**学习大纲**：

* 了解最佳实践的5条一般规则
* 如何使用Invoke Workflow活动
* 如果使你的项目更加简洁高效

**1. 最佳实践**

1) 在开始一个项目时，为每一个流程、子流程选择合适的布局（FlowChart和Sequence）。

* 主流程用Flowchart/State Machine。一是能更清晰的了解整个项目；二是便于测试主流程中的某个单独部分。
* 更高层面的决策或业务逻辑用Flowchart。
* UI交互用Sequence。一般交互都是一个活动接着一个活动的执行，没有很多变化。
* 尽量不要使用嵌套的If活动，用Flowcharts代替会更容易理解。

2) 将整个流程分解为几个更小的流程。开发者会通过将非常复杂的应用程序和算法分解成更小的可控的任务来实现它们。

* 每一小块都可以独立开发和测试。
* 一些流程可以在其他项目中重复利用，以减少开发时间。
* 在同一个项目中与多人合作开发。

3) 做好异常处理。异常处理用来捕获和处理错误，至少要记录这些错误用于之后分析。

* 把产生问题的流程放到Try Catch活动里。
* 把调用的外部流程放到Try Catch活动里。
* 将恢复操作放到Catch部分。

4) 使流程清晰可读。可读性好的流程更便于调试，而且在团队项目里，别人可能需要使用你的流程。

* 使用描述性的名称命名流程组件。
* 做好注释和说明。
* 记录实时的执行进度，这样做有助于更快地查明错误。
* 将Workflow的设置保存到单独的文件中，推荐使用JSON文件。

5) 保持运行环境的清洁。让UiPath及时关掉不用的应用程序、窗口、网页。

**应用最佳实践原则的好处：**

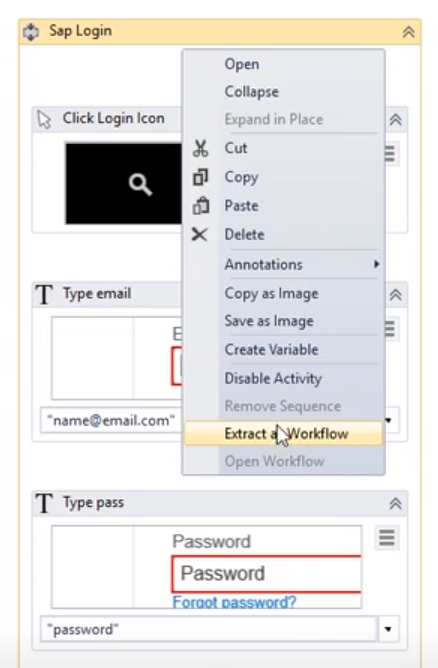
* Reliable 可靠：能处理各种外部和内部的错误异常。
* Efficient 高效：减少开发时间；使程序在生产环境中流畅运行。
* Maintainable 维护：易于修改、合并、交接。
* Extensible 扩展：易于添加新情况。

**2. Invoke Workflow File活动**

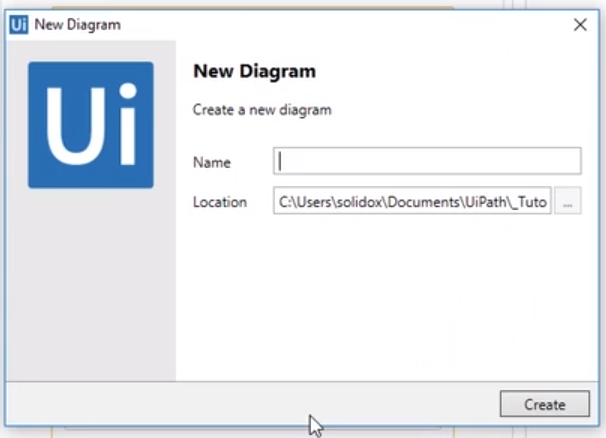
使用**Invoke Workflow File活动**可以将一个项目分解成几个小的组件，并且能将独立的Workflow连接到一起。

1). 将Workflow中的某个Sequence保存为新的Workflow并调用它。

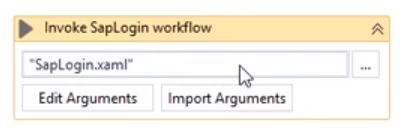
* 在Sap Login的Sequence上右键，选择Extract as Workflow。



* 在弹出的窗口中命名新的Workflow为Sap Login。



* 在原Workflow里，**Invoke Workflow File活动**自动取代原先的Sequence。



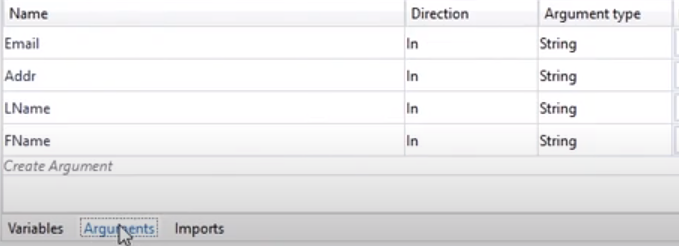
这样做的好处是，你可以随意拖拽、复制Invoke Workflow File活动到Workflow的任何位置，也可以在其他项目里重复利用它。它便于维护和更新，也便于单独进行测试。

2) 传递参数

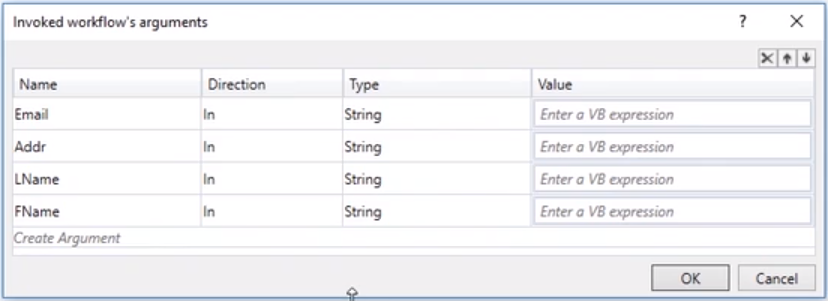
* 在Enter User Data的Sequence里，timeout变量是本地变量，其余四个变量是全局变量。



* 右键Enter User Data，选择Extract as Workflow。在提取出的Workflow中，本地变量timeout依然在Variables面板里保持不变，而其它四个全局变量出现在Arguments面板里。



* 在原本的Workflow中，会以一个Invoke Workflow File活动取代原先的Enter User Data Sequence。点击Import Arguments，可以看到新Workflow的四个参数。



* 在Value列指定参数的值。可以直接填入固定值，也可以填入一个变量。
* 在Direction列指定参数的传递方向，In表示把值从Invoke活动所在的Workflow传入被引用的Workflow（在本例中为从Flowchart传给Enter User Data）；Out表示把值从被引用的Workflow传给Invoke活动所在的Workflow（从Enter User Data传给Flowchart）；In/Out则表示双向传递，先传入再传出。

3) 多个Workflow组合

一般来说在执行Enter User Data（录入用户数据），首先需要Get User Data（获取用户数据）。

* 在主程序Flowchart里新建一个Sequence名为Get User Data，添加一些UI交互活动实现从网站取得用户信息。取得的信息存放在变量里。
* 把这个Sequence提取为Workflow，把原先用来存储用户信息的变量设置为Out的参数。Out的参数是把Get User Data子Workflow的值，传到主程序Flowchart里。Invoke Workflow File活动的Value一栏用来填写保存传出来的值的地方，一般是本地变量。
* 最后把Flowchart里保存用户信息的变量填入到Enter User Data子Workflow对应的In参数的Value一栏里，把信息传递过去，并填入系统中。

**3. 项目组织**

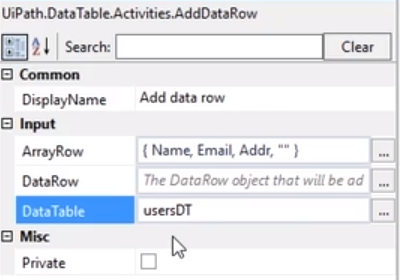
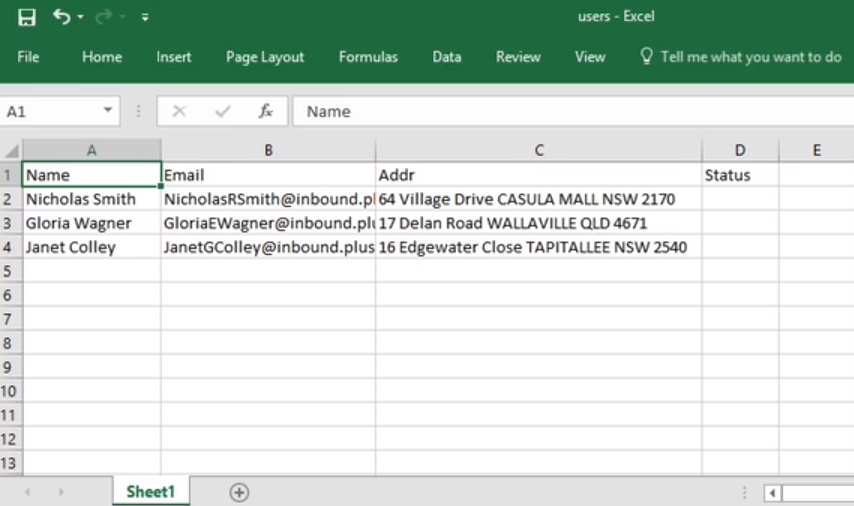
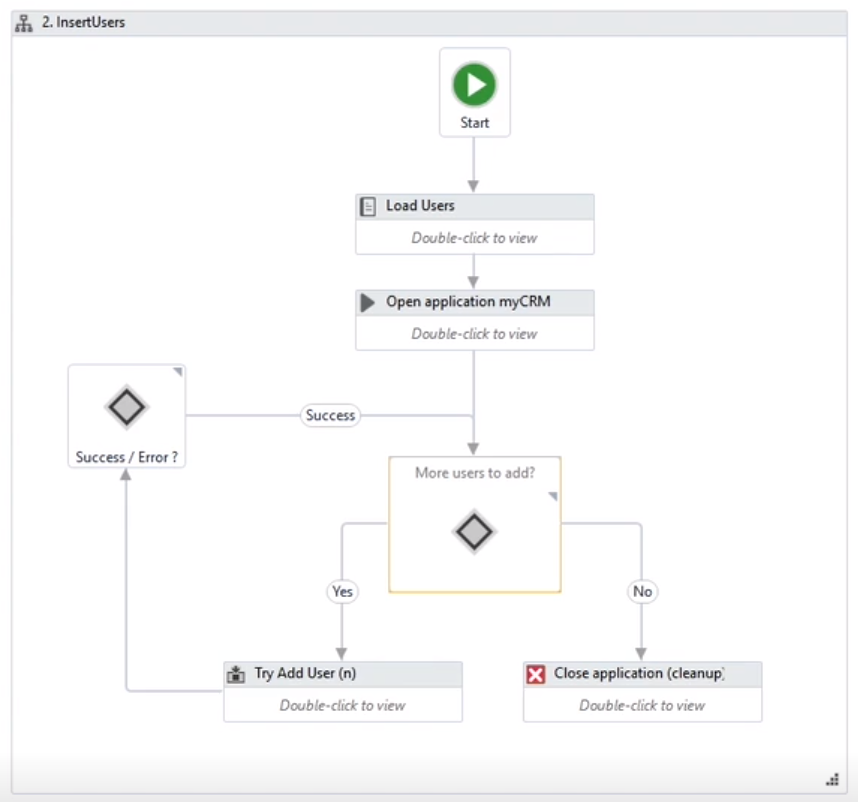
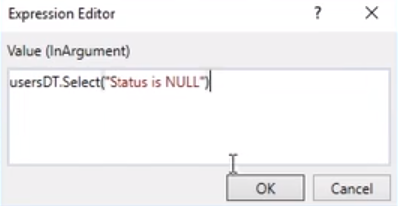
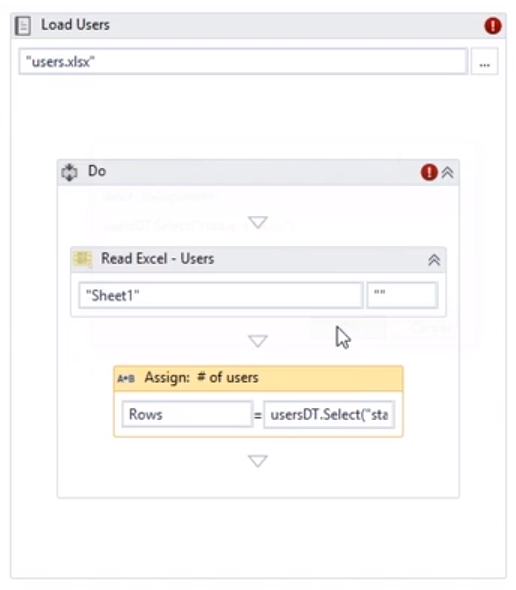
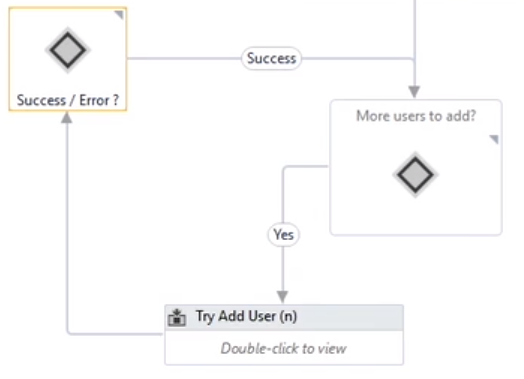
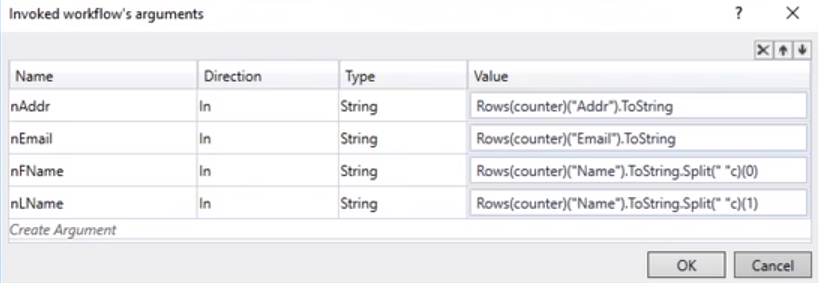
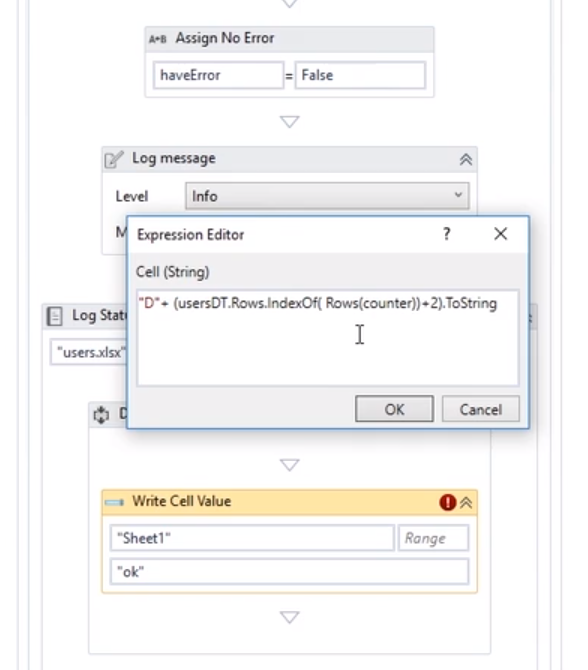
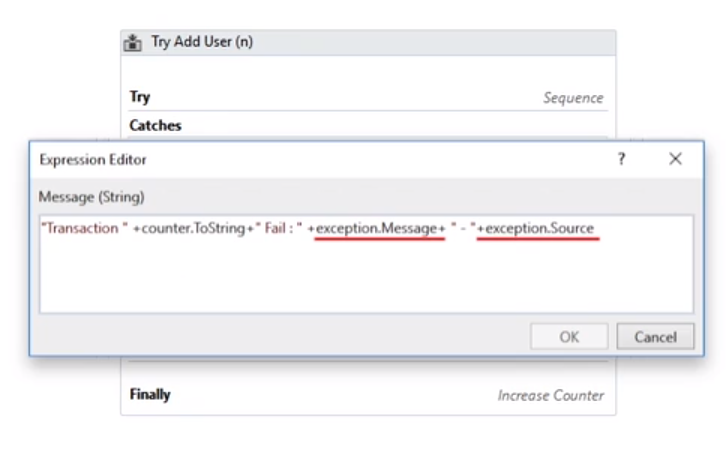
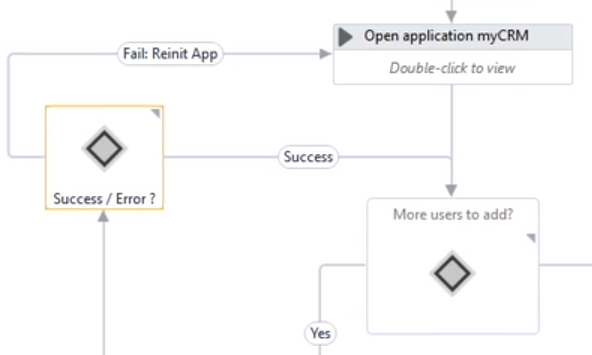
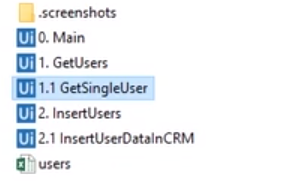
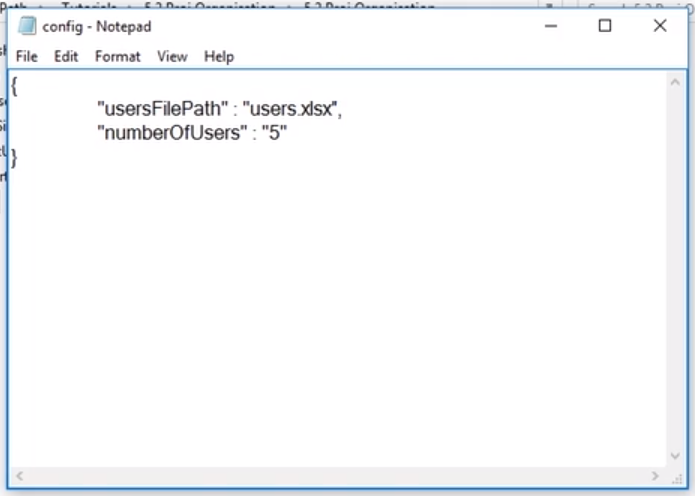
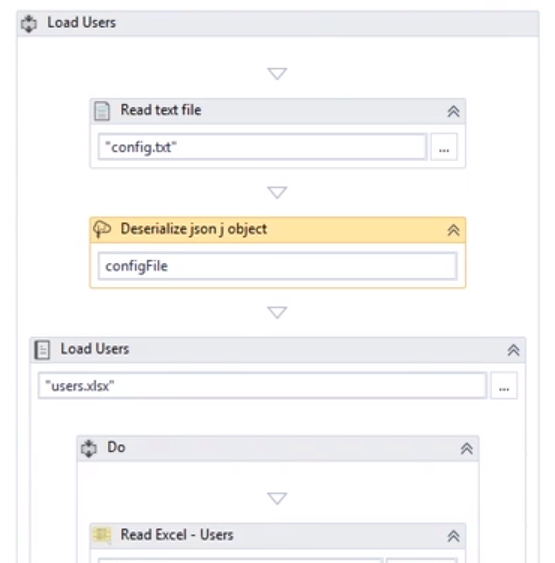
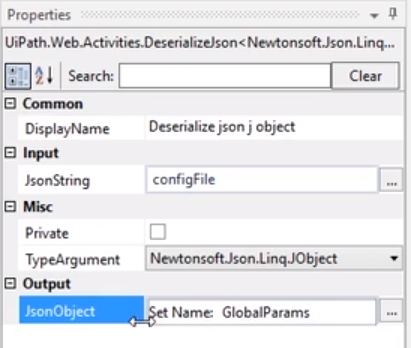
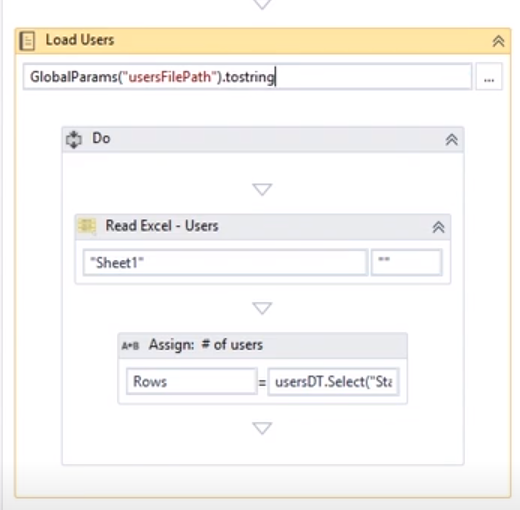
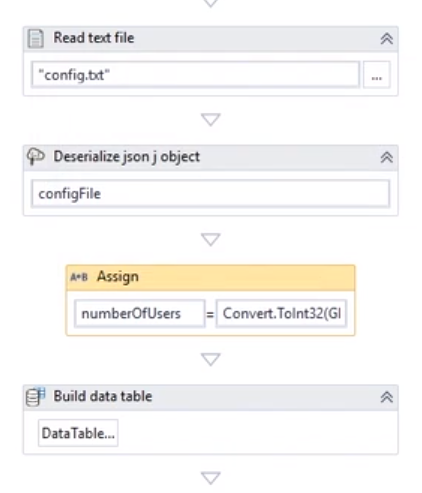
**实例**：获取x个用户信息，并填入myCRM系统里。

**思路**：

* 获取x个用户信息
  + 打开生成用户信息的网站
  + 生成并存储x个用户信息
  + 关闭浏览器
  + （上述活动放入一个Try-Catch里以保证程序的稳定性。）
* 把x个用户信息填入myCRM系统
  + 打开CRM应用程序
  + 录入x个用户信息
    - 循环x次（在循环内部添加Try-Catch，保证填写某个用户信息出错时，不影响剩余的用户信息的填写。）
      * Try-填入用户信息
      * Catch-登记错误信息
  + 关闭应用程序

**过程**：

本课需要我们整体的看某个项目，因此对于具体操作、活动不做详细的说明。

* **主程序：**
  + 创建一个主Workflow和两个子Workflow，可以按照从上到下的、从里到外的顺序编号命名，提高可读性。比如：0. Main，1. Get User，2. Insert User。
* **子程序-获取用户信息：**
  + 子程序1. Get User由一个Try-Catch活动构成。
  + 在Try的部分，添加一个Open Browser活动打开生成用户信息的网站。
  + 在Open Browser的Do里添加一个Do While循环。设置一个counter变量作为循环变量，初始值为0。假设我们要取得10个用户的信息，那么Do While活动的条件就是counter<10。10可以用参数numberOfUsers传递，方便随时修改。Do While的主体里需要添加一个Assign活动，使counter=counter+1，否则会进入死循环。
  + 在Do While里添加一个Invoke Workflow File活动，引入我们之前已经做好的获取用户信息的子程序，配好参数（Out，实在搞不清直接填In/Out），该子程序可以命名为1.1 getUserData。
  + 保存多个用户的信息最好使用datatable。在Do While活动之前添加一个Build Data Table活动，创建一个空的datatable变量usersDT，它包括Name列，Email列，Addr列和Status列（Status列用来记录用户信息是否正确输入CRM）。在Do While里的Invoke Workflow File活动之后，添加一个Add Data Row活动，将从1.1 getUserData里取得的name，email，address写入datatable。
  + 
  + 
  + 在Do While活动之后添加一个Excel Application Scope活动，把usersDT存到excel文件users.xlsx里。添加一个Close Application活动关闭浏览器。
  + 添加一个Output Data Table活动和Log Message活动用来验证结果是否正确。
* **子程序-录入用户信息：**
  + 子程序2. InsertUsers由一个Workflow构成。如下图：
  + 
  + Load User 是一个Excel Application Scope活动，在其中添加Read Range活动读取子程序1中存储用户数据的Excel文件users.xlsx。
  + 在Read Range活动之后添加一个Assign活动，新建一个Datarow[]（数组）类型的变量Rows，给它赋值usersDT.Select("Status is NULL")，表示选取Datatable里Status值为空或NULL的行。
  + 
    - Select可以用来对Datatable的数据进行筛选，筛选结果是一个Datarow的数组。
    - 在Google搜索datatable select可以查询Select的语法。
  + Load Users的部分现在如下图所示：
  + 
  + 在Flowchart的Load User后面添加Open Application活动，用来打开myCRM应用程序。
  + 接下来，我们要做一个Flowchart的循环，来逐一添加用户信息，而且可以使用我们之前已经写好的EnterUser Workflow。这个循环基本如下：
    - 判断Datatable里是否还有下一个用户数据需要录入。
    - 如果有则录入数据。没有则执行Close Application活动。
    - 如果数据录入成功，则继续判断Datatable是否还有下一行数据。
    - 
  + 首先是一个Decision，判断是否有下一行数据的。这需要知道当前执行到哪一行了，因此要创建一个计数用的变量counter，设置其初始值为0。
    - 判断条件为counter < Rows.GetLength(0)。Rows.GetLength(0)计算Rows这个数组变量中一共有几个元素（DataRow）。
    - 私以为在循环里应该有Assign活动为变量counter+1，视频里没有讲到，此处请自行实验。
  + Try Add User实际上是一个Try Catch活动。Try的部分里：
    - 添加一个Log Message活动记录和显示实时运行情况。
    - 添加Invoke Workflow File活动把之前做好的录入信息的workflow引入进来，并设置参数。
    - 添加Assign活动，新建Boolean型变量haveError，赋值为False。因为如果程序运行到该位置还没Catch到错误，说明用户信息已经录入成功了。
    - 再添加一个Log Message活动记录实时运行情况。
    - 添加Excel Application Scope活动+Write Range活动，在users.xlsx的Status列写入ok。
  + 在Try的部分，我们可能遇到的问题是：
    - 配置Invoke Workflow File的参数。如图所示：
      * 
      * Rows(counter)表示索引号为counter的行，即当前录入的行。其后加上("Addr")就可以表示该行对应的Addr列的数据，最后加上.ToString将其转换为字符串型。
      * .Split(" "c)表示已空格为分隔符对前面的字符串进行拆分，(0)表示用Split方法拆分后得到的字符串数组里的第一个元素，在此例中即人名，(1)则表示姓。这个方法在此例中可以正常运行，因为全名里没有包括中间名。
    - Write Range活动的单元格地址。我们已知Status在D列，但是行号是变化的。写法是"D"+(usersDT.Rows.IndexOf(Rows(counter))+2).ToString。
      * Rows(counter)表示的是Rows变量里的第counter个元素（DataRow），也就是当前行。
      * usersDT.Rows.IndexOf(DataRow)表示的是DataRow在usersDT的所有行里的索引号。
      * 最后的结果+2。Excel的行号从1开始，而Datatable是从0开始，因此要+1。考虑到Excel里的第一行是标题行，因此需要再+1.
      * 
  + 对于Try Add User的Catch的部分：
    - 首先添加一个Log Message活动，写入exception.Message和exception.Source，以获得错误信息方便后期调试。
    - 
    - 添加一个Assign活动，把haveError赋值为True。
    - 添加一个和Try部分里的一样的Excel Application Scope活动，只是写入的内容为fail，表示录入用户信息失败。
  + 最后，对录入信息是否成功的Decision进行设置，判断条件为变量haveError。如果haveError的值为True则接着执行下一行；如果为False，此处使用的处理方法需要手动关闭myCRM，程序会自动打开myCRM。
  + 
* **修改程序项目-增加可读性和灵活性**
  + 可读性：修改每一个程序的名称，使结构一目了然。
  + 
  + 灵活性：**将一些后期容易发生变化的设置写入配置文件里**，可以是excel文件、csv文件、或者JSON文件。此处使用JSON文件，它的易用性比较好。
    - 私以为增加可读性和灵活性的工作应该在开始制作这个项目的时候就同时进行，否则后期对活动的参数的修改很容易有遗漏或错误。
  + 如何进行配置，可以参考下面的方法：
    - 在项目文件夹里新建一个txt文件命名为config。
    - 在config文件里按照JSON的书写方式写入配置的参数，如下图。此处写入了两个参数，一个是Excel文件的路径，一个是每次运行时提取多少个用户的信息（回忆一下，在1.GetUser子程序里，Do While循环的条件是counter<你想要提取用户信息的数量）。
    - 
    - 保存并关闭config文件。
    - 在UiPath里，以设置Excel文件路径为例，首先在Load Users添加一个Read Text File活动。读取config.txt的内容到变量configFile中。
    - 添加Deserialize JSON Object活动，并配置jsonString参数为configFile，Output为一个新变量GlobalParams。
    - 
    - 
    - 接着就需要把所有的Excel文件的地址修改为GlobalParams("usersFilePath").ToString。
    - 
    - 对所以的主程序和子程序进行修改：
    - 

**4. 练习（略）**

**\*本课使用过的新活动、方法、函数等：**

**Deserialize JSON Object**

**Convert.ToInt32()**

ps. OTZ 我已经记不清之前有什么了...