C#程序设计语言与

.Net框架基础（双语）

**实**

**验**

**报**

**告**

**学 院 软件学院**

**年 级 2017级**

**班 级 一班**

**学 号 3017218060**

**姓 名 刘杭学**

**2019 年 4 月 4 日**



**C#程序设计语言与.Net框架基础（双语）**

**上机实验报告**

实验二 题目：Windows Form实现MIDI音乐文件的播放APP

**学院名称 软件学院**

**专 业 软件工程**

**学生姓名 刘杭学**

**学 号 3017218060**

**年 级 2017级**

**班 级 一班**

**时 间 2019年4月4日**

目 录

[实验名称 1](#_Toc509419319)

[实验目的 1](#_Toc509419320)

[实验内容 1](#_Toc509419321)

[实验要求 4](#_Toc509419322)

[实验分析 4](#_Toc509419323)

[实验结论及心得体会 5](#_Toc509419324)

## 实验名称 WPF实现MIDI乐队APP

## 实验目的

1. 理解和掌握基于WPF的APP应用程序开发
2. 理解Xaml在WPF界面设计中的作用

2) 学习使用WPF Midi Band提供的源程序实现WPF MIDI Band APP

3) 理解WPF Midi Band内部的类、event、delegate构成机制

## 实验内容

1) 使用WPF Midi Band提供的源程序，在Visual Studio中建立相应的解决方案。

2) 能够成功编译WPF Midi Band提供的演示程序。并能正常播放MIDI文件。

3) 理解演示程序的内部工作机制: 参照WPF Midi Band文章内容，理解以Event/Delegate方式实现的模块间的耦合机制，各种类的继承关系等。

4) 对GUI界面中的控件大小、位置进行完善，使之能够随APP界面大小自动调整其自身大小。需要使用相应的Event完成此项工作。

5) 其他GUI界面的用户体验提升：由同学们自己提出创新功能并实现。

## 实验要求

参实验内容完善详细的实验步骤，每位同学独立或两位同学结对编程完成实验。

## 实验分析

1. 实验结果

1.1项目名称： 实现WPF中控件的自适应。

1.2操作步骤：

增加代码

<Viewbox VerticalAlignment="Top" Stretch="Fill">

<Canvas Width="510" Height="700">

</Canvas>

</Viewbox>

1.3实际结果描述、结论：

实现WPF中控件的自适应

2.1项目名称： 实现WPF播放器

2.2操作步骤：

将实验二中WinForm实现的功能移植到实验三中的WPF中

2.3实际结果描述、结论：

成功实现了类似实验二中的功能

## 实验结论及心得体会

1) 理解和掌握基于WPF的APP应用程序开发

2) 理解Xaml在WPF界面设计中的作用

3) 学习使用WPF Midi Band提供的源程序实现WPF MIDI Band APP

4) 理解WPF Midi Band内部的类、event、delegate构成机制

5) 掌握了WPF中控件的使用方法和相关布局

6) 进一步理解了线程保护和跨线程调用

7) 了解到了dataBinding 的有关知识

相关代码

private void btnOpen\_Click(object sender, RoutedEventArgs e)

{

if (openMidiFileDialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

fileName = openMidiFileDialog.FileName;

lastFilename = fileName;

try

{

sequencer1.Stop();

playing = false;

sequence1.LoadAsync(fileName);

this.Cursor = System.Windows.Input.Cursors.Wait;

btnStart.IsEnabled = false;

btnContinue.IsEnabled = false;

btnStop.IsEnabled = false;

btnOpen.IsEnabled = false;

this.Dispatcher.Invoke(

System.Windows.Threading.DispatcherPriority.Normal,

new Action(

delegate()

{

Storyboard sbClockOpen = (Storyboard)FindResource("sbClockOpen");

grdClock.Visibility = System.Windows.Visibility.Visible;

sbClockOpen.Begin();

}

)

);

}

catch (Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message, "Error!", MessageBoxButtons.OK, MessageBoxIcon.Stop);

}

}

}

private void btnLoop\_Click(object sender, RoutedEventArgs e)

{

AutoLoop = !AutoLoop;

}