МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Лабораторная работа №1

По предмету «Системы искусственного интеллекта»

Распознавание циферблата часов

Выполнил студент: Пшеничный Д.О.

Факультет, группа: ФКФН, ПО(б)-81

Руководитель работы: Тормозов В.С.

Хабаровск – 2022г.

**Задание:** Для распознавания предъявляется циферблат электронных часов, который состоит из четырех позиций: две позиции для вывода часов и две позиции для вывода минут (чч:мм). Каждая цифра представляется набором пикселов из 9 строк по 6 пикселов в каждой строке. Каждый пиксел может принимать одно из двух значений символов «\*» или «.». Сочетания «\*» и «.» задают одну из 10 цифр. Образцы написания всех цифр имеются.

В идеале на циферблате часов показывается реальное время. Однако в системе часов произошел сбой и некоторые пикселы приняли «неправильные» значения.

Требуется определить, какое же время выводят часы или вывести сообщение о невозможности определения времени.

*Исходные данные.* В текстовом файле “etalon.txt” приведены образцы цифр в виде 10 групп 6 х 8 символов «\*» или «.». Во втором текстовом файле “times.txt” записаны показания с циферблата часов приведены образцы цифр в виде 4 групп 6 х 8 символов «\*» или «.».

*Выходные данные.* На экран выводится сообщение о времени в форме «чч:мм» или сообщение «Распознать не удается».

**Исходный код программы:**

public partial class Form1 : Form

{

List<char[,]> basedNumbers = new List<char[,]>();

List<char[,]> readedNumbers = new List<char[,]>();

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string baseFilePath = "";

using (OpenFileDialog openFileDialog = new OpenFileDialog())

{

openFileDialog.RestoreDirectory = true;

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

baseFilePath = openFileDialog.FileName;

textBox1.Text = baseFilePath;

using (StreamReader sr = new StreamReader(baseFilePath))

{

InitializeBasedNumbers(sr);

}

MessageBox.Show("OK");

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

string readingFilePath = "";

using (OpenFileDialog openFileDialog = new OpenFileDialog())

{

openFileDialog.RestoreDirectory = true;

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

readingFilePath = openFileDialog.FileName;

textBox2.Text = readingFilePath;

using (StreamReader sr = new StreamReader(readingFilePath))

{

ReadNumbers(sr);

}

string stringRepresentation = "";

for(int i = 0; i < 8; i++)

{

foreach(var j in readedNumbers)

{

for(int k = 0; k < 6; k++)

{

stringRepresentation += j[i, k];

}

}

stringRepresentation += "\n";

}

MessageBox.Show("OK");

}

}

}

private void InitializeBasedNumbers(StreamReader fileStream)

{

basedNumbers.Clear();

for(int i = 0; i <= 9; i++)

{

basedNumbers.Add(new char[8, 6]);

for (int j = 0; j < 8; j++)

{

string line = fileStream.ReadLine();

for(int k = 0; k < 6; k++)

{

basedNumbers[i][j, k] = line[k];

}

}

}

}

private void ReadNumbers(StreamReader fileStream)

{

readedNumbers.Clear();

for (int i = 0; i <= 3; i++)

{

readedNumbers.Add(new char[8, 6]);

for (int j = 0; j < 8; j++)

{

string line = fileStream.ReadLine();

for (int k = 0; k < 6; k++)

{

readedNumbers[i][j, k] = line[k];

}

}

}

}

private int CompareNumbers(char[,] comparing, char[,] based)

{

int weigth = 0;

for(int i = 0; i < 8; i++)

{

for(int j = 0; j < 6; j++)

{

if (comparing[i, j] == based[i, j])

weigth++;

}

}

return weigth;

}

private void button3\_Click(object sender, EventArgs e)

{

string resultStr = "";

foreach(var i in readedNumbers)

{

bool recognized = false;

int[] weights = new int[] { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

for(int j = 0; j <= 9; j++)

{

weights[j] = CompareNumbers(i, basedNumbers[j]);

}

int maxInd = 0;

for(int j = 1; j < 9; j++)

{

if (weights[j] > weights[maxInd])

maxInd = j;

}

if (weights[maxInd] > 24)

{

recognized = true;

resultStr += maxInd;

}

if (!recognized)

{

resultStr = "unrecognizable";

break;

}

}

if(resultStr == "unrecognizable")

{

MessageBox.Show("Дисплей повреждён, невозможно распознать");

return;

}

string hours = string.Format("{0}{1}", resultStr[0], resultStr[1]);

string minutes = string.Format("{0}{1}", resultStr[2], resultStr[3]);

if (int.Parse(hours) >= 0 && int.Parse(hours) <= 23 && int.Parse(minutes) >= 0 && int.Parse(minutes) <= 59)

{

MessageBox.Show(string.Format("Текущее время: {0}:{1}", hours, minutes));

}

else

{

MessageBox.Show("Часы показывают неверное время!");

}

}

}

**Скриншоты работы программы:**

Изображение выглядит как текст

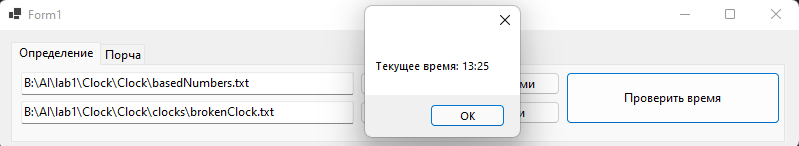
Автоматически созданное описание

*Основное окно программы*

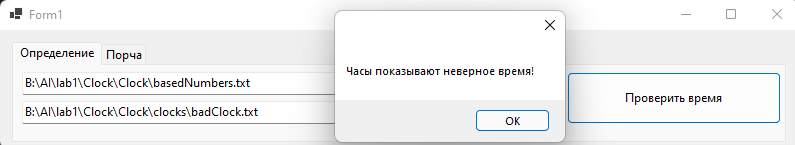
*Изображение выглядит как текст

Автоматически созданное описание*

*Результат распознавания часов с идеально прописанными цифрами*

**

*Результат распознавания часов с повреждёнными цифрами*

**

*Результат распознавания часов с некорректным временем*