

# LECTURE01B. INTRODUCTION TO UIPATH STUDIO

---

**Course Presentation**

**[1 October 2019]**

Elective Course, 2019-2020, Fall Semester

Camelia Chisăliță-Crețu, Lecturer PhD

Babeș-Bolyai University

# Acknowledgements

This course is presented to our Faculty with the support of UiPath Romania.



# Contents

- UiPath Studio Introduction
  - Automation Project
    - Definition. Types. Structure
  - User Interface
    - Ribbon. Panels
  - Variables
    - Data Types: integer, String, Boolean, Generic, Array of T
    - Variables Panel. Refactoring names. Scope. `""` for Strings
  - Choices
    - If Activity, Flow Decision Activity, If Operator, Switch Activity, Flow Switch Activity
  - Demo1
  - Control Flow Activities
    - For Each, While, Do While
  - Demo2

# Automation Project. Definition

- An **activity** is
  - the smallest action in UiPath;
  - a **step** in a process workflow;
- An **automation project** is
  - A **set of steps** that allows to perform a meaningful task;
  - a graphical representation of the business **process**;
- it allows to automate a rule-based process, formed by custom set of steps;
- E.g.:
  - Click on a button;
  - Read a file;
  - Write to a log file.

# Automation Project. Types

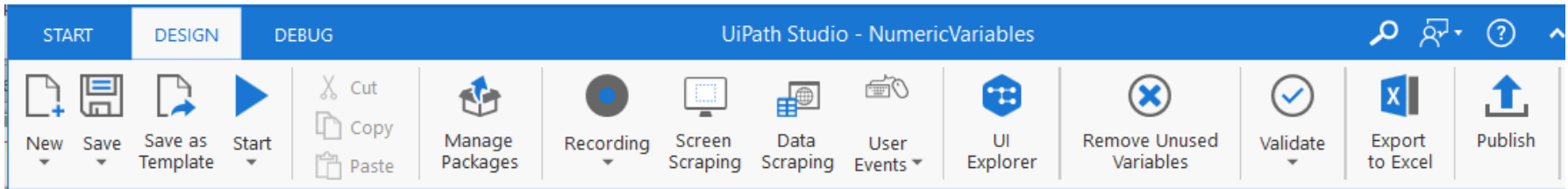
- Types of supported projects
  - **Sequences** - for linear processes;
    - it connects one activity to another without cluttering the project;
    - *when to use*: simple scenarios, activities follows one after another;
    - easy to assemble and understand;
  - **Flowcharts** - for more complex processes;
    - it integrates decisions and connects activities in a more diverse manner through multiple branching and logic operators;
    - it provides a two dimensional view of the workflow;
    - *when to use*: to show decision points in a process, no form constraints, visual appealing;
    - *cons*: prone to chaotic interweaving of activities;
  - **State machine** - for very large projects;
    - it applies to projects that use a finite number of states during execution which are triggered by a condition or an activity;
    - *when to use*: to represent standard high-level process diagram of transactional business process templates.

# Automation Project. Structure

- by default:
  - **Main.xaml file** – it consists of the main flow;
    - a sequence or a flowchart can be initially added;
    - other .xaml files may be added;
    - at run time this file will be executed only ==> all other .xaml files are connected in **Main.xaml** through the **Invoke Workflow File** activity;
  - **.screenshots folder** – it is generated if the project uses UI automation;
    - to save the screenshot;
  - **project.json** – it contains details on the automation project;

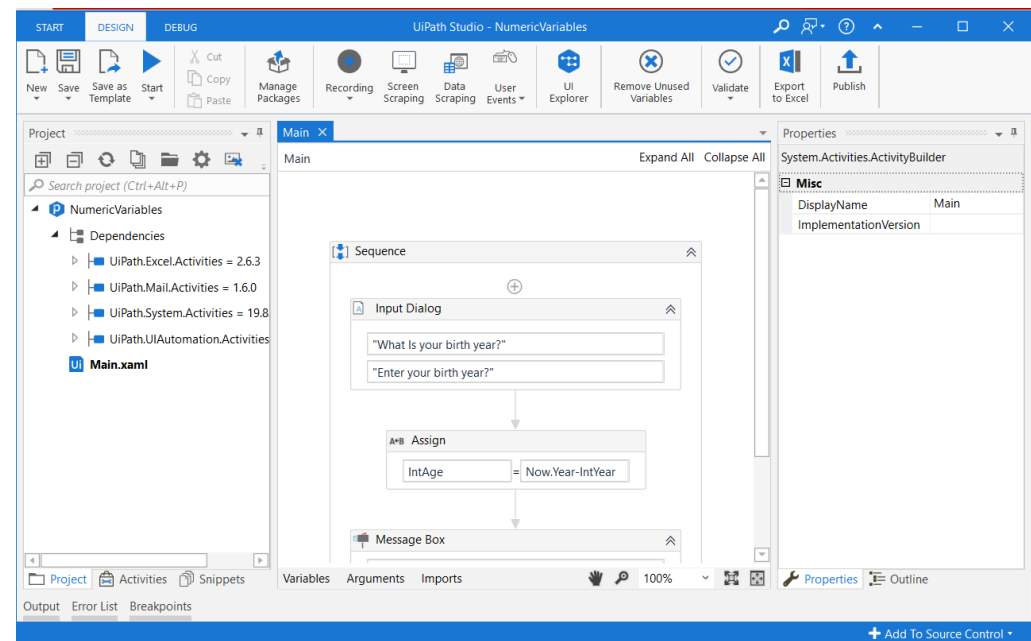
# The User Interface. Ribbon

- There are 3 menus on the top ribbon:
  - **Start**- to create a new project, i.e., a new process;
    - It Connects one activity to another without cluttering the project;
  - **Design**- to design the process;
    - Actions allowed: add activities (sequences, flowcharts, state machine), UI interaction, export to Excel, publish to Orchestrator;
  - **Execute/Debug**- debug related actions;
    - Actions allowed: validate, run, debug, monitor the execution step by step;



# The User Interface. Panels

- Main areas (panels) in UiPath Studio:
  - design time:
    - **Project, Activities, Snippets;**
    - **Designer;**
    - **Variables, Arguments, Imports;**
    - **Properties Panel, Outline Panel;**
  - run/debug time:
    - **Output Panel, Locals Panel;**
    - **Error List, Breakpoints.**





# Variables. Data Types

- **Variables** are used to store different types of data: numeric, text, image, file, colour;
- main types of variables:
  - **Integer**;
  - **String** – with quotes, e.g., “abc”, “123”;
  - **Boolean** = {True, False};
  - **Generic** – almost any data type;
  - **Array of [T]** – all values have the same type;
- a variable defined within an activity is available in all activities included below;
- **Variable Panel** shows the properties of the defined variables:
  - Name;
  - Type;
  - Scope;
  - Default value;
- it presents the variables available in the selected activity.

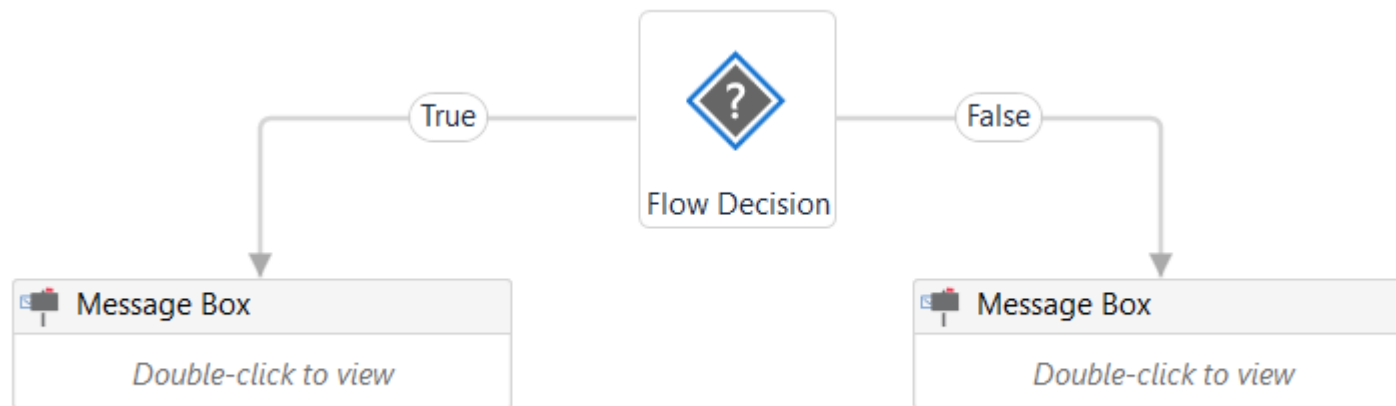
# Choices. If Activity

- **If** activity:
  - It split the sequence vertically;
  - adequate for short linear branches;
- *cons*:
  - more than one if else if chained affects perception on the screen;

The image shows a screenshot of the 'If' activity interface in a flowchart tool. The interface is a rectangular box with a light gray border. At the top, there is a header bar with the text 'If' on the left and a blue information icon on the right. Below the header, the word 'Condition' is displayed. Underneath 'Condition' is a text input field with the placeholder text 'Enter a VB expression'. Below the input field, the interface is split into two columns. The left column is labeled 'Then' and the right column is labeled 'Else'. Each column contains a large rectangular box with the text 'Drop activity here' centered inside it.

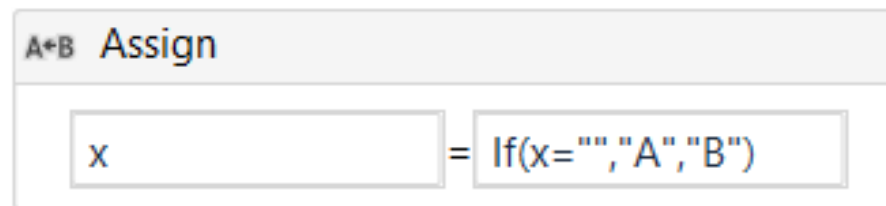
# Choices. Flow Decision Activity

- **Flow Decision** activity:
  - It shows important decision logic and related conditions;
  - It can be placed inside a **flowchart**.



# Choices. If Operator

- **If** operator:
  - this is the VB operator;
  - useful for **small local conditions or data computations**;
  - it reduces the block to a single **Assign** activity;



# Choices. Switch Activity

- **Switch** activity:
  - if can be used together with If operator; to streamline and compact if else id cascade, with distinct conditions and activities per branch;

Switch

Expression

Default

Write Line

Text

Case 1 WriteLine

Case 2 WriteLine

[Add new case](#)

Common

DisplayName Switch

Misc

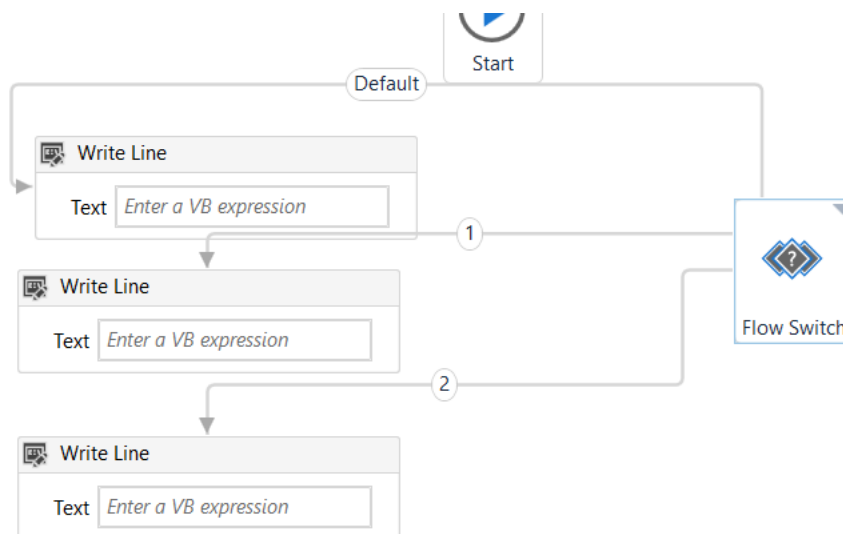
Expression  ...

Private ☐

TypeArgument  ▼

# Choices. Flow Switch Activity

- **Flow Switch** activity:
  - if selects the next node depending on the value of expression;
  - **Flow Switch** activity in *flowcharts* = **Switch** activity in *sequences*;



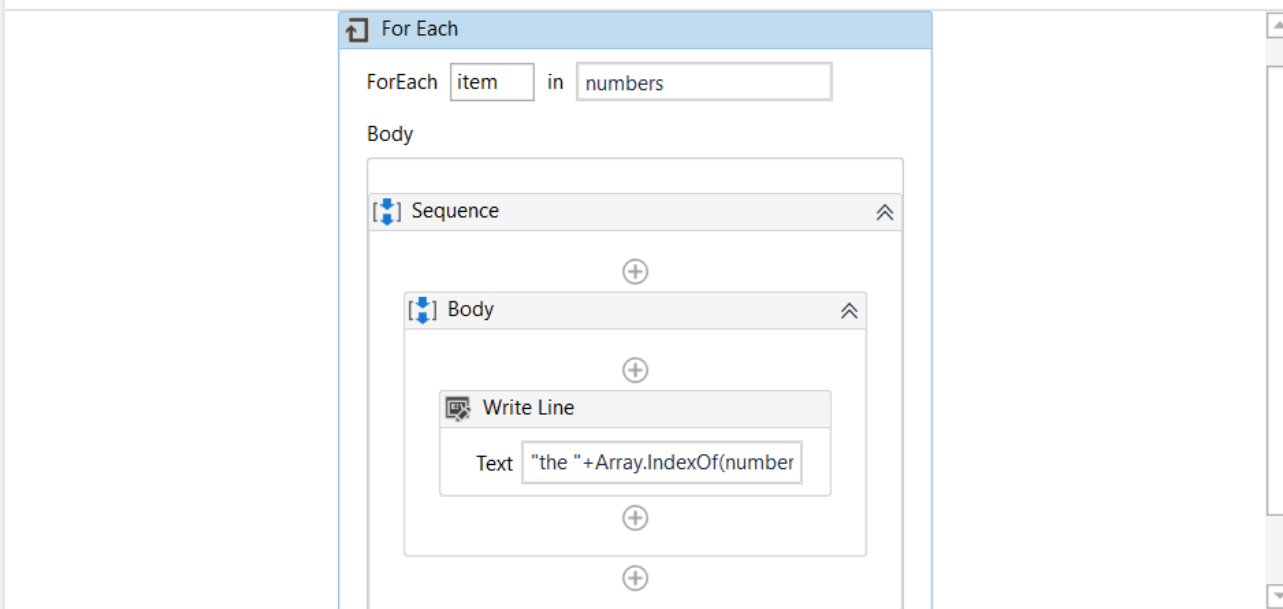
DisplayName	Flow Switch
Expression	a
TypeArgument	Int32

# Demo 1

- Create a process that performs the following actions:
  - 1. *read* the name of the person;
  - 2. *read* the birth year;
  - 3. *computes* the age in years (considering the current date);
  - 3. *print* “Congratulations, Z! You are x years old!”

# Control Flow. For Each Activity

- **For Each** activity:



The screenshot shows the configuration of a 'For Each' activity in a flowchart. The 'ForEach' property is set to 'item' and 'in' is set to 'numbers'. The 'Body' contains a 'Sequence' container, which contains a 'Body' container, which contains a 'Write Line' activity. The 'Text' property of 'Write Line' is set to 'the ' + Array.IndexOf(number'.

Name	Variable type	Scope	Default
numbers	Int32[]	Flowchart	{0,2,4,6,8,10}
x	String	Sequence	"c"
-	Int32	Sequence	0



# Control Flow. While Activity

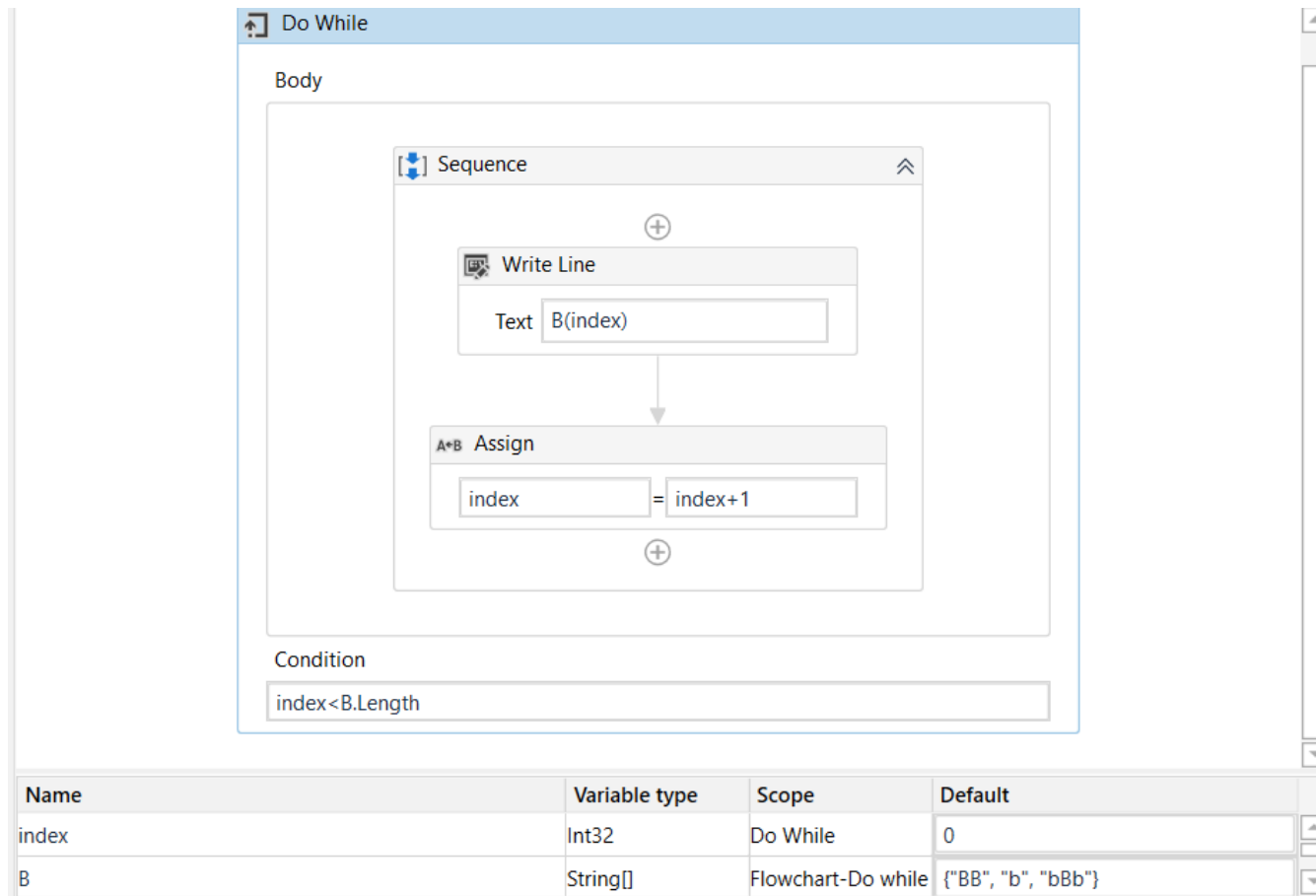
- **While** activity:

The screenshot displays a Visual Studio .NET IDE window with a flowchart project. A **While** activity is selected, showing its configuration in the Properties window. The **Condition** is set to `index < z.Length`. The **Body** contains a **Sequence** container with two activities: a **Write Line** activity (Text: `"item"+z(index).ToString`) and an **Assign** activity (index = index+1). The **Properties** window on the right shows the **Common** and **Misc** tabs. The **Misc** tab shows the **To** property set to `index` and the **Value** property set to `index+1`.

Name	Variable type	Scope	Default
index	Int32	While	0
z	Int32[]	Flowchart	{1,3,5,7,9}

# Control Flow. Do While Activity

- **Do While** activity:



# Demo 2

- Create a process that performs the following actions:
  - 1. *generates* an integer number from 1 to 7;
  - 2. *try to guess* the generated number;
  - 3. *compare* the generated value
    - 3.1. print the message “Enter a smaller number!” or
    - 3.2. print the message “enter a bigger number!”;
  - 4. *repeat* steps 2 and 3 until you succeed to find the number;
  - 5. *show* the message “Well done!!!”

# References

- UiPath Academy - <https://academy.uipath.com>
  - Awareness Training;
  - Level 1 – Foundation Training;
- UiPath Docs - <https://docs.uipath.com/studio>