LECTURE 03. COLLECTIONS. PART I

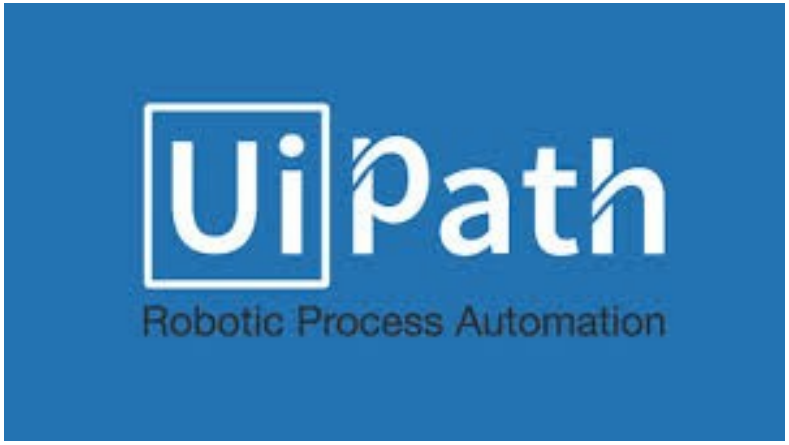
Robotic Process Automation
[17 October 2023]

Elective Course, 2023-2024, Fall Semester

Camelia Chisăliţă-Creţu, Lecturer PhD
Babeş-Bolyai University

Acknowledgements

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



Contents

Arguments. Definition

- **Arguments** are
  - used to pass data from a project to another;
  - similar to parameters in method definition;

- **advantages**:
  - increase workflow readability;
  - increase sequence/flowchart reusability in workflows;

- **arguments vs variables:**
  - **argument** – it stores data dynamically and passes it on, between *automations*, i.e., projects;
  - **variable** – it passes data between *activities*.

see **Demo1 - Arguments**

UiPath™

Arguments. Types. Direction

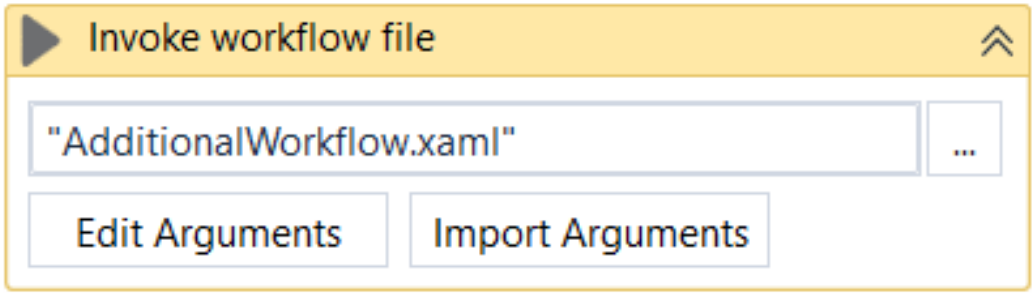
- **argument types** – similar to *variable types* available in UiPath;
- **argument direction** – it indicates *where* the information stored in them is supposed to go;
- possible argument directions:
  - **In** – data can be used in the current project only; it was sent by another project;
  - **Out** – data can be used outside the current project;
  - **In/Out** – **In** + **Out**.

see **Demo1 - Arguments**

UiPath™

Arguments. Invoke Workflow File Activity

- **Invoke Workflow File activity**
  - allows to invoke a specified workflow, passing a list of input/output arguments (parameters);
- **Properties to set:**
  - **WorkflowFileName** - a string with the file path of the **.xaml** file to be invoked; the file path is relative to the current project folder;
  - **Arguments** - the parameters that are passed to the invoked workflow;
  - **Isolated** - if checked, the invoked workflow runs in a separate Windows process; this is useful when isolating a faulty workflow from the main workflow (invoker).



see **Demo1 - Arguments**

Demo 1

- **Create a process that performs the following actions:**
  - **1. *read* two integer numbers;**
  - **2. *compute* the maximum value between the given numbers;**
    - ***define a distinct workflow having:***
      - ***two input arguments;***
      - ***one output argument;***
    - ***invoke the defined workflow;***
  - **3. *print* the computed maximum value.**

see **Demo1 - Arguments**

Variable Categories

- **Scalar** - single value of a fixed type:
  - **Int32, Boolean, Character, Date Time**, etc.;
- **Collections** – *single dimension* structures, where multiple values are viewed as an entity:
  - **Array, List, Queue;**
  - **String;**
  - **Dictionary;**
- **Tables** – *two dimensional* structures, where multiple values are viewed as an entity:
  - **Data Table.**

Arrays. Details

- **Array** characteristics in UiPath:
  - **an array has a fixed length**, set at instantiation/initialization time;
  - it implements the **IEnumerable** interface ==> can be iterated by using a **For Each** activity.

see **Demo2 - Arrays**

UiPath™

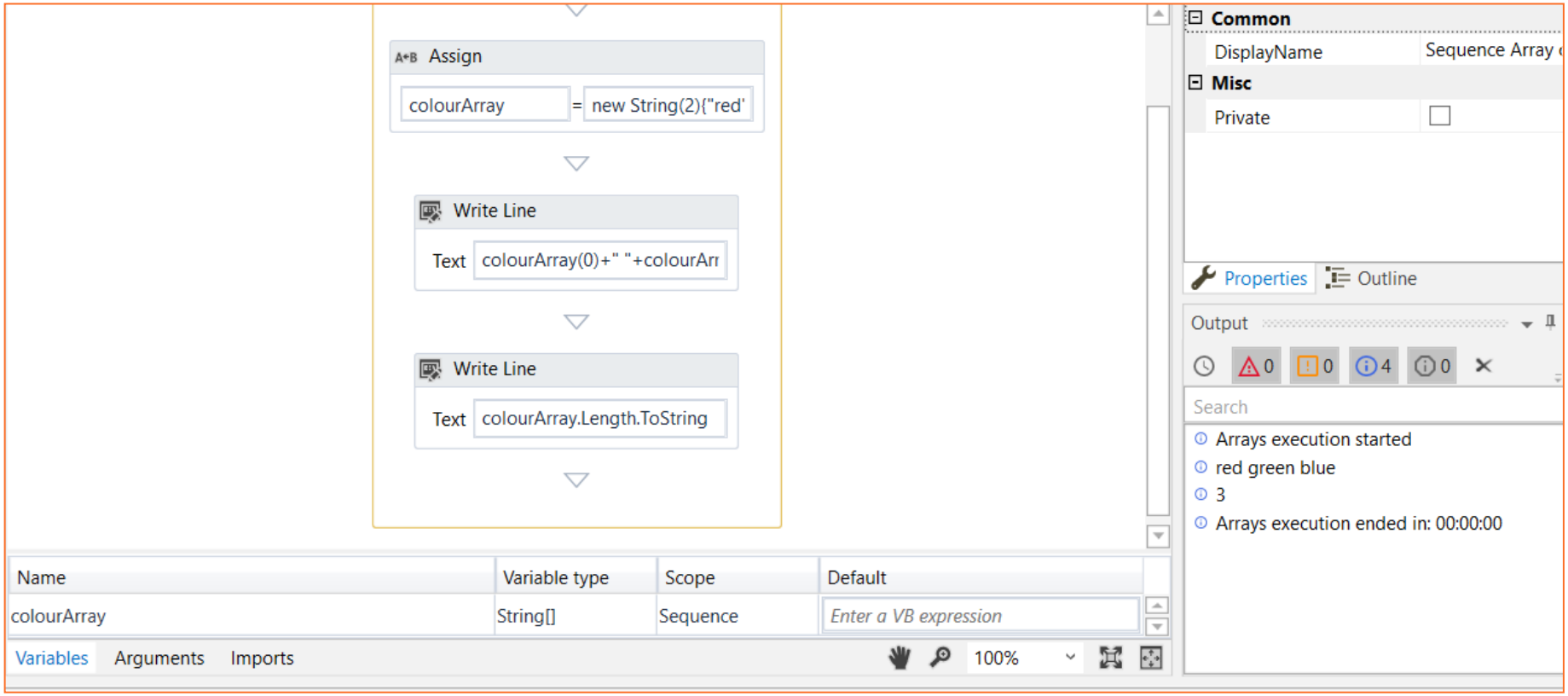
Arrays. Declaration. Instantiation. Initialization

- ways to **declare/instantiate/initialize** an array:
  - **Variables Panel:**
    - **Name:** colourArray; **Type:** String[]; **Default:** {“red”, “green”, “blue”} *//Length=3*
  - **Assign** activity:
    - colourArray= new String(){"red", "green", "blue"}*//Length=3*
    - colourArray= new String(2){"red", "green", "blue"}*//Length=3*
    - colourArray= new string(2){}*//values are set later, Length=3, valid indices: 0, 1, 2*
- ways to **set/change the values** in arrays:
  - **Assign** activity:
    - colourArray(0)= “red"// overrides or initializes the value on index = 0

see **Demo2 - Arrays**

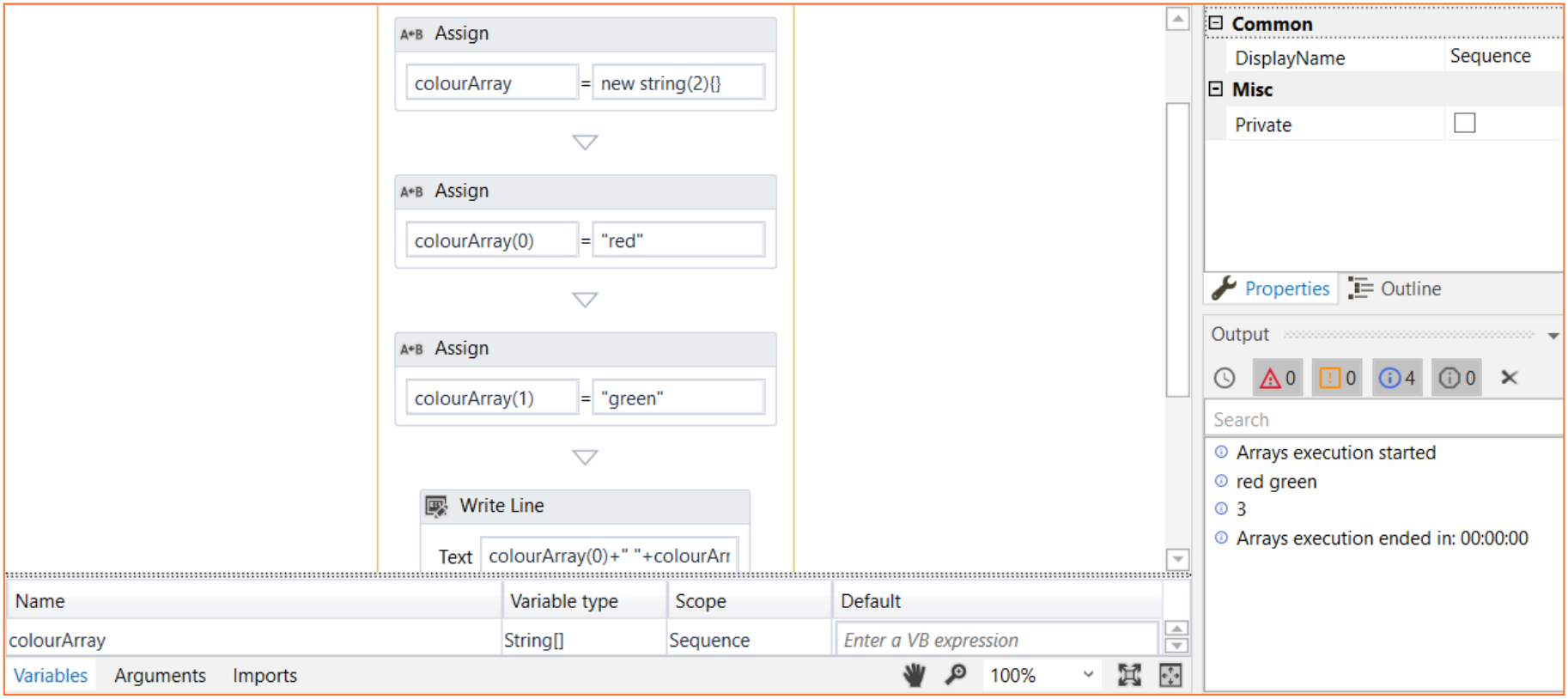
UiPath™

Arrays. Example 1



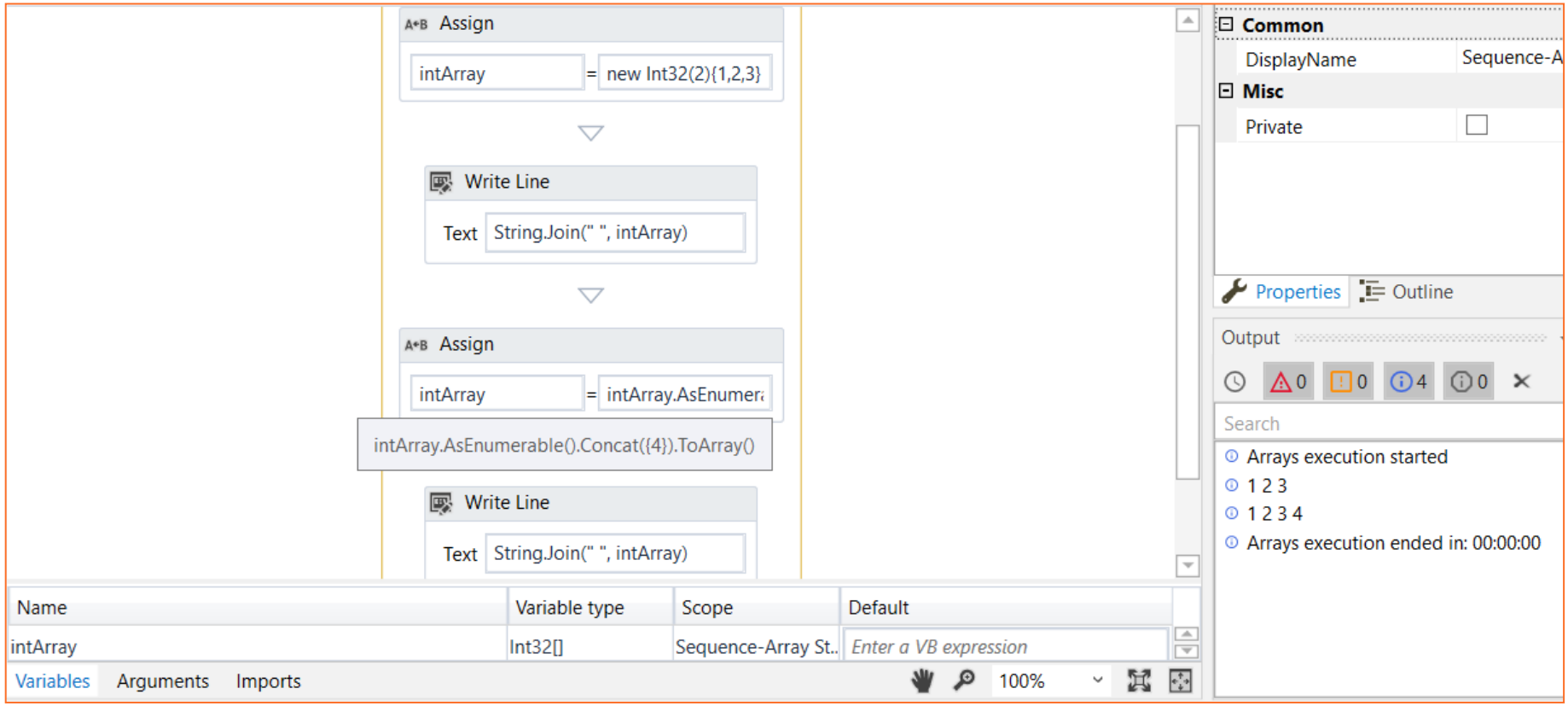
see **Demo2 - Arrays**

Arrays. Example 2



see **Demo2 - Arrays**

Arrays. Example 3



see **Demo2 - Arrays**

Demo 3

- **Create a process that performs the following actions:**
  - **1. *instantiate* an array of 5 integer numbers;**
  - **2. *set* the values of the generated numbers from 1 to 10 and:**
    - **2.1. *print* the generated number;**
    - **2.2. *check* if the number is even:**
      - **2.2.1. if TRUE then *print* number+1 ;**
      - **2.2.2. if FALSE then *print* number-1 ;**
      - ***use For Each activity to iterate the array;***
      - ***use Log activity to print the generated values;***
  - **3. *compute* the sum of numbers in the array;**
  - **4. *print* the sum;**
  - **5. *print* the array;**
    - ***use a String-based method.***
  - **Discuss the presence nd usage of the local variable *item***

see **Demo3 – RandomNumbersArray**

Lists. Details

- **List** characteristics in UiPath:
  - **a list has a flexible length**;
  - it implements the **IEnumerable** interface ==> can be iterated by using a **For Each** activity.

see **Demo4 - Lists**

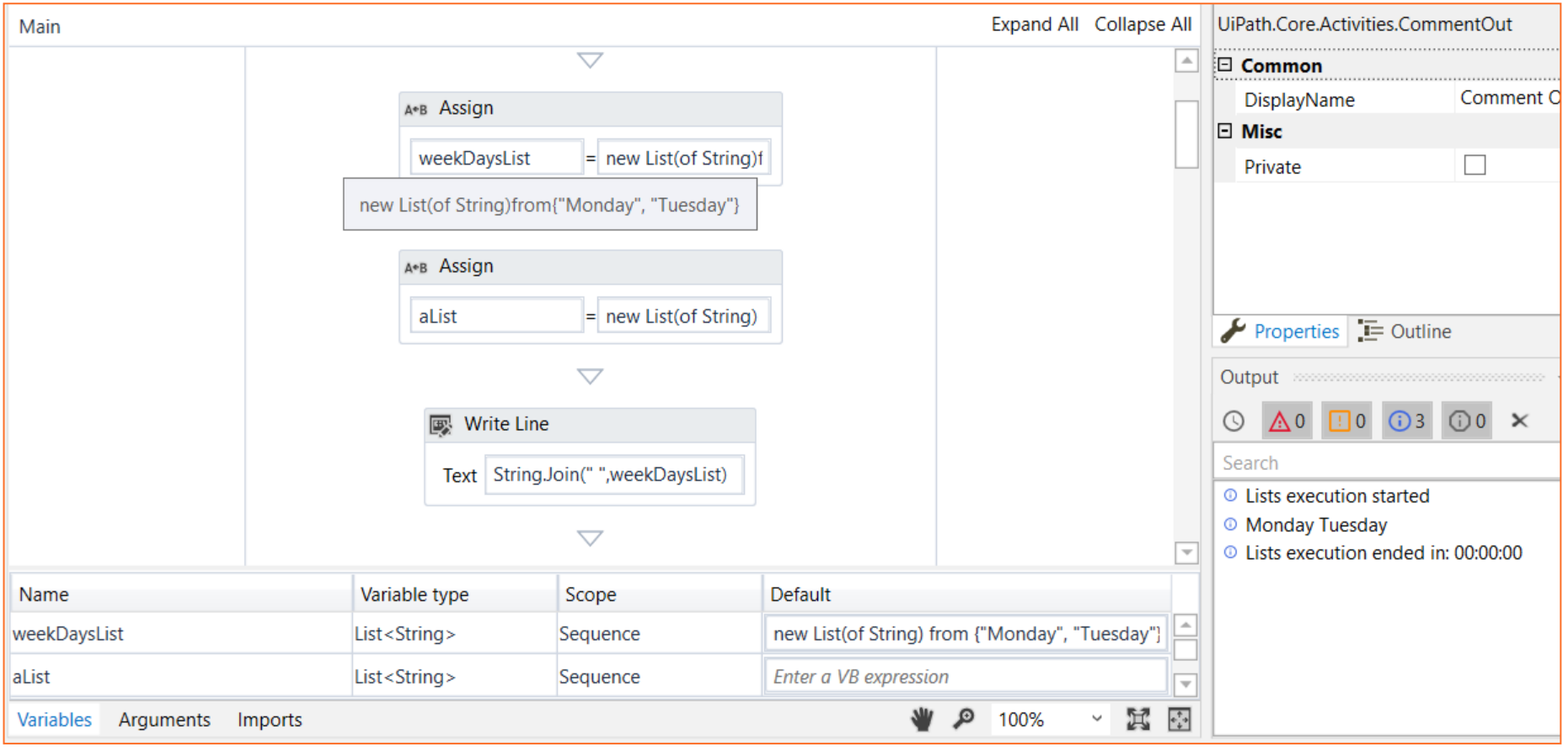
Lists. Declaration. Instantiation. Initialization

- ways to **declare/instantiate/initialize** a list:
  - **Variables Panel:**
    - **Name:** weekDaysList; **Type:** List<String>;
    - **Default:** new List(of String) from {"Monday", "Tuesday"} *//Length=2*
  - **Assign** activity:
    - weekDaysList = new List(of String)from{"Monday", "Tuesday"}*//Length=2*
    - weekDaysList = new List(of String) *//Length=0, values are added later*
- way to **change values already added to the list**:
  - **Assign** activity:
    - weekDaysList(0)= "Friday“ // overrides the value on index = 0

see **Demo4 - Lists**

UiPath™

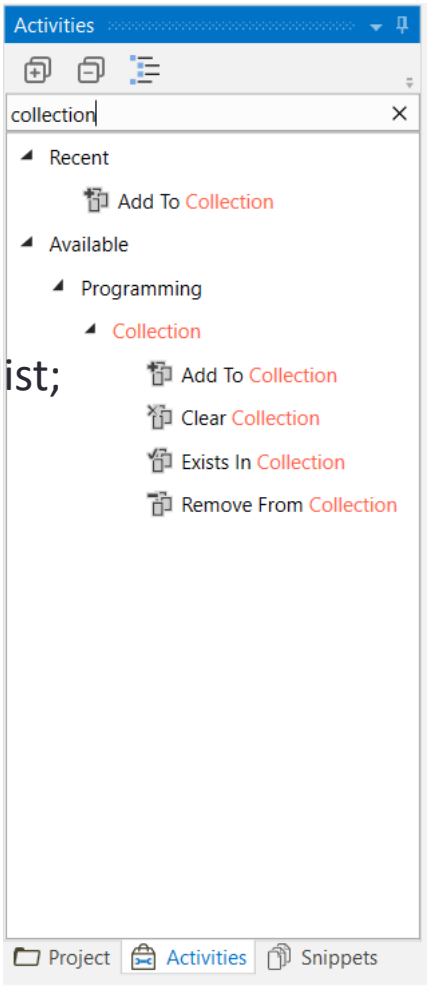
Lists. Example 1

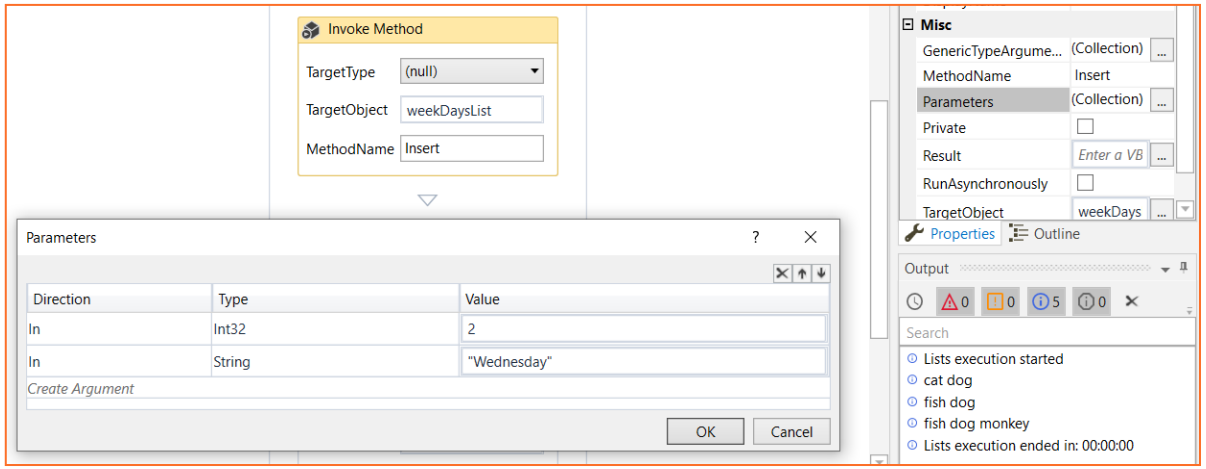


see **Demo4 - Lists**

Lists. Operations

- ways to apply predefined operations on lists:
  - **Invoke Method** activity:
    - the user needs to know the signature of the use method;
    - it can be used to *Add, Insert, Remove, etc.* an item to/from a list;
  - **Collection** package of activities:
    - **Add To Collection;**
    - **Clear Collection;**
    - **Exists In Collection;**
    - **Remove From Collection;**

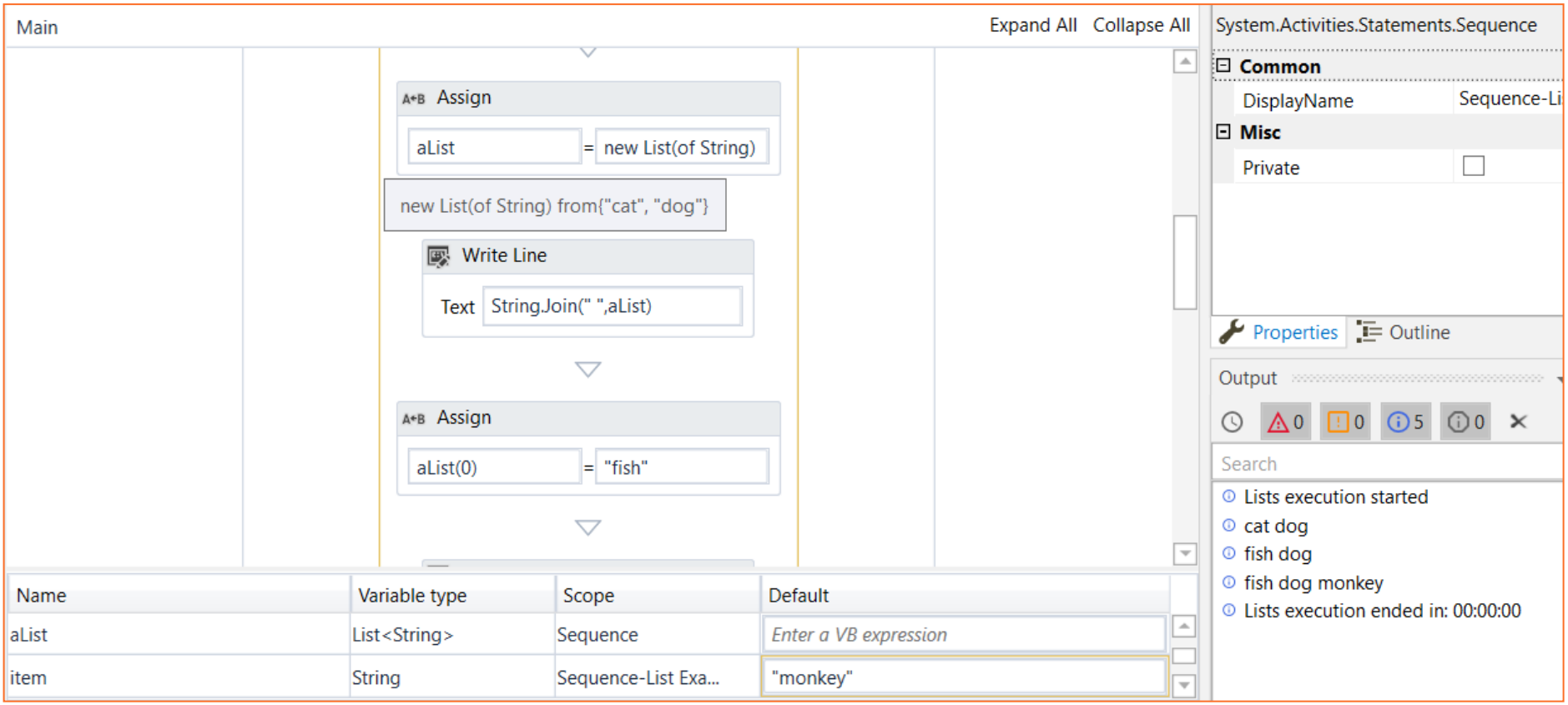




see **Demo4 - Lists**

UiPath™

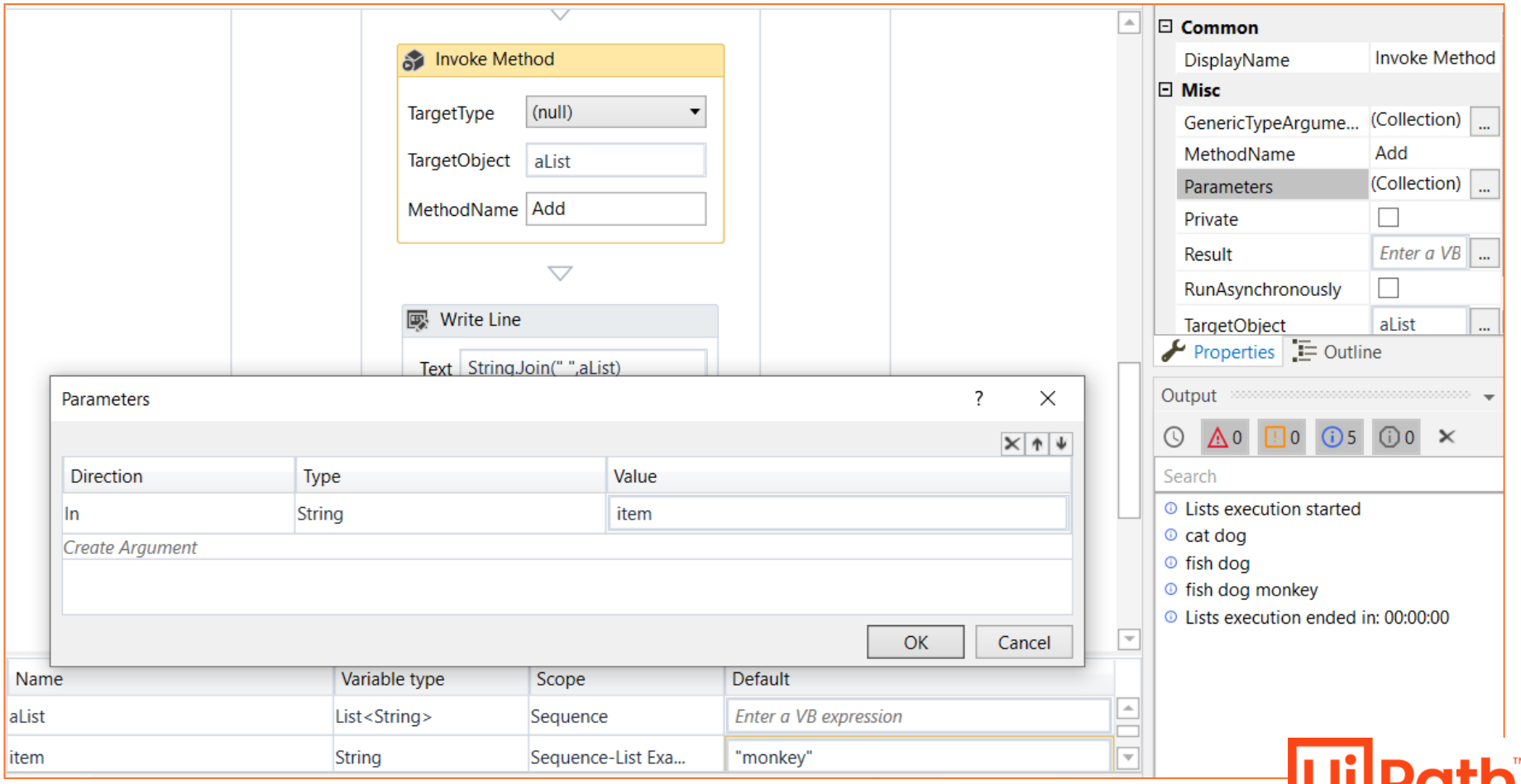
Lists. Example 2A. Set/update an item



see **Demo4 - Lists**

Lists. Example 2B. Add an item

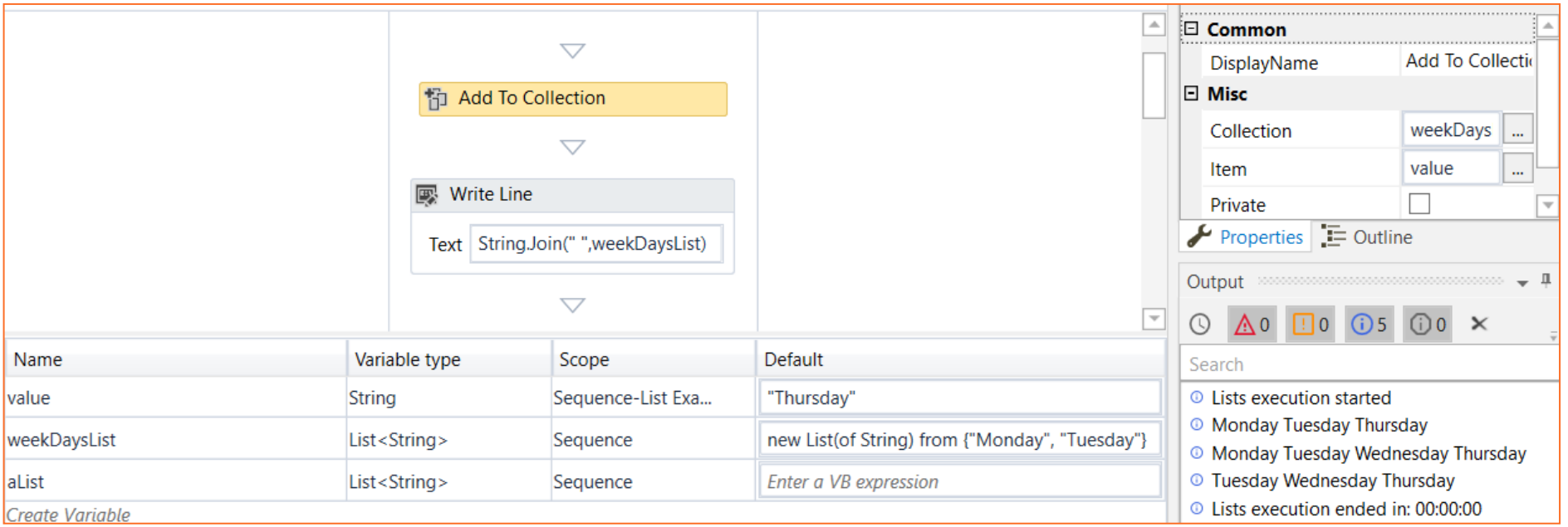
- **Invoke Method** activity can be used to *add, insert, remove, etc.* an item to/into/from a list;



see **Demo4 - Lists**

Lists. Example 3. Collection Package

- **Add To Collection** activity is used add an item to the list;



see **Demo4 - Lists**

Demo 5

- **Create a process that performs the following actions:**
  - **1. *create* a list of the following items:** {“still water”, “sparkling water”, “herbal tea”, “coffee”, “wine”, “juice”, “green tea”, “black tea”};
  - **2. *generate* an index value (from 0 to 7);**
  - **3. *print* the message *“Do you like to drink some juice/green tea/…?”* and enter the answer (*yes* or *no*):**
    - **3.1. if “YES” move the item to the front of the list;**
    - **3.2. if “NO” move the item to the end of the list;**
  - **4. *repeat* step 3. *four* times;**
  - **5. *print* the bill, i.e., the list of accepted drinks.**

see **Demo5 - DrinksLists**

References

- UiPath Docs - https://docs.uipath.com/
- UiPath Studio Docs - https://docs.uipath.com/studio/standalone/2023.4
- UiPath Forum - https://forum.uipath.com/
- UiPath Academy - https://academy.uipath.com/