

Urządzenia peryferyjne
Ćwiczenie 2. Kody kreskowe (EAN-13)

Autorzy: Matsvei Samakar 274002,
Tomasz Piwoda 272941
Prowadzący: dr inż. Dominik Żelazny

Temat i zadanie do zrealizowania:

Tematem ćwiczenia były kody kreskowe, dokładnie kod EAN-13. Mieliśmy napisać program generujący (dla wprowadzanej sekwencji cyfr) kod EAN-13 i dalej odczytać ten kod za pomocą skanera.

Wstęp teoretyczny:

Wymiarowość:

Kody kreskowe można podzielić na różne rodzaje w zależności od liczby wymiarów, w których są zapisywane:

- **Kody jednowymiarowe:** są to tradycyjne kody kreskowe, które składają się z pasków o różnej szerokości, zazwyczaj w kolorze czarnym i białym. Przykładem jest kod EAN-13.
- **Kody dwuwymiarowe:** dzielą się na dwa typy:
 - a) **Piętrowe** – kody jednowymiarowe, ułożone jeden nad drugim, jak stos pasków (np. PDF417)
 - b) **Matrycowe** – kody, które mają formę siatki, gdzie dane są zapisywane za pomocą czarnych i białych punktów. Przykładem jest kod QR.
- **Kody trójwymiarowe:** są to zazwyczaj kody jednowymiarowe wytłoczone na różnego rodzaju powierzchni. Wyglądają jak wypukłe kody jednowymiarowe.
- **Kody złożone:** to połączenie kodów jedno- i dwuwymiarowych, które mogą mieć cechy obu rodzajów (np. Aztec Mesas, RSS).

Modułowość:

W kodach kreskowych modułem jest podstawowy element (najmniejsza kreska lub przerwa). Modułowość określa, ile modułów przypada na jeden kodowany znak.

Zakres kodowania:

Oznacza, jakie znaki lub dane mogą być zakodowane w danym kodzie. Na przykład w EAN-13 kodowane są tylko cyfry (0-9), natomiast QR Code obsługuje alfanumeryczne znaki (cyfry, litery, a także inne symbole).

Ciągłość:

Oznacza, jak kody są połączone i jak są odczytywane. Ciągłość może dotyczyć tego, czy kody są zwarte i odczytywane w całości, bez przerw.

Długość:

Długość kodu określa liczbę znaków, które można zakodować. Na przykład, EAN-13 składa się z 13 cyfr, a QR Code może przechowywać tysiące znaków.

Kody EAN:

- Organizacja zarządzająca: GS1 (Global Standards One) to organizacja, która zarządza systemami standardów kodów kreskowych EAN, UPC i innymi. Organizacja jest odpowiedzialna za tworzenie, utrzymanie oraz zarządzanie standardami kodów.
- Struktura administracyjna: GS1 działa globalnie poprzez lokalne organizacje, które zarządzają przydzielaniem kodów kreskowych w różnych krajach. Każda firma, która chce używać kodów EAN, musi zarejestrować się w GS1, po czym otrzymuje zakres numerów, który może wykorzystywać do generowania swoich kodów EAN.
- Procedury uzyskiwania: aby uzyskać kody EAN, firma musi złożyć wniosek do odpowiedniego lokalnego oddziału GS1, potem otrzymuje prefiks firmy, który stanowi początek numeru EAN dla każdego produktu. Następnie firma może generować EAN kody dla swoich produktów.

Kod EAN13:

Symbolika: EAN-13 to kod kreskowy jednowymiarowy, który składa się z 13 cyfr. Kod zawiera cyfry identyfikujące kraj producenta, kod firmy, kod produktu oraz cyfrę kontrolną.

Alfabet: EAN-13 używa wyłącznie cyfr (0-9). Każda cyfra jest kodowana w postaci binarnej w systemie kresk (czarnych/białych) i przerw.

Budowa: 2-3 pierwsze cyfry to kod kraju, następne 5-7 cyfr to kod producenta, kolejne 3-5 cyfr to kod produktu, a ostatnia, 13. cyfra, to cyfra kontrolna, obliczana na podstawie pozostałych cyfr (Rys. 1)



Rys.1 Przykład kodu EAN13

Samosprawdzalność: Kod EAN-13 posiada cyfrę kontrolną, która jest obliczana algorytmem mod 10. Algorytm ten pozwala na wykrycie błędów, takich jak błędnie wprowadzone lub odczytane cyfry.

Obliczanie cyfry kontrolnej:

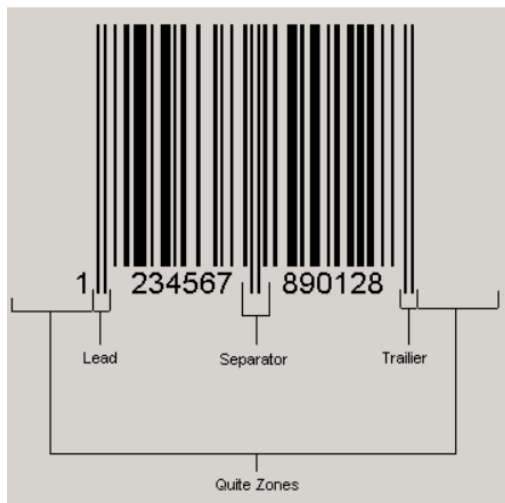
Cyfrę kontrolną (ostatnią) oblicza się według stałego algorytmu modulo 10. Należy pomnożyć wszystkie cyfry kodu (prócz kontrolnej) na nieparzystych pozycjach (pierwszą, trzecią itd., licząc pozycje cyfr od ostatniej do pierwszej) przez 3. Następnie zsumować wszystkie iloczyny wraz z cyframi na parzystych pozycjach:

$$S = \sum_{i=1}^{n-1} ((2 - (-1)^i) \cdot d_i) = 3d_1 + d_2 + 3d_3 + d_4 + \dots + ((2 - (-1)^{n-1}) \cdot d_{n-1})$$

gdzie d_i to kolejna cyfra kodu, a n to długość kodu.

Realizacja zadania:

Do wykonania tego zadania używaliśmy tablic kodowania oraz wzorów parzystości: wzór parzystości nieparzystej lewej cyfry (L-code), wzór parzystości parzystej lewej cyfry (G-code) i wzór prawej cyfry (R-code). Wzorce lewej cyfry zaczynają się od spacji, a wzór prawej cyfry zaczyna się od kresek (Rys. 2)



Rys. 2 Części kodu EAN13

1. Wprowadzenie kodu przez użytkownika:

Program najpierw prosi użytkownika o wprowadzenie 12-cyfrowego kodu. Dodatkowo program sprawdza, czy zostały wprowadzone dokładnie 12 symboli i czy wszystkie są cyframi. Po sprawdzeniu poprawności wprowadzonych danych, jest wywołana funkcja ***CalculateCheckDigit***, która oblicza cyfrę kontrolną zgodnie z powyższym wzorem i tworzy pełny 13-cyfrowy kod EAN-13.

2. Obliczanie cyfry kontrolnej

Cyfra kontrolna jest obliczana na podstawie sumy ważonej cyfr na odpowiednich pozycjach. Na parzystych pozycjach (indeksy zaczynając od 0) cyfra jest sumowana bezpośrednio, natomiast na nieparzystych pozycjach jest mnożona przez 3 przed dodaniem do sumy. Następnie obliczana jest reszta z dzielenia sumy przez 10, a

cyfra kontrolna to 0, jeśli reszta wynosi 0, lub 10 minus reszta w przeciwnym przypadku. Funkcja zwraca obliczoną cyfrę kontrolną (Rys.3)

```
118 // funkcja obliczająca cyfrę kontrolną
119 1 reference
120 private static int CalculateCheckDigit(string data)
121 {
122     int sum = 0;
123     for (int i = 0; i < data.Length; i++)
124     {
125         int digit = data[i] - '0';
126         // cyfry na nieparzystych pozycjach są mnożone przez 3
127         sum += (i % 2 == 0) ? digit : digit * 3;
128     }
129     // obliczenie cyfry kontrolnej
130     int mod = sum % 10;
131     return mod == 0 ? 0 : 10 - mod;
132 }
```

Rys. 3. Funkcja do obliczania cyfry kontrolnej

3. Generowanie wzorca binarnego kodu kreskowego:

Na podstawie pełnego kodu EAN-13, program generuje wzorzec binarny, który reprezentuje kod kreskowy (Rys.4)

Wzorzec zawiera:

- Sekwencje startowe (trailer), środkowe (separator) i końcowe (lead).
- Zakodowane cyfry za pomocą odpowiednich wzorców binarnych (LCode, GCode i RCode).

```
134 private static string GenerateBarcodePattern(string ean13Code)
135 {
136     if (ean13Code.Length != 13 || !ean13Code.All(char.IsDigit))
137         throw new ArgumentException("EAN-13 code must be 13 digits.");
138     StringBuilder result = new StringBuilder("101");
139     int firstDigitValue = ean13Code[0] - '0';
140     string parityPattern = ParityPatterns[firstDigitValue];
141     // Kodowanie cyfr po lewej stronie
142     for (int i = 1; i <= 6; i++)
143     {
144         int digit = ean13Code[i] - '0';
145         char parity = parityPattern[i - 1];
146         string encoded = parity == 'L' ? LCode[digit] : GCode[digit];
147         result.Append(encoded);
148     }
149     // wzorzec środkowy (separator)
150     result.Append("01010");
151     // kodowanie cyfr po prawej stronie
152     for (int i = 7; i <= 12; i++)
153     {
154         int digit = ean13Code[i] - '0';
155         result.Append(RCode[digit]); // dodanie zakodowanej cyfry
156     }
157     // wzorzec końcowy (Lead)
158     result.Append("101");
159     return result.ToString();
160 }
```

Rys. 4. Funkcja do generowania wzorca binarnego kodu EAN13

4. Rysowanie kodu kreskowego i zapisywanie do pliku:

Dalej tworzymy bitmapę i na jej obrazie rysujemy kod kreskowy jako czarne i białe prostokąty (paski), a poniżej rysowane są odpowiednie cyfry zgodnie z wygenerowanym kodem. Kod kreskowy jest wyśrodkowany na białym tle z odpowiednim marginesem (Rys.5)

```
56 // parametry graficzne kodu kreskowego
57 int moduleWidth = 2; // szerokość pojedynczego paska
58 int barcodeWidth = pattern.Length * moduleWidth; // całkowita szerokość kodu kreskowego
59 int barcodeHeight = 80; // wysokość samych pasków
60 int totalHeight = 130; // całkowita wysokość
61 int padding = 20; // odstęp wokół kodu kreskowego
62
63 // tworzenie bitmapy z odpowiednimi wymiarami i odstępami
64 Bitmap bitmap = new Bitmap(barcodeWidth + 2 * padding, totalHeight + 2 * padding);
65 using (Graphics g = Graphics.FromImage(bitmap))
66 {
67     g.Clear(Color.White); // wypełnienie tła na białe
68     g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.None;
69
70     // rysowanie wzoru kodu kreskowego
71     for (int i = 0; i < pattern.Length; i++)
72     {
73
74         bool isGuardBar = i < 3 || (i >= 45 && i < 50) || i >= 92;
75         int barHeight = isGuardBar ? barcodeHeight + 10 : barcodeHeight;
76         // rysowanie czarnego paska, jeśli w wzorcu jest '1'
77         if (pattern[i] == '1')
78         {
79             g.FillRectangle(Brushes.Black, padding + i * moduleWidth, padding, moduleWidth, barHeight);
80         }
81     }
82
83     // rysowanie cyfr pod kodem kreskowym
84     string firstDigit = ean13Code.Substring(0, 1); // pierwsza cyfra
85     string leftDigits = ean13Code.Substring(1, 6); // następne 6 cyfr
86     string rightDigits = ean13Code.Substring(7, 6); // 6 cyfr od prawej strony
87
88     Font font = new Font("Arial", 10);
89     StringFormat format = new StringFormat { Alignment = StringAlignment.Center };
90
91     // rysowanie pierwszej cyfry po lewej stronie
92     float firstDigitPosition = moduleWidth * 3;
93     g.DrawString(firstDigit, font, Brushes.Black, new PointF(padding + firstDigitPosition - 12, padding + barcodeHeight), format);
94
95     // rysowanie cyfr po lewej stronie
96     for (int i = 0; i < leftDigits.Length; i++)
97     {
98         float x = padding + firstDigitPosition + (7 * moduleWidth * i) + (7 * moduleWidth / 2);
99         g.DrawString(leftDigits[i].ToString(), font, Brushes.Black, new PointF(x, padding + barcodeHeight), format);
100     }
101
102     // rysowanie cyfr po prawej
103     float rightSideStart = padding + firstDigitPosition + 42 * moduleWidth + 5 * moduleWidth;
104     for (int i = 0; i < rightDigits.Length; i++)
105     {
106         float x = rightSideStart + (7 * moduleWidth * i) + (7 * moduleWidth / 2);
107         g.DrawString(rightDigits[i].ToString(), font, Brushes.Black, new PointF(x, padding + barcodeHeight), format);
108     }
109 }
110
111 // zapisz do pliku
112 string outputPath = Path.Combine("../..", "ean13barcode.png");
113 bitmap.Save(outputPath, System.Drawing.Imaging.ImageFormat.Png);
114 Console.WriteLine($"Barcode image saved to {outputPath}");
115
```

Rys. 5. Rysowanie kresek czarno-białych oraz cyfr pod kreskami

I na koniec, wygenerowany kod kreskowy jest zapisywany w formacie .PNG w naszym folderze z programem.

Wnioski

Działanie powyższego kodu zostało sprawdzone w trakcie laboratorium. Wygenerowany kod kreskowy udało się zaskanować skanerem, który zapisał go do notatnika w postaci cyfrowej.

Bibliografia

- [1] <https://www.codeproject.com/Articles/10162/Creating-EAN-13-Barcodes-with-C>
- [2] <https://pl.wikipedia.org/wiki/EAN>