

Wrocław, 10 stycznia 2020r.

Mikołaj Baran, 241128
Jakub Aniszewski, 241133

prowadzący: Dominik Żelazny

Laboratorium Urządzeń Peryferyjnych

Ćwiczenie 8 - Drukowanie kodów paskowych

1 Cel ćwiczenia

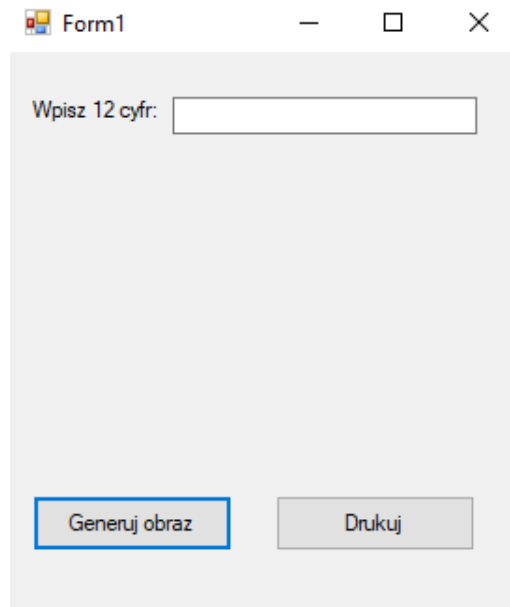
1. Korzystając z instrukcji obsługi drukarki STAR SP300 lub drukarki LaserJet 6, zapoznać się z zasadami jej obsługi i sterowania
2. Napisać program do drukowania kodów kreskowych w standardzie EAN13 wykorzystując drukarkę:
 - wprowadzić kod (cyfrowo - bez ostatniej cyfry)
 - wydrukować kod kreskowy w standardzie EAN13
 - wydrukować cyfry kodu

2 Wstęp

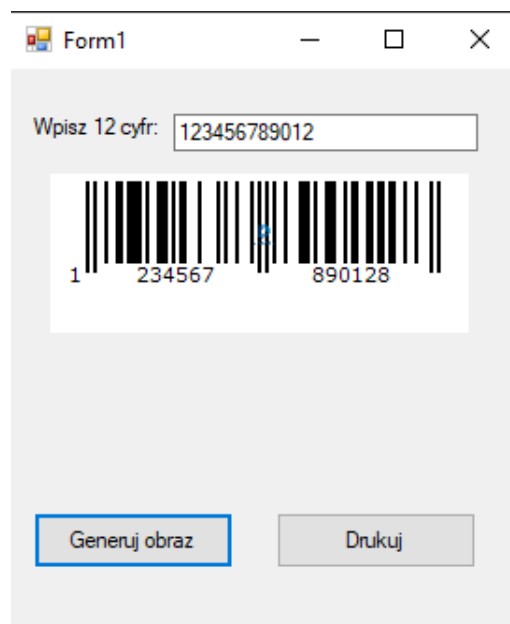
Kody paskowe są graficzną formą reprezentowania informacji dla czytników elektronicznych. Są szeroko stosowane w logistyce. Odczyt odbywa się poprzez naświetlenie ciągu pasków - jasne odbijają światło, ciemne pochłaniają. Informacja o odbitych i pochłoniętych obszarach jest odczytywana przez światłoczuły sensor i tłumaczone na cyfry. Kod jest zakończony cyfrą sumy kontrolnej. Aby ją obliczyć należy dodać wszystkie liczby z pozycji nieparzystych i ich sumę pomnożyć przez 3. Następnie zsumować wszystkie cyfry z pozycji parzystych i dodać do poprzedniej sumy. Cyfra kontrolna jest różnicą do poprzednio wyliczonej sumy do najbliższej liczby podzielnej przez 10.

3 Opis programu

Do wykonania zadania wykorzystaliśmy język C# oraz bibliotekę *Spire.Barcode*.



Aby wygenerować kod paskowy należy wprowadzić 12 cyfr. W przypadku błędnej ilości podanych cyfr program poinformuje nas o tym komunikatem.



Po wygenerowaniu kodu pojawia się jego graficzna reprezentacja. Aby wydrukować kod należy wcisnąć przycisk **Drukuj** i wybrać urządzenie, z którego chcemy skorzystać.

4 Najważniejsze funkcje w programie

4.1 Funkcja generująca kod paskowy

Funkcja *buttonGenerate_Click()* pobiera ciąg liczb podany przez użytkownika. Następnie sprawdza czy wpisano poprawną ilość cyfr (12). W następnym kroku tworzony jest nowy obiekt *BarCo-*

deGenerator, który tworzy kod typu **EAN13**. Wygenerowany kod jest następnie konwertowany do postaci obrazka i umieszczany w obiekcie *PictureBox* interfejsu graficznego programu.

```
1 private void buttonGenerate_Click(object sender, EventArgs e) {
2     var code = text_box_bar_codedigits.Text.Trim();
3     if (text_box_bar_codedigits.TextLength < 12)
4         MessageBox.Show("Nieprawidłowa ilość cyfr!
5         Wpisz jeszcze {12 - text_box_bar_codedigits.TextLength}");
6     else if (text_box_bar_codedigits.TextLength > 12)
7         MessageBox.Show("Nieprawidłowa ilość cyfr!
8         0 {text_box_bar_codedigits.TextLength - 12} cyfr za dużo");
9
10
11
12     var bar_code_generator = new BarcodeGenerator(new BarcodeSettings()
13     {
14         Type = BarcodeType.EAN13,
15         BackColor = Color.White,
16         ForeColor = Color.Black,
17
18         Data = code,
19         ShowText = true,
20         ShowTextOnBottom = true,
21         UseChecksum = CheckSumMode.ForceEnable,
22         TextAlignment = StringAlignment.Center,
23     });
24
25
26     barcode_img = bar_code_generator.GenerateImage();
27     barcode_img = cropImage(barcode_img, new Rectangle(0, 15, barcode_img.Width,
28         barcode_img.Height - 15));
29
30     picture_box_bar_code.Image = barcode_img;
31     picture_box_bar_code.Update();
32     picture_box_bar_code.Visible = true;
33 }
```

4.2 Funkcja odpowiedzialna za drukowanie

Funkcja sprawdza czy istnieje obraz kodu paskowego do wydrukowania. Jeśli tak to bitmapa jest konwertowana na tablicę bajtów. Następnie tworzony jest nowy obiekt *PrintDocument*, który wskazuje na dokument, który ma zostać wydrukowany. W kolejnym kroku wyświetlane jest okno dialogowe, w którym użytkownik może wybrać drukarkę.

```
1 private void buttonPrintBarCode_Click(object sender, EventArgs e) {
2     if (barcode_img == null) {
3         MessageBox.Show("obrazek jest pusty");
4         return;
5     }
6
7     ImageConverter _imageConverter = new ImageConverter();
8     byte[] xByte = (byte[])_imageConverter.ConvertTo(barcode_img, typeof(byte[]));
9
10    PrintDocument pd = new PrintDocument();
11    pd.PrintPage += new PrintPageEventHandler(PrintPage);
12 }
```

```
12
13     PrintDialog pdi = new PrintDialog();
14     pdi.Document = pd;
15     if (pdi.ShowDialog() == DialogResult.OK) {
16         pd.Print();
17     }
18     else {
19         MessageBox.Show("drukowanie anulowane");
20     }
21 }
```

4.3 Funkcja ustawiająca parametry obrazka

Funkcja ustawia rozmiary obrazka kodu paskowego, który ma zostać wydrukowany, a następnie zapisuje go w zmiennej *image*.

```
1     private void PrintPage(object o, PrintPageEventArgs e) {
2         try {
3             Rectangle m = e.MarginBounds;
4             m.Height = 100;
5             m.Width = 300;
6
7             System.Drawing.Image image = barcode_img;
8             e.Graphics.DrawImage(image, m);
9         }
10        catch (Exception ex)
11        {
12            throw ex;
13        }
14    }
```

5 Wnioski

Podczas przeprowadzonych zajęć laboratoryjnych mieliśmy okazję zapoznać się z sposobem tworzenia kodów paskowych w języku C#. Zdecydowaliśmy się skorzystać z biblioteki *Spire.Barcode*. Udało się dzięki niej szybko i bezproblemowo wykonać wszystkie punkty zadania. Zabrakło jedynie implementacji metody weryfikacji wyliczonej przez bibliotekę cyfry kontrolnej kodu paskowego.