**实验3.5 委托（二）**

如果你出去旅行，计划在旅途中做很多事（吃饭、游玩景点、体验挑战、探索自然等）并且所有步骤已安排妥当，但是具体乘坐哪种交通工具不清楚，交通工具的选择决定出行时间，出行时间决定旅行期间能够安排多少项目，由于出行的方式有很多种，需要为每一种出行制定一个旅行方案，例如 TravelPlanbyCar()、TravelPlanbyBike()、TravelPlanbyMoto()……。为了避免这种情况，于是我们设计如下程序。

**步骤一：新建一个windows 窗体工程（不要用WPF模板）、设计窗体（要求用自己的名字命名窗体）**



**步骤二：在public partial class Form1 : Form 这个类中定义一个委托：**

*public Form1()*

*{*

*InitializeComponent();*

*}*

public delegate double TravelPlan(double s); //定义名为TravelPlan的委托

**步骤三：将委托对象作为Form1的Active方法的参数**

private void Active(TravelPlan travelPlan)

{

double distance = 200;//旅行距离

double time=travelPlan(distance);

MessageBox.Show("旅途需要的时间是："+ time+"小时");

// 后续安排：享受之旅(吃饭、游玩景点、体验挑战、探索自然等)

Eat();

//…

// 结束

}

public void Eat()

{

MessageBox.Show("先吃顿大餐!");

}

**步骤四：实现button1的click事件处理函数：**

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

{

Form1 frm = new Form1();

String goMethod = comboBox1.SelectedItem.ToString();

if (goMethod == "开汽车")

{

TravelPlan myTravelPlan = new TravelPlan(DriveCar);

frm.Active(myTravelPlan);

}

else if (goMethod == "骑摩托")

{

TravelPlan myTravelPlan = new TravelPlan(DriveMoto);

frm.Active(myTravelPlan);

}

else if (goMethod == "骑自行车")

{

TravelPlan myTravelPlan = new TravelPlan(RideBike);

frm.Active(myTravelPlan);

}

}

**步骤五：实现各种出行方式的具体业务**

private double DriveCar(double distance)

{

// 开汽车

double speed = 100;

double time = distance / speed;

return time;

}

private double DriveMoto(double distance)

{

// 骑摩托

double speed = 60;

double time = distance / speed;

return time;

}

private double RideBike(double distance)

{

// 骑自行车

double speed = 30;

double time = distance / speed;

return time;

}

思考：从这个例子可以看出使用委托有哪些好处？

1. 方便在不同窗体之间传递数据（实验3.4）。
2. 能够以委托为中间桥梁，将方法作为参数传递给另一个方法，从而减少方法的重载次数。

**实验截图（截第一弹窗即可）：**

namespace sy3\_5

{

public partial class Form1 : Form

{

// 定义名为TravelPlan的委托

public delegate double TravelPlan(double s);

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void Active(TravelPlan travelPlan)

{

double distance = 200;

double time = travelPlan(distance);

MessageBox.Show("旅途需要的时间是：" + time + "小时");

Eat();

}

public void Eat()

{

MessageBox.Show("先吃顿大餐");

}

private double DriveCar(double distance)

{

// 开汽车

double speed = 100;

double time = distance / speed;

return time;

}

private double DriveMoto(double distance)

{

// 骑摩托

double speed = 60;

double time = distance / speed;

return time;

}

private double RideBike(double distance)

{

// 骑自行车

double speed = 30;

double time = distance / speed;

return time;

}

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

{

Form1 form1 = new Form1();

string? goMethod = comboBox1.SelectedItem.ToString();

if (goMethod=="开汽车")

{

TravelPlan travelPlan = new TravelPlan(DriveCar);

form1.Active(travelPlan);

}

else if (goMethod == "骑摩托")

{

TravelPlan travelPlan = new TravelPlan(DriveMoto);

form1.Active(travelPlan);

}

else if (goMethod == "骑自行车")

{

TravelPlan travelPlan = new TravelPlan(RideBike);

form1.Active(travelPlan);

}

}

}

}





