Технічне Забезпечення Інформаційних Технологій

Лабораторна робота №3

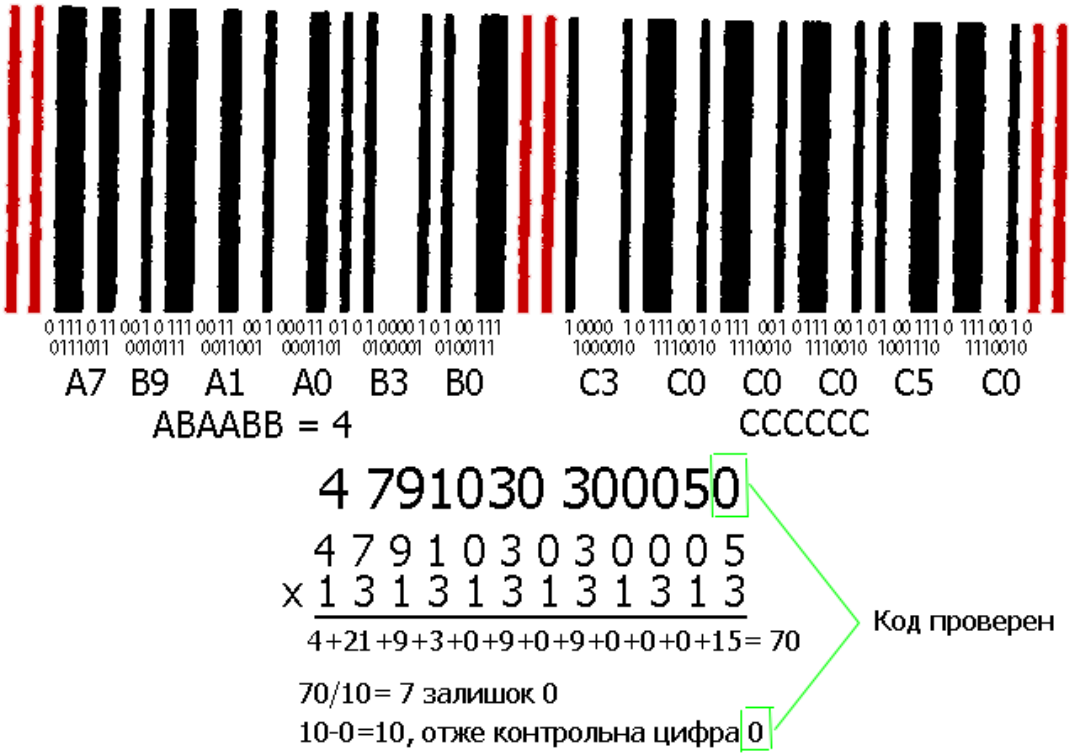
«Дослідження штрихового коду EAN 13»

Виконав:
Студент групи
ДА-21
Воронін Ігор

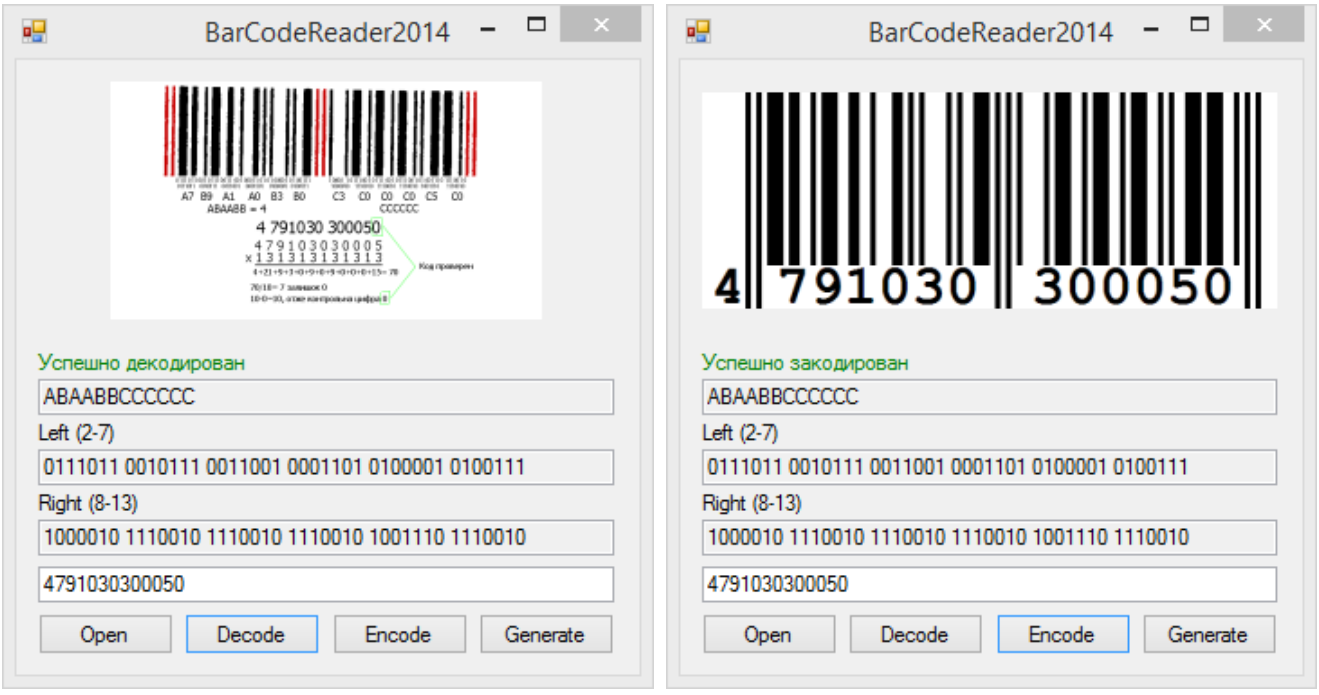
Київ – 2014

<u>Мета</u>: дослідження основних принципів побудови штрихового коду EAN

1. Результати ручного декодування штрихового коду.



2.Результати автоматичного декодування й кодування штрихового коду.



3. Складні й дуже складні випадки

4. Лістинг коду програми

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace BarCodeReader2014
{
    public partial class Form1 : Form
    {
        Bitmap Img;
        Dictionary<string, number_type> encoding_table = new Dictionary<string, number_type>();
        string[] A_codes = new string[10] { "0001101", "0011001", "0010011", "0111101", "0100011", "0110001",
            "0101111","0111011","0110111","0001011"};
        string[] first_number_code = new string[10]{"AAAAAA", "AABABB", "AABBAB", "AABBBA", "ABAABB", "ABBAAB",
                "ABBBAA", "ABABAB", "ABABBA", "ABBABA"};
        public Form1()
        {
            InitializeComponent();
            for (int i = 0; i < 10; i++)
            {
                encoding_table.Add(str_mutate(A_codes[i], 'A'), new number_type(i, 'A'));
                encoding_table.Add(str_mutate(A_codes[i], 'B'), new number_type(i, 'B'));
                encoding_table.Add(str_mutate(A_codes[i], 'C'), new number_type(i, 'C'));
            }
        }

        private void OpenBtn_Click(object sender, EventArgs e)
        {
            DialogResult Res = OpenDlg.ShowDialog();
            if (!Res.HasFlag(DialogResult.Cancel) && System.IO.File.Exists(OpenDlg.FileName))
            {
                Img = new Bitmap(OpenDlg.FileName);
                PicBox.Image = Img;
            }
```

```
}

private bool IsBlack(Color c)
{
    return (c.R * 0.3 < 120 && c.B * 0.11 < 120 && c.G * 0.59 < 120);
}

struct number_type
{
    public number_type(int n, char t)
    {
        number = n;
        type = t;
    }
    public int number;
    public char type;
}

private bool TryDecode(string full_bin_list)
{
    BinaryBox1.Clear();
    BinaryBox2.Clear();
    PatterBox.Clear();
    number_type[] n_type_array = new number_type[12];
    for (int i = 0; i < 12; i++)
    {
        if (!encoding_table.TryGetValue(full_bin_list.Substring(i * 7, 7), out n_type_array[i]))
        {
            label1.Text = "Error. There is no such bit patter in encoding table";
            return false;
        }
        if (i < 6)
            BinaryBox1.Text += full_bin_list.Substring(i * 7, 7) + " ";
        else
            BinaryBox2.Text += full_bin_list.Substring(i * 7, 7) + " ";
    }

    string encoded_type_str = "";
    for (int i = 0; i < 6; i++)
    {
        encoded_type_str += n_type_array[i].type.ToString();
    }
    for (int i = 0; i < 12; i++)
    {
        PatterBox.Text += n_type_array[i].type.ToString();
    }

    int first_number = -1;
    for (int i = 0; i < 10; i++)
    {
        if (first_number_code[i] == encoded_type_str)
        {
            first_number = i;
            break;
        }
    }
    if (first_number == -1)
    {
        label1.Text = "Error. First number wasn't recognized";
        return false;
    }

    string fullEAN = first_number.ToString();
    for (int i = 0; i < 12; i++)
    {
        fullEAN += n_type_array[i].number.ToString();
    }

    DigitBox.Text = fullEAN;

    if (get_control_sum(fullEAN) != int.Parse(fullEAN[12].ToString()))
    {
        label1.Text = "Error. Checksum is wrong";
        return false;
    }
    return true;
}

private void DrawLine(int Pos, int barwidth, Bitmap b)
{
    for (int y = 0; y < b.Height; y++)
        for (int x = barwidth * Pos; x < barwidth * Pos + barwidth; x++)
            b.SetPixel(x, y, Color.Black);
}

private void Encode(string ean)
```

```
        {
            Bitmap Img2 = new Bitmap(530, 200);
            using (Graphics g = Graphics.FromImage(Img2))
            {
                int barwidth = 5;
                int shift = 7;
                g.Clear(Color.White);
                DrawLine(1 + shift, barwidth, Img2);
                DrawLine(3 + shift, barwidth, Img2);
                DrawLine(47 + shift, barwidth, Img2);
                DrawLine(49 + shift, barwidth, Img2);
                DrawLine(95 + shift, barwidth, Img2);
                DrawLine(93 + shift, barwidth, Img2);
                PatterBox.Text = first_number_code[int.Parse(ean[0].ToString())] + "CCCCCC";
                ean += get_control_sum(ean).ToString();
                DigitBox.Text = ean;
                string bin_code = "";
                for (int i = 1; i < 7; i++)
                {
                    bin_code += str_mutate(A_codes[int.Parse(ean[i].ToString())],
first_number_code[int.Parse(ean[0].ToString())][i - 1]);
                }
                for (int i = 7; i < 13; i++)
                {
                    bin_code += str_mutate(A_codes[int.Parse(ean[i].ToString())], 'C');
                }
                BinaryBox1.Clear();
                BinaryBox2.Clear();
                for (int i = 0; i < 6; i++)
                {
                    BinaryBox1.Text += bin_code.Substring(i * 7, 7) + " ";
                }
                for (int i = 6; i < 12; i++)
                {
                    BinaryBox2.Text += bin_code.Substring(i * 7, 7) + " ";
                }
                for (int i = 0; i < 42; i++)
                {
                    if (bin_code[i] == '1') DrawLine(4 + i + shift, barwidth, Img2);
                }
                for (int i = 42; i < 84; i++)
                {
                    if (bin_code[i] == '1') DrawLine(9 + i + shift, barwidth, Img2);
                }

                g.FillRectangle(Brushes.White, (4 + shift) * barwidth, Img2.Height - 40, 42 * barwidth, 40);
                g.FillRectangle(Brushes.White, (51 + shift) * barwidth, Img2.Height - 40, 42 * barwidth, 40);
                g.DrawString(ean.Substring(1, 6), new Font("Courier New", 38, FontStyle.Bold), Brushes.Black, (5 + shift) *
barwidth, Img2.Height - 47);
                g.DrawString(ean.Substring(7, 6), new Font("Courier New", 38, FontStyle.Bold), Brushes.Black, (52 + shift) *
barwidth, Img2.Height - 47);
                g.DrawString(ean[0].ToString(), new Font("Courier New", 38, FontStyle.Bold), Brushes.Black, (-7 + shift) *
barwidth, Img2.Height - 47);
            }
            Img = Img2;
            PicBox.Image = Img2;
        }

        private void DecodeBtn_Click(object sender, EventArgs e)
        {
            if (Img == null) return;
            List<List<int>> bb = new List<List<int>>();
            for (int yi = 0; yi < Img.Height; yi++)
            {
                List<int> a = new List<int>();
                int b = 1;
                bool bc = IsBlack(Img.GetPixel(0, yi));
                for (int xi = 1; xi < Img.Width; xi++)
                {
                    bool bc2 = IsBlack(Img.GetPixel(xi, yi));
                    if (bc2 != bc)
                    {
                        a.Add(b);
                        b = 0;
                    }
                    bc = bc2;
                    b++;
                }
                if (a.Count == 60)
                bb.Add(a);
            }
            if (bb.Count == 0 )
            {
                label1.Text = "Не удалось разпознать!";
                label1.ForeColor = Color.Red;
                return;
```

```
            }
            double b_len = 0;
            foreach (List<int> a in bb)
            {
                b_len += a[1] + a[3] + a[a.Count - 1] + a[a.Count - 3];
            }
            b_len /= 4 * bb.Count;

            double w_len = 0;
            foreach (List<int> a in bb)
            {
                w_len += a[2] + a[a.Count - 3];
            }
            w_len /= 2 * bb.Count;

            int bb_inner_length = bb[0].Count;
            string full_bin_list = "";
            for (int i = 4; i <= bb_inner_length - 4; i++)
            {
                int cur_sum = 0;
                foreach (List<int> cur_line in bb)
                {
                    cur_sum += cur_line[i];
                }
                int int_bits_amount = 0;

                double average_length = (double)cur_sum / bb.Count;
                if (i % 2 == 0)
                {
                    double bit_amount = (double)average_length / w_len;
                    int_bits_amount = (int)Math.Round(bit_amount);
                    for (int j = 0; j < int_bits_amount; j++)
                    {
                        full_bin_list += "0";
                    }
                }
                else
                {
                    double bit_amount = (double)average_length / b_len;
                    int_bits_amount = (int)Math.Round(bit_amount);
                    for (int j = 0; j < int_bits_amount; j++)
                    {
                        full_bin_list += "1";
                    }
                }
            }
            if (full_bin_list.Length < 89)
            {
                label1.Text = "Не удалось разпознать!";
                label1.ForeColor = Color.Red;
                return;
            }
            full_bin_list = full_bin_list.Substring(0, 42) + full_bin_list.Substring(47, 42); //     Видалємо середні допоміжні
біти

            label1.Text = "Успешно декодирован";
            label1.ForeColor = Color.Green;
            if (!TryDecode(full_bin_list))
            {
                string s = "";
                foreach (char c in full_bin_list) s = c.ToString() + s;
                full_bin_list = s;
                label1.Text = "Успешно декодирован";
                if (!TryDecode(full_bin_list))
                {
                    label1.ForeColor = Color.Red;
                }
            }


        }


        private int get_control_sum(string source_digits)
        {
            int sum_evens = 0;
            int sum_odds = 0;
            for (int i = 0; i < 12; i++)
            {
                if ((i % 2) == 1)
                {
                    sum_evens += int.Parse(source_digits[i].ToString());
                }
                else
                {
                    sum_odds += int.Parse(source_digits[i].ToString());
```

```
            }
        }
        return (10 - ((sum_odds + 3 * sum_evens) % 10)) % 10;
    }

    private string str_mutate(string source, char param)
    {
        if (param == 'A')
            return source;
        string res = "";
        for (int i = 0; i < source.Length; i++)
        {
            char c;
            if (source[i] == '0')
            {
                c = '1';
            }
            else
            {
                c = '0';
            }
            if (param == 'B')
                res = c.ToString() + res;
            else if (param == 'C')
                res = res + c.ToString();
        }
        return res;
    }


    private void button1_Click(object sender, EventArgs e)
    {
        if (DigitBox.Text.Length > 11)
        {
            Encode(DigitBox.Text.Substring(0, 12));
            label1.ForeColor = Color.Green;
            label1.Text = "Успешно закодирован";
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Random r = new Random();
            DigitBox.Clear();
        for (int i = 0; i < 12; i++)
            DigitBox.Text += r.Next(9).ToString();
        DigitBox.Text += get_control_sum(DigitBox.Text).ToString();
        button1_Click(sender, e);
    }

}
}
```

Висновки:

Під час лабораторної роботи було ознайомлено з структурої штрих кодів EAN-13, методами їх кодування та декодування. Була розроблена программа здатна коректно зчитувати код з картинки, перевіряти його контрольну сумму, а також кодувати довільні коди у штрих код.