C#实验报告1：控制台编程

3017218114 张启晗

一、实验要求

1. 熟悉并掌握C#控制台编程方式；
2. 掌握C#控制台程序的命令行参数读取方法
3. 理解QrCode的编码的基本原理；
4. 学会使用第三方已完成的NuGet安装包。

二、实验内容

a. 当用户在命令行输入参数时，判断参数是否是QrCode可以生成的信息，长度是否符合本程序的基本要求（自定义长度），如果符合要求就直接产生QrCode的结果，将QrCode的编码矩阵在控制台屏幕上输出对应黑、白字符方块组成的QrCode码。

b. 如果不符合要求，提示用户输入正确的命令行信息，运行结束。

c. 将程序的每一行都加上注释，并说明其功能。

d. 用手机识别是否可以还原数据，如果不能，请分析原因，修改程序，直至手机能正确识别QrCode内的原有信息。

e. 在命令行里传递一个文件名(包括文件所在目录，可以是可执行文件所在目录的相对目录例如data\qrcode.txt)，该文件是文本文件，文件中每行有一条字符串信息。假定myqrcode为本程序编译后的可执行文件名，-f表示QrCode信息放在-f后的data\qrcode.txt文件中。在控制台界面输入如下命令，可以将qrcode.txt中的每一行信息生成一个QrCode。

生成的QrCode需要保存到.png或bmp文件中，保存的文件名以信息所在行号三位数+信息的前四个字符构成，例如:”001+测试QR.png“为该文件的第一行信息生成的QrCode图片。

如果没有-f则以上述a的方式在控制台输出QrCode。此应用场景用于大量快速生成QrCode图片的应用场合。例如：支付宝卖支付码贴纸的场景。此外，如果QrCode信息是记录在Excel文件里或MySql数据库里，是否可以实现同样的功能。三种文件类型都能实现则更好。需要自己调研的内容：C#.NET的文本文件打开/关闭/读取；Office的Excel文件的打开/读取；数据库课程中的MySql数据库的表的建立/读取,.NET与MySql的连接方式。

三、详细设计

using Gma.QrCodeNet.Encoding;

using Gma.QrCodeNet.Encoding.Windows.Render;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Shiyan1

{

class Program

{

static void Main(string[] args)

{

switch (args.Length)

{

case 1: //唯一指定参数个数

if (args[0].Length >= 2 && args[0].Substring(0, 2).Equals("-f"))

{

//输入参数为文件名

if (args[0].Length == 2) goto default;

String[] lines;

//尝试读入文件所有行，若发生错误则报错

try

{

lines = File.ReadAllLines(args[0].Substring(2));

}

catch (Exception e)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("无法打开指定文件，请检查文件名是否正确");

Console.ResetColor();

goto default;

}

//分解路径中的各级目录

String[] pathElements = args[0].Substring(2).Split('\\');

String path = "";

//组合成文件所在目录

for (int i = 0; i < pathElements.Length - 1; ++i)

{

path += pathElements[i] + '\\';

}

//GraphicsRenderer类用于将QrCode对象输出为图片

GraphicsRenderer renderer = new GraphicsRenderer(new FixedCodeSize(256, QuietZoneModules.Four));

for (int i = 0; i < lines.Length; ++i)

{

Console.WriteLine("正在编码第" + (i + 1) + "行");

QrCode qrCode = EnCode('M', lines[i]);

if (qrCode == null) continue;

Console.WriteLine("编码成功，正在保存图片文件");

//按照规定的格式命名文件流

FileStream stream = new FileStream(path + i / 100 % 10 + i / 10 % 10 + i % 10 + lines[i].Substring(0, 4) + ".png", FileMode.Create);

//将二维码图片写入文件流

renderer.WriteToStream(qrCode.Matrix, System.Drawing.Imaging.ImageFormat.Png, stream);

Console.WriteLine("保存成功");

}

Console.WriteLine("完成");

}

else

{

//输入参数为待编码信息

QrCode qrCode = EnCode('M', args[0]);

if (qrCode == null) goto default;

else Console.WriteLine(CodeToString(qrCode));

}

break;

default: //出现问题，输出帮助信息

ShowHelpInfo();

return;

}

}

//输出帮助信息

static void ShowHelpInfo()

{

Console.WriteLine("参数格式说明：");

Console.WriteLine("Shiyan1.exe [message] ：在控制台输出message的二维码");

Console.WriteLine("Shiyan1.exe -f[path] ：为path指定的文件中的每行信息生成二维码并保存为png格式图片");

Console.WriteLine("可编码的最大长度就随随便便定为100好了");

}

//编码QRCode

static QrCode EnCode(char level, String content)

{

//判断长度是否符合要求

if (content.Length > 100)

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("待编码信息超过可编码的最大长度");

Console.ResetColor();

return null;

}

QrEncoder qrEncoder;

//根据传入的参数设置纠错等级，默认为M

switch (level)

{

case 'L':

qrEncoder = new QrEncoder(ErrorCorrectionLevel.L);

break;

case 'M':

qrEncoder = new QrEncoder(ErrorCorrectionLevel.M);

break;

case 'Q':

qrEncoder = new QrEncoder(ErrorCorrectionLevel.Q);

break;

case 'H':

qrEncoder = new QrEncoder(ErrorCorrectionLevel.H);

break;

default:

goto case 'M';

}

return qrEncoder.Encode(content);

}

//生成供控制台输出的字符串

static String CodeToString(QrCode qrCode)

{

if (qrCode == null)

{

return "";

}

String ret = "";

for (int i = -1; i <= qrCode.Matrix.Width; ++i) ret = ret + "█";

ret = ret + "\n";

for (int i = 0; i < qrCode.Matrix.Height; ++i)

{

ret = ret + "█";

for (int j = 0; j < qrCode.Matrix.Width; ++j)

ret = ret + (qrCode.Matrix[i, j] ? " " : "█");

ret = ret + "█\n";

}

for (int i = -1; i <= qrCode.Matrix.Width; ++i) ret = ret + "█";

return ret;

}

}

}

四、上机实验步骤

1) 新建一个控制台项目

2) 安装QrCode库

3) 程序编写实现“实验内容”中要求的内容。

五、实验结果

控制台运行结果

qrcode.txt的内容及data文件夹下生成的二维码

扫描控制台中二维码的结果