

Urządzenia Peryferyjne

Laboratorium nr 6 - Drukarka kodów kreskowych

Maciej Bronikowski 248838
Michał Droń 248832

1 Wstęp

1.1 Wstęp teoretyczny

European Article Number (EAN) to standard opisujący kody kreskowe i ich numerowanie używane w międzynarodowym handlu. Numer opisuje typ produktu, pakowanie oraz producenta.

Najpopularniejszym standardem jest EAN-13, który zawiera trzycyfrowy prefiks GS1 (oznacza on kraj rejestracji produktu lub specjalny typ produktu), kod producenta, kod produktu oraz cyfrę kontrolną.

1.1.1 Kompozycja kodu EAN-13

Prefiks GS1 - Dwie lub trzy pierwsze cyfry kodu EAN-13 identyfikują kraj członkowski organizacji GS1 do której należy producent. Nie zawsze jest to kraj produkcji.

Kod producenta to unikalny kod nadawany każdemu producentowi przez organ wskazany przez prefiks GS1. Do niedawna długość jego była stała i wynosiła 5 cyfr. Wiele producentów w swojej ofercie posiada o wiele mniej niż 100 000 produktów, co prowadziło do marnowania się dużej ilości kodów. Z tego powodu wprowadzono kody producentów o zmiennej długości.

Kod produktu przydzielany jest produktom przez producenta. Wraz z kodem producenta powinien mieć on długość 9-10 cyfr w zależności od długości prefiksu GS1.

Cyfra kontrolna służy sprawdzeniu czy kod został poprawnie zeskanowany. System EAN-13 jest w stanie wykryć wszystkie błędy jednocyfrowe oraz 90% błędów transpozycji.

1.1.2 Obliczanie cyfry kontrolnej

Suma kontrolna obliczana jest poprzez dodanie iloczynów kolejnych cyfr kodu EAN-13 oraz ich wag. Wagi cyfr mają naprzemienną wartość 3 oraz 1 i przydzielane są w taki sposób, żeby ostatnia cyfra kodu EAN (która nie jest cyfrą kontrolną) miała wartość 3. W praktyce oznacza to, że liczone od końca nieparzyste pozycje mają zawsze wagę 3, a parzyste - wagę 1.

Cyfra kontrolna to jednocyfrowa wartość dopełnienia otrzymanej sumy w taki sposób, by reszta tej sumy wynosiła 10.

$$n_0 = \left(10 - \left(\left(\sum_{i=1}^6 3n_{2i-1} + n_{2i} \right) \bmod 10 \right) \right) \bmod 10$$

Dla przykładu:



Rysunek 1: Przykładowy kod EAN-13.

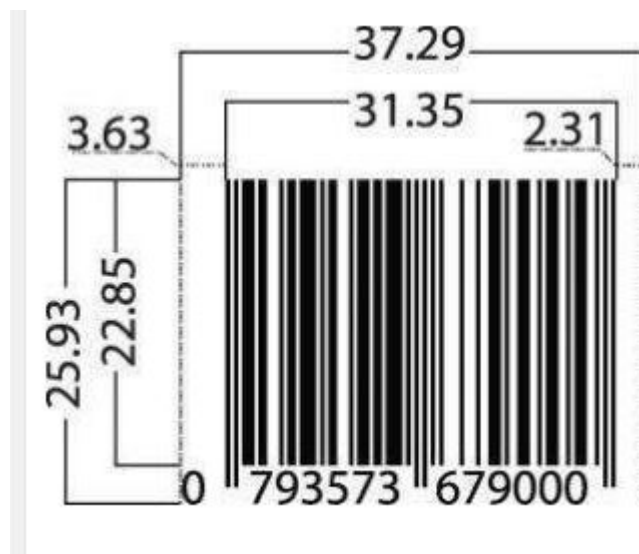
Pozycja	12	11	10	9	8	7	6	5	4	3	2	1
Kod	4	0	0	6	3	8	1	3	3	3	9	3
Waga	1	3	1	3	1	3	1	3	1	3	1	3
Suma częściowa	4	0	0	18	3	24	1	9	3	9	9	9

A więc:

$$n_0 = (10 - (89 \bmod 10)) \bmod 10 = (10 - 9) \bmod 10 = 1$$

1.1.3 Wymiary kodu kreskowego

Kod EAN-13 ma ściśle określone wymiary.



Rysunek 2: Wymiary kodu EAN-13.

Szerokość kodu: 31.35mm

Lewy margines: 3.63mm

Prawy margines: 2.31mm

Szerokość całkowita: 37.29mm

Wysokość kodu: 22.85mm

Wysokość kodu z cyframi: 25.93mm

Kod kreskowy EAN-13 można przeskalować ze współczynnikiem skalowania w zakresie $[0,8; 2,0]$. Skalowanie spowoduje powstanie kodu kreskowego między minimalnym dopuszczalnym rozmiarem 29,83 mm szerokości i 20,74 mm wysokości a maksymalnym dopuszczalnym rozmiarem 74,58 mm szerokości i 51,86 mm wysokości.

1.1.4 Struktura kodu kreskowego

Kod kreskowy jest podzielony na 113 modułów jednakowej szerokości, z których każdy może być czarny (reprezentując logiczne 1) lub biały (reprezentując logiczne 0). Wskazując od lewej kolejne sekcje to:

- 11 modułów o wartości 0 przedstawiających "Quiet Zone", ułatwiającą odczyt kodu skanerom
- 3 moduły markera startu kodu o wartości (101)
- 42 moduły wskazujące wartość pierwszych 7 cyfr kodu kreskowego
- 5 modułów markera środka kodu o wartości (01010)
- 42 moduły wskazujące wartość ostatnich 6 cyfr kodu kreskowego
- 3 moduły markera stopu o wartości (101)
- 7 modułów o wartości 0 przedstawiających "Quiet Zone" z prawej strony

Aby zakodować 13 cyfrową liczbę, cyfry podzielone są na 3 grupy. W pierwszej grupie znajduje się jedynie pierwsza cyfra kodu. Druga grupa składa się z kolejnych 6 cyfr kodu i jest zakodowana używając wzoru parzystego (oznaczonego literą G) oraz nieparzystego (oznaczonego literą L). Trzecia grupa składająca się z 6 ostatnich cyfr zakodowana jest używając wzoru oznaczonego literą G, która jest negacją kodu oznaczonego literą L. Pierwsza cyfra zakodowana jest poprzez użycie odpowiednich kodowań (L oraz G) dla kolejnych cyfr znajdujących się w drugiej grupie, których kolejność można zobaczyć na poniższej tabeli.

Struktura EAN-13:

Pierwsza cyfra	Pierwsza grupa 6 cyfr	Druga grupa 6 cyfr
0	LLLLLL	RRRRRR
1	LLGLGG	RRRRRR
2	LLGGLG	RRRRRR
3	LLGGGL	RRRRRR
4	LGLLGG	RRRRRR
5	LGGLLG	RRRRRR
6	LGGGLL	RRRRRR
7	LGLGLG	RRRRRR
8	LGLGGL	RRRRRR
9	LGGLGL	RRRRRR

Kodowanie cyfr:

Cyfra	Kod L	Kod G	Kod R
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

1.2 Cel ćwiczenia

Celem ćwiczenia jest stworzenie aplikacji drukującej kody kresowe EAN-13.

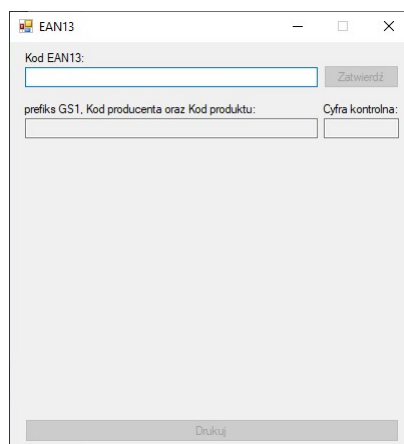
2 Implementacja

2.1 Działanie aplikacji

W ramach ćwiczeniach wykonano implementację aplikacji desktopowej, która służy do drukowania kodów kreskowych EAN-13. Aplikacja posiada następujące funkcje:

- Generowanie cyfry kontrolnej na podstawie 12 pierwszych cyfr kodu EAN-13
- Generowanie kodu kreskowego
- Drukowanie kodu kreskowego

Aplikacja została napisana w języku C#. Graficzny interfejs został przygotowany z użyciem interfejsu Windows Forms platformy .NET Framework.



Rysunek 3: GUI aplikacji po włączeniu

2.2 Klasa Program.cs

Jest to klasa główna programu. W metodzie statycznej Main ustawiane są podstawowe parametry aplikacji, oraz uruchamiana jest standardowa pętla aplikacji z wskazaniem na główny formularz klasy Form1.cs, który będzie wyświetlony.

2.3 Klasa Form1.cs

Klasa Windows Forms. Odpowiedzialna jest za GUI aplikacji. Znajdują się w niej wszystkie elementy potrzebne do obsługi formularza aplikacji, takie jak metody zdarzeń wywoływanych podczas manipulacji elementami formularza.

2.3.1 Metody klasy

Form1_Load — Metoda uruchamiana podczas inicjalizacji formularza. Dezaktywuje elementy, które są aktywne dopiero po wczytaniu kodu.

buttonAccept_Click — Tworzy nowy obiekt klasy Ean13, pobierając 12 pierwszych cyfr kodu EAN-13 z textboka. Wyświetla kod kreskowy na formularzu i usuwa stary, jeśli takowy został wcześniej załadowany.

textBoxCode_KeyPress — Zapobiega wpisywaniu znaków innych niż cyfry do pola tekstowego z kodem.

textBoxCode_TextChanged — Uaktywnia przycisk "zatwierdź" po wpisaniu 12 cyfr.

buttonPrint_Click — Obsługuje drukowanie wygenerowanego kodu kreskowego.

Kod drukowania kodu kreskowego znajdującego się w metodzie **buttonPrint_Click** wygląda następująco:

```
if (Ean13 == null || Ean13.GetBarcodeBitmap() == null)
    MessageBox.Show("Brak przypisanego obrazka",
        "Błąd",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error
    );

var printDocument = new PrintDocument();
var printDialog = new PrintDialog();
printDocument.PrintPage += (s, arg) =>
{
    var mmPerInch = 25.4f;
    var eanSize = Ean13.GetCodeSize();
    var barCode = Ean13.GetBarcodeBitmap();

    arg.Graphics.DrawImage(Ean13.GetBarcodeBitmap(),
        new RectangleF(
            arg.PageSettings.Margins.Left,
            arg.PageSettings.Margins.Top,
            100 * eanSize.X / mmPerInch,
            100 * eanSize.Y / mmPerInch));
};
printDialog.Document = printDocument;
if (printDialog.ShowDialog() == DialogResult.OK) printDocument.Print();
```

Według standardu EAN-13 należy zapewnić dokładny rozmiar kodu kreskowego, dlatego przed wydrukowaniem należy przeskalować wygenerowaną bitmapę kodu kreskowego. Napisana przez nas klasa Ean13 zwraca rozmiar z użyciem zdefiniowanej skali w mm, dlatego należy przekonwertować wartość do jednostki 1/100 cala.

2.4 Klasa Ean13.cs

2.4.1 Statyczne atrybuty

_EncodingEvenLeft — Tablica sekwencji modułów dla cyfr 0-9 według kodowania oznaczonego literą G

_EncodingOddLeft — Tablica sekwencji modułów dla cyfr 0-9 według kodowania oznaczonego literą L

_EncodingRight — Tablica sekwencji modułów dla cyfr 0-9 według kodowania oznaczonego literą R

_LeftEncoding — Tablica kolejnych kodowań pierwszej grupy 6 cyfr, których należy użyć aby przechować informację na temat pierwsze cyfry

_StartStopMarker — Kodowanie modułów dla znaczników początku i końca kodu kreskowego

_CenterMarker — Kodowanie modułów dla znacznika środka kodu kreskowego

_LeftQuiteZone — Kodowanie modułów dla "Quite Zone" (11 modułów o wartości 0)

_Width — Podstawowa szerokość kodu kreskowego

_Height — Podstawowa wysokość kodu kreskowego

_DigitHeight — Podstawowa wysokość między końcem krótszego modułu, a końcem kodu kreskowego (miejscą dla cyfry)

_Scale — Skalowanie kodu kreskowego (domyślnie 1)

_DigitSize — Wielkość czcionki użytej do zapisanie cyfr kodu kreskowego

2.4.2 Atrybuty

BarCode — 12 pierwszych cyfr kodu kreskowego

Checksum — wyliczona ostatnia cyfra kontrolna

_barCodeImage — ostatnio wygenerowana bitmapa z kodem kreskowym

2.4.3 Metody

Ean13 (konstruktor) — Zapisuje przekazany kod bez ostatniej cyfry kontrolnej, oraz oblicza cyfrę kontrolną.
CalculateControl — Metoda obliczająca i zwracająca wartość cyfry kontrolnej na podstawie przechowywanej sekwencji kodu

EncodeCode — Metoda zwraca ciąg znaków (wartości 0 lub 1) zakodowanej sekwencji 107 modułów. (7 ostatnich modułów "Quite Zone" nie jest wymagane, ponieważ cały kod kreskowy będzie miał białe tło i wymagany rozmiar)

GenerateBarCode — Na podstawie ciągu znaków zwróconych przez metodę **EncodeCode**, generuje i zwraca bitmapę kodu kreskowego

GetCodeSize — zwraca informację na temat rozmiaru kodu kreskowego po przeskalowaniu

Obliczanie cyfry kontrolnej:

```
var checksum = 0;
for (var i = 1; i < 12; i += 2)
    checksum += Convert.ToInt32(BarCode.Substring(i, 1));

checksum *= 3;
for (var i = 0; i < 12; i += 2)
    checksum += Convert.ToInt32(BarCode.Substring(i, 1));

return (10 - checksum % 10) % 10;
```

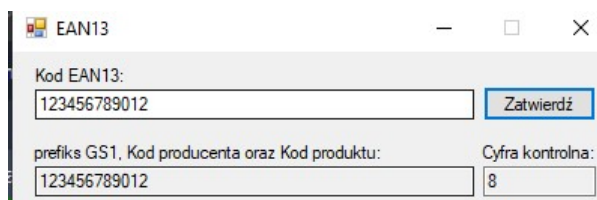
Wyznaczanie ciągu znaków kodowania:

```
var encoded = "";
var encoding = _LeftEncoding[(int) char.GetNumericValue(BarCode[0])];
// Quite zone
encoded += _LeftQuiteZone;
// Marker pocztku
encoded += _StartStopMarker;
// lewa czesc kodu
for (var i = 1; i <= 6; i++) {
    var digit = (int) char.GetNumericValue(BarCode[i]);
    if (encoding[i - 1] == 'L')
        encoded += _EncodingOddLeft[digit];
    else
        encoded += _EncodingEvenLeft[digit];
}
// Marker srodka
encoded += _CenterMarker;
// prawa cz kodu
for (var i = 7; i < 12; i++)
    encoded += _EncodingRight[(int) char.GetNumericValue(BarCode[i])];

encoded += _EncodingRight[(int) char.GetNumericValue(CheckSum[0])];
// Marker konca
encoded += _StartStopMarker;
return encoded;
```

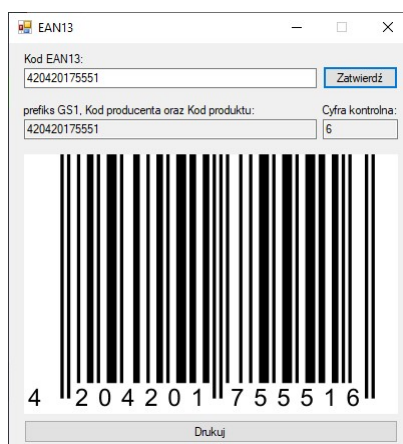
3 Realizacja zadań

3.1 Generowanie cyfry kontrolnej



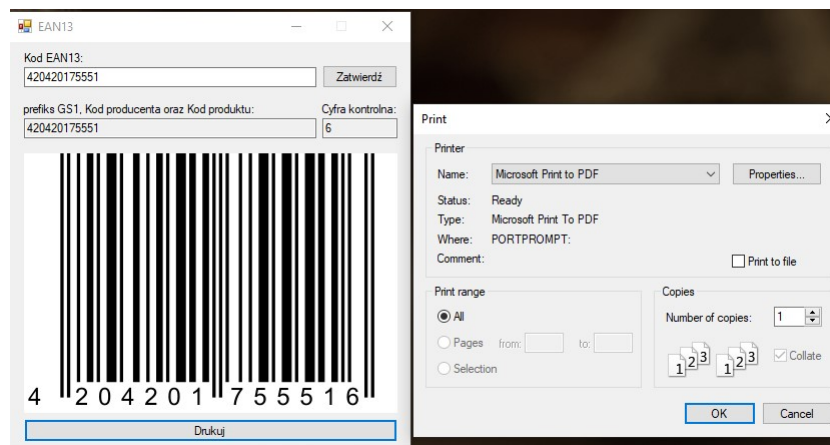
Rysunek 4: Aplikacja generuje cyfrę kontrolną kodu EAN-13 po wpisaniu pierwszych 12 cyfr kodu.

3.2 Generowanie kodu kreskowego



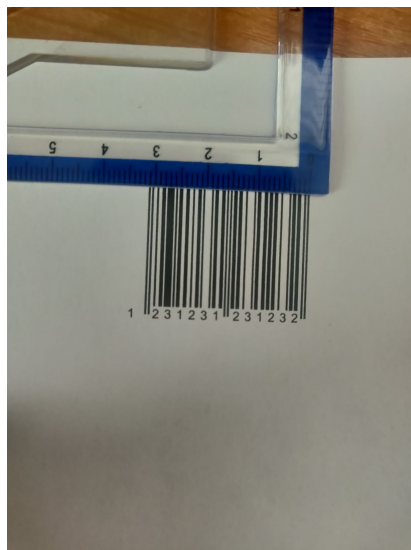
Rysunek 5: Aplikacja generuje kod kreskowy EAN-13 po wpisaniu pierwszych 12 cyfr kodu.

3.3 Drukowanie



Rysunek 6: Możliwe jest wydrukowanie wygenerowanego kodu.

3.4 Wymiary kodu kreskowego



Rysunek 7: Po wydrukowaniu kod kreskowy posiada odpowiednie wymiary.

4 Podsumowanie

W wyniku realizacji ćwiczeń laboratoryjnych stworzona została aplikacja desktopowa do drukowania kodów kreskowych EAN-13. Posiada ona stosunkowo szeroką funkcjonalność. Podczas tworzenia aplikacji nie zostały napotkane żadne problemy, które uniemożliwiłyby realizację ćwiczeń.

Pełny kod aplikacji znajduje się w serwisie GitHub:

<https://github.com/JeLLek1/UrzadzeniaPeryferyjne> w folderze Lab_8