sortowanie bąbelkowe, które ułoży numerki od wartości małej do dużej. Wszystkie numerki są ze sobą porównywane i zamieniane miejscami, jeżeli dalszy jest mniejszy od wcześniejszego.

```

//c.d.
for(int j=0;j<5;j++)
{
    for(int k=j+1;k<6;k++)
    {
        if(myIntArray[i,j]>myIntArray[i,k])
        {

```

```
        temp = myIntArray[i,j];  
        myIntArray[i,j]= myIntArray[i,k];  
        myIntArray[i,k] = temp;  
    }  
}  
//c.d.n.
```

Następnie, jeżeli posortowaliśmy numerki oraz sprawdziliśmy czy się nie powtarzają w zakładzie możemy przejść do wyświetlenia.

```
for(int j=0;j<6;j++)  
{  
    x = x + " " + myIntArray[i,j].ToString();  
}  
    listBox1.Items.Add(x);  
} // zamykamy funkcję Dodaj()
```

Ostatnią czynnością jest dodanie zdarzenia do przycisku Wyczyść, klikamy dwa razy i dodajemy kod:

```
private void button2_Click(object sender, System.EventArgs e)  
{  
    listBox1.Items.Clear();  
}
```

Metoda Clear() oczyści nam listBoxa.

KONRAD ROZWADOWSKI

Drukuj

Zamknij