

A.4.2 属性和方法（WPF）

参考如图 A-8 所示的运行界面，编写 WPF 应用程序，实现以下功能。

（1）声明一个名为 `CourseBeginTime` 的枚举类型，枚举值有：秋季、春季。

（2）定义一个 `CourseInfo` 类，该类包含 4 个属性：`CourseName`（课程名）、`CourseTime`（开设学期）、`BookName`（书名）、`Price`（定价）4 个属性，其中 `CourseTime` 为 `CourseBeginTime` 类型。

（3）在 `CourseInfo` 类中包含一个静态变量 `Counter`，每创建一个 `Course` 实例，该变量值都会自动加 1。

（4）分别为 `CourseInfo` 类提供无参数的构造函数和带参数的构造函数，在构造函数中分别设置 4 个属性的值。

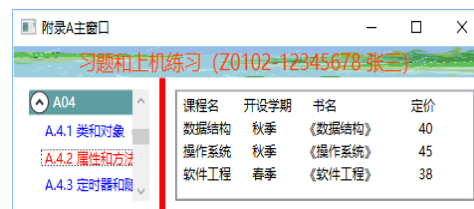


图 A-8 A.4.2 的运行效果

（5）在 `CourseInfo` 类中提供一个 `Print` 方法，显示该实例的 4 个属性值。

（6）在 WPF 页的 `Loaded` 事件中分别创建不带参数的 `CourseInfo` 实例和带参数的 `CourseInfo` 实例，测试类中提供的功能，并将结果在 `ListBox` 中显示出来。

A.4.3 定时器和随机数（WPF）

编写 WPF 应用程序实现以下功能：定义一个 `RandomHelp` 类，该类提供一个静态的 `GetIntRandomNumber` 方法、一个静态的 `GetDoubleRandomNumber` 方法。

在页面中让用户指定随机数范围，当用户单击【开始】按钮时，启动定时器，在定时器事件中调用 `RandomHelp` 类中的静态方法生成随机数，并在页面中显示出来。当用户单击【停止】按钮时，停止定时器，然后用比原字体大一倍的字体显示最后生成的随机数。

程序运行效果如图 A-9 所示。



图 A-9 定时器和随机数

A.5.1 类继承——构造函数（WPF）

编写 WPF 应用程序完成下列功能。

- （1）创建一个类 A，构造函数内输出“A”，定义一个名为 Result 的 string 类型的属性。
- （2）创建一个类 B，让其继承自 A，并在 B 的构造函数中将 Result 属性赋值为“B”。
- （3）创建一个类 C，让其继承自 B，并在 C 的构造函数中将 Result 属性赋值为“C”。
- （4）声明一个类型为 B 的变量 b，并将 b 初始化为类 C 的实例。

A.5.2 类继承——虚拟和重写（WPF）

编写 WPF 应用程序完成下列功能。

- （1）创建一个类 D，然后在 D 中声明一个扩充类可写入值的名为 Result 的 string 类型的属性，并编写一个可以被重写的带 int 型参数的方法 MyMethod，在该方法中将传递给该方法的整型值加 10 后的结果添加到 Result 属性中。
- （2）创建一个类 E，使其继承自类 D，然后在该类中重写 D 中的 MyMethod 方法，将 D 中接收的整型值加 50，并将结果添加到 Result 属性中。
- （3）分别创建类 D 和类 E 的对象，分别调用其 MyMethod 方法。