C# 第一次实验 实验报告

1. 实验要求：

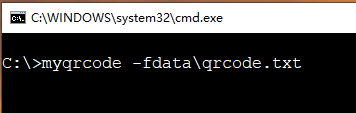
a. 当用户在命令行输入参数时，判断参数是否是QrCode可以生成的信息，长度是否符合本程序的基本要求（自定义长度），如果符合要求就直接产生QrCode的结果，将QrCode的编码矩阵在控制台屏幕上输出对应黑、白字符方块组成的QrCode码。

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。

2、实验内容：安装相关程序，实现。

3、详细设计：



using Gma.QrCodeNet.Encoding;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Cry

{

class Program

{

static void Main(string[] args)

{

QrEncoder IWannaCry = new QrEncoder();

QrCode Crying = new QrCode();

string intoEncode = String.Join(" ", args);

if (args.Length > 4)

{

Console.WriteLine("请输入小于3个参数");

}else if (args.Length<2)

{

Console.WriteLine("您没有输入任何参数");

}

else

{

IWannaCry.TryEncode(intoEncode, out Crying);

//Crying.Matrix[0,0];

for (int aTemp = 0, i = Crying.Matrix.Height; i > 0 || aTemp < Crying.Matrix.Height; i--)

{

for (int bTemp = 0, k = Crying.Matrix.Width; k > 0 || bTemp < Crying.Matrix.Width; k--)

{

if (Crying.Matrix[aTemp, bTemp] == true)

{

Console.Write("　");

}

else

{

Console.Write("■");

}

bTemp++;

}

Console.Write("\n");

aTemp++;

}

}

Console.Read();

}

}

}

4、上机实验步骤：

1、安装相关程序

2、阅读实验要求

3、官方网站阅读技术文档

4、搜索引擎上搜索相关教程

5、实现

6、调试

5、实验结果：



可以成功正常读取命令行输入，然后将之转化为二维码