

7.2. MAUI

课程目标

- 了解MAUI的特点
- 了解MAUI的开发环境和工具
- 了解MAUI的基本控件和布局
- 能够使用MAUI创建一个简单的图形化界面

课程内容

介绍MAUI

- MAUI是什么？

MAUI是.NET Multi-platform App UI的缩写，是一种新兴的**跨平台**应用程序开发框架，用**C#和XAML**开发可以在多个平台上运行的应用程序，不仅支持移动平台（Android和iOS），还支持桌面平台（macOS和Windows）。

- MAUI有什么优势？

1. MAUI可以让你用一个单一的代码库来开发可以运行在不同平台的应用，无需为每个平台编写不同的代码。
2. MAUI可以让你共享UI布局和设计，同时也可以根据需要添加平台特定的代码和资源。

搭建MAUI开发环境

- 安装Visual Studio 2022和.NET 6 SDK（课前完成）

安装Visual Studio 2022时，需要选择.NET Multi-platform App UI开发、.NET 桌面开发工作负荷。

- 安装MAUI工作负载（课前完成）

1. 安装/检查/修复MAUI工作负载需要使用maui-check命令行工具，它可以验证你的开发环境并安装任何缺失的组件。
2. 要安装maui-check命令行工具，需要运行以下.NET命令行接口（.NET CLI）命令：`dotnet tool install -g redth.net.maui.check`。
3. 要运行maui-check命令行工具，需要在命令提示符中输入：`maui-check`。这个命令会检查你的开发环境中是否有MAUI所需的组件，如果有缺失或不匹配的组件，它会提示你是否要修复或更新它们。
4. maui-check命令行工具还支持一些参数，可以用来指定不同的选项，如使用预览版的清单文件，跳过某些检查项目，自动修复问题等。要查看所有可用的参数，可以输入：`maui-check --help`。

- 配置模拟器或真机调试（课前可选）

1. 模拟器是一种虚拟设备，它可以在你的开发机上模拟不同平台的设备，如Android手机或iOS手机。配置方式请参考：<https://learn.microsoft.com/zh-cn/dotnet/maui/android/emulator/device-manager>
2. 真机是一种实体设备，它可以通过USB或无线连接到你的开发机上，如Android手机或iOS手机。配置方式请参考：<https://learn.microsoft.com/zh-cn/dotnet/maui/android/device/setup>

- 创建一个MAUI项目并运行

1. App是MAUI应用的入口点，它们定义了一个App类，继承自Application类；App.xaml是用XAML语言声明App类的属性和资源，如主题颜色，字体大小等；App.xaml.cs是用C#语言编写App类的逻辑，如初始化应用，设置主窗口，处理生命周期事件等。
2. AppShell是MAUI应用的导航框架，它们定义了一个AppShell类，继承自Shell类；AppShell.xaml是用XAML语言声明AppShell类的内容和结构，如导航菜单，标签页，飞出页等；AppShell.xaml.cs是用C#语言编写AppShell类的逻辑，如注册路由，处理导航事件等。
3. MainPage是MAUI应用的主页面，它们定义了一个MainPage类，继承自ContentPage类；MainPage.xaml是用XAML语言声明MainPage类的用户界面，如控件，布局，样式等；MainPage.xaml.cs是用C#语言编写MainPage类的逻辑，如事件处理，数据绑定，页面生命周期等。
4. MauiProgram.cs是应用的跨平台入口点，它用来配置和启动应用。模板启动代码指向了App.xaml文件中定义的App类。

使用MAUI控件和布局

- 熟悉MAUI的控件库

MAUI的控件库包含了三种主要的控件类型：页面，布局，和视图。

1. 页面（Page）是MAUI应用的基本单元，它通常占据整个屏幕或窗口，每个页面至少包含一个布局。
2. 布局（Layout）是用来组织和排列用户界面控件的容器，每个布局通常包含多个视图或其他布局。
3. 视图（View）是用户界面的基本元素，它可以显示数据或接收用户输入，如标签（Label），按钮（Button），文本框（Entry），图片（Image）等。

- 学习如何使用常用的控件，如Label, Button, Entry, Image等

1. Label是用来显示文本的视图，它可以设置文本的内容，字体，颜色，对齐方式等属性。
2. Button是用来响应用户点击的视图，它可以设置按钮的文本，图标，背景色等属性，并通过Clicked事件来执行相应的逻辑。
3. Entry是用来接收用户输入的单行文本框视图，它可以设置文本框的占位符，键盘类型，密码模式等属性，并通过TextChanged事件来监听文本变化。
4. Image是用来显示图片的视图，它可以设置图片的来源（Source），缩放模式（Aspect），填充模式（Fill）等属性。

- 学习如何使用布局管理控件，如StackLayout, Grid, FlexLayout等

1. StackLayout是用来将子视图按照垂直或水平方向堆叠起来的布局，它可以设置方向 (Orientation)，间距 (Spacing)，填充 (Padding) 等属性。
2. Grid是用来将子视图按照行列网格排列起来的布局，它可以设置行定义 (RowDefinitions)，列定义 (ColumnDefinitions)，行间距 (RowSpacing)，列间距 (ColumnSpacing) 等属性，并通过Grid.Row和Grid.Column来指定子视图的位置。
3. FlexLayout是用来将子视图按照灵活的方式堆叠或换行的布局，它可以设置方向 (Direction)，对齐 (AlignItems)，换行 (Wrap)，间距 (Spacing) 等属性，并通过FlexLayout.Basis和FlexLayout.Grow来指定子视图的大小。

- 学习如何使用XAML和C#代码编写界面逻辑

示例：为初始代码创建一个登陆界面。

创建一个简单的图形化界面

- 设计一个简单的计算器界面，包含数字键盘，运算符键盘，显示屏等控件
- 使用Grid布局控件对界面进行分区和对齐
- 使用C#代码编写计算器的功能逻辑

课后作业（二选一）

1. 完善计算器calculator

- 要求：
 1. 添加DEL按钮，功能：删除上一个输入的数字字符或运算符；当上一个字符为"="时，按下DEL会清空显示屏但**不改变所存储的计算结果**（即lastNumber的值）
 2. 考虑程序的鲁棒性，在原始代码中，当使用者重复按下运算符时，会将lastNumber作为第二个操作数进行计算，也即按下"AC"、"1"、"+"、"任意运算符"后会显示2，按下"AC"、"/"、"任意运算符"会显示NaN，这在一定程度上是不合适的。希望进行如下修改（只考虑"+"、"-"、"*"、"/"）：当重复输入运算符时，不进行运算，根据**最后输入**的运算符进行更新，直至用户输入数字字符。
 3. 添加第二页面（复杂计算模式），选中该页面后，会添加乘方 (x^y)、对数 ($\lg x, \ln x$)、开根号 (\sqrt{x})、阶乘 ($x!$)、三角函数 (\sin, \cos, \tan, \dots) 等运算符和 π 、 e 等数学常数；用户可以随时切换这两个页面并保持**各参数不变**（如displayLabel.Text、lastNumber等属性）
 4. 如果能美化一下外观就更好啦o(=• ㄟ •=)m
- 作业提交方式：通过Github提交，fork 仓库：<https://github.com/Shawgeem/EESAST-hw-2023-MAUI-calculator> 到个人仓库，按要求修改好后，从个人仓库提pr到原仓库，pr信息填写为：`calculator_姓名_班级`（如：`calculator_神様_无20`）。

2. 使用MAUI创建一个简单的图形化界面来实现一个日程管理程序scheduler

- 要求：
 1. 每一项“日程”应具有日程名、紧迫程度、DDL、备注等属性
 2. 程序应具有新建日程、显示日程信息、修改日程、删除日程等功能
 3. 显示日程列表时，采用按时间排序、按紧迫程度排序等不同方式排列

- 可以参考\clgg/: <https://github.com/TCL606/MAUI-Scheduler>
- 作业提交方式：通过Github提交，fork 仓库：<https://github.com/Shawqeem/EESAST-hw-2023-MAUI-scheduler> 到个人仓库，按要求修改好后，从个人仓库提pr到原仓库，pr信息填写为：`scheduler_姓名_班级`（如：`scheduler_神様_无20`）。