Intro in RPA

3rd Lab

UiPath™

Extracting data from a website

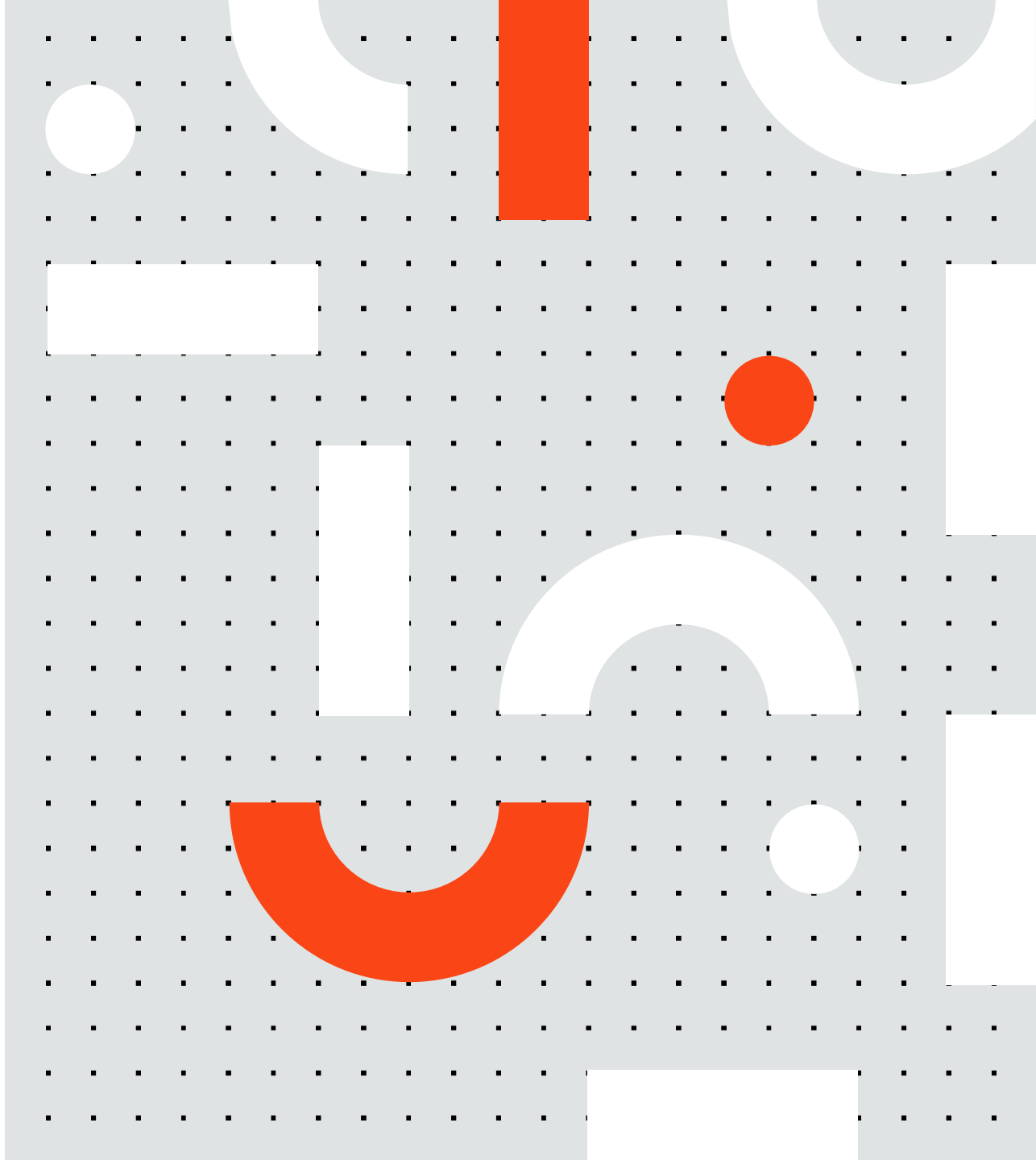
1. Extract data from a website

Objective: To code a Robot in UiPath Studio to scrape data from a website and store it in .CSV File.

Learning Outcomes

After completion of this exercise you will get familiar with the following:

✓ **"Sequence"**
✓ **"Comment"** and "**Annotation".**
✓ **"Open Browser"** activity.
✓ **"Type into"** activity.
✓ **"Browser scope"** activity.
✓ **"Extract data"** activity.
✓ **"Write CSV"** activity.

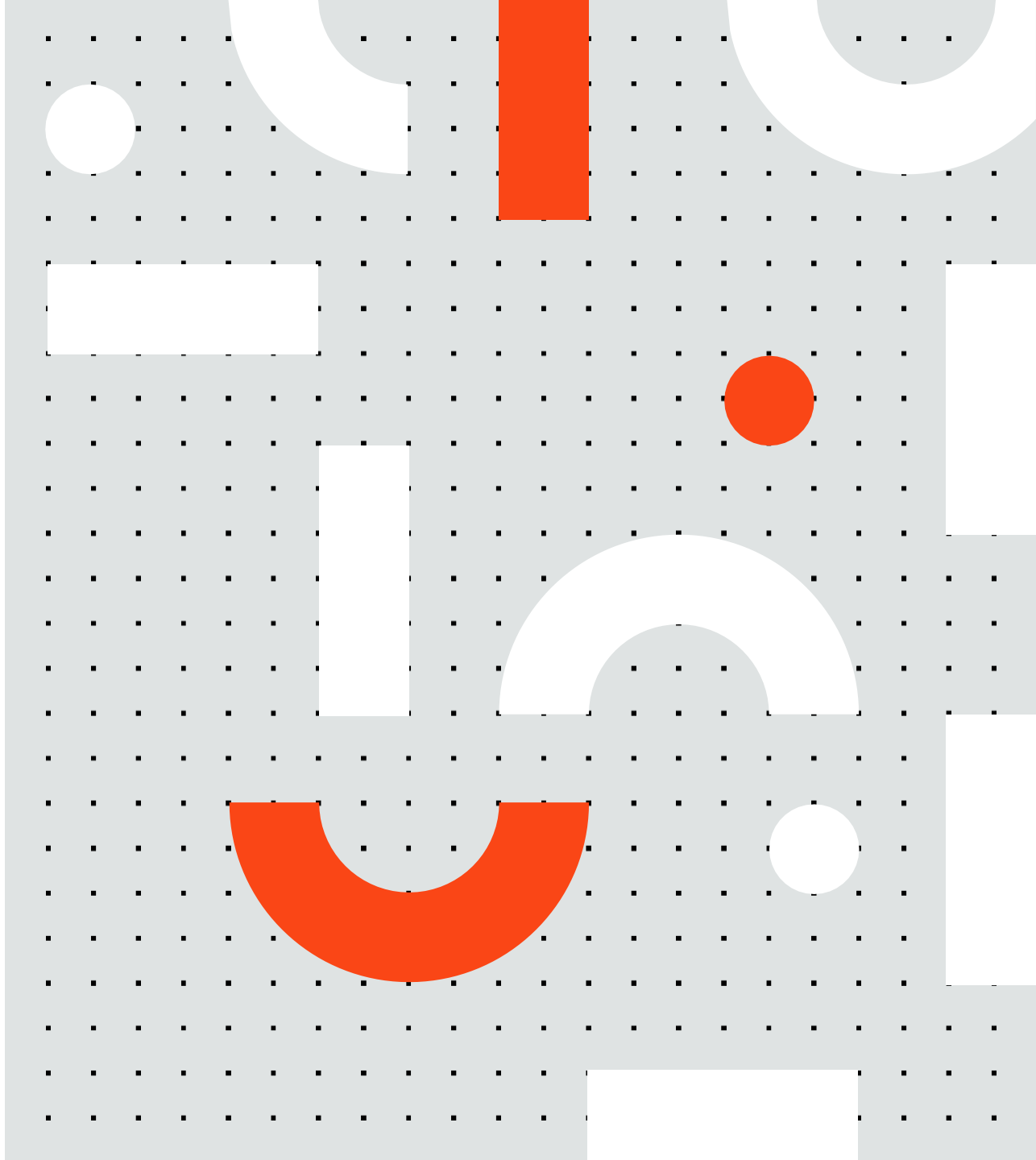


Extracting data from a website

1. Extracting data from a website

Algorithm:

Step 1: START
Step 2: Open the URL using Open Browser Activity
Step 3: Declare variables as ‘CSVFile’, ‘ExtractedDT’, ‘SearchItem’, ‘URL’.
Step 4: Use the variables in the different activity blocks to search, find the given item
Step 5: Output the result in the write csv file activity
Step 6: STOP



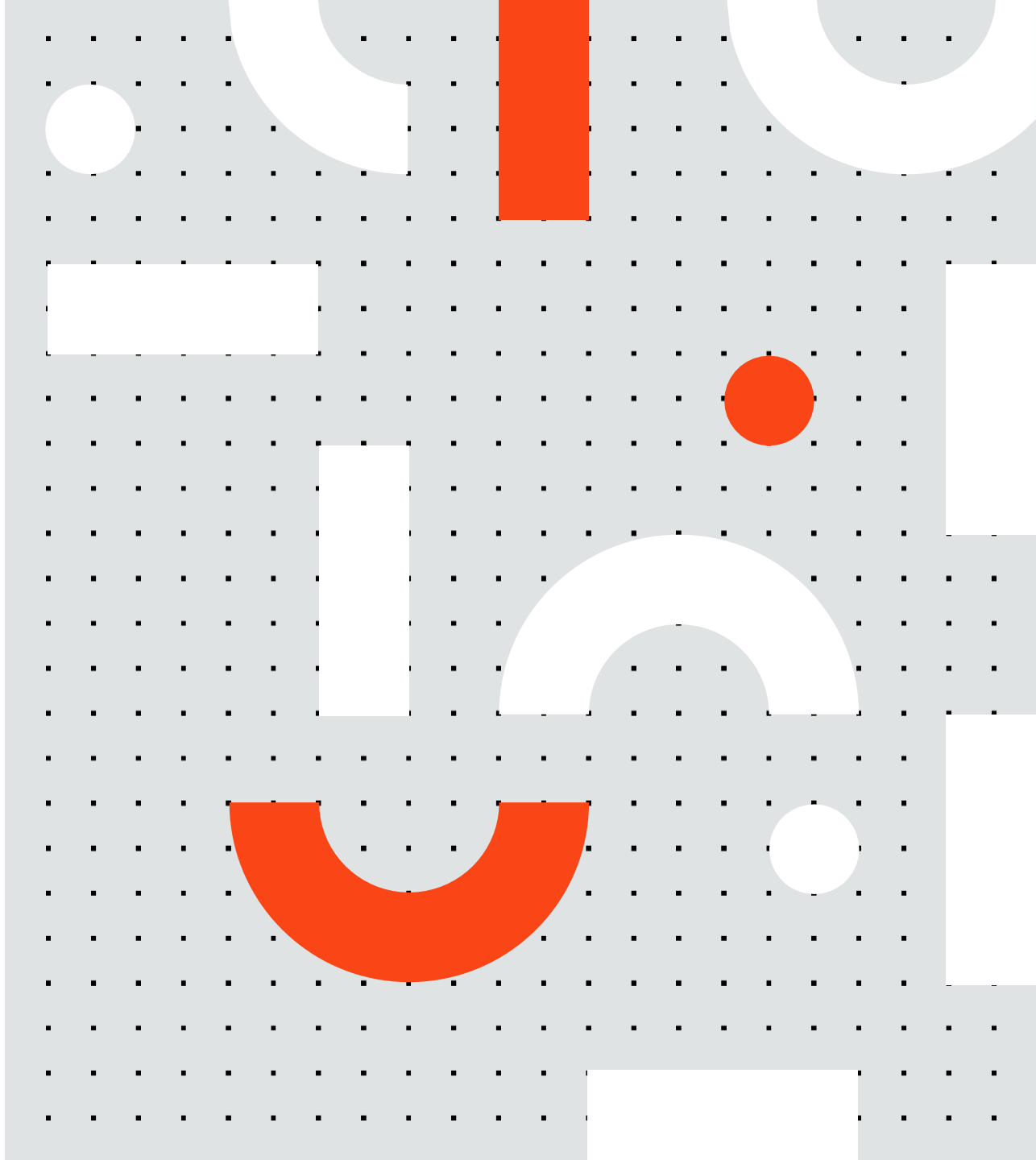
Step by Step process:

Step 1: Open UiPath Studio.
Step 2: Create the process and name it.
Step 3: Create new xaml file as **Sequence**.
Step 4: Name the new file **"Extract data from website".**
Step 5: Drag the '**Open browser'** activity and drop it in the workflow
Step 6: Create an **in argument** corresponding to the URL (e.g. in_URL). Value set in Main: www.amazon.com. This argument will be the input for the '**Open browser'** activity.
Step 7: Drag the **Type Into** activity and drop it in the workflow
Step 8: Change the properties accordingly
Step 9: Create an **in argument** e.g. **in_SearchItem** – contains the name of the product we are searching for (e.g. iPhone). This argument will represent the **input for Type into activity.**
Step 10: Add "Enter" key in the Type Into activity. Check **"Simulate Type"** property deactivated.
Step 11: Drag and drop "Attach Browser" activity.
Step 12: Drag and drop "Extract Data" activity into the previous added "Attach Browser"
Step 13: Create the **output argument** of the Extract Data activity e.g. **out_ExtractedInfoDT**. Type DataTable.

UiPath™

Step by Step process:

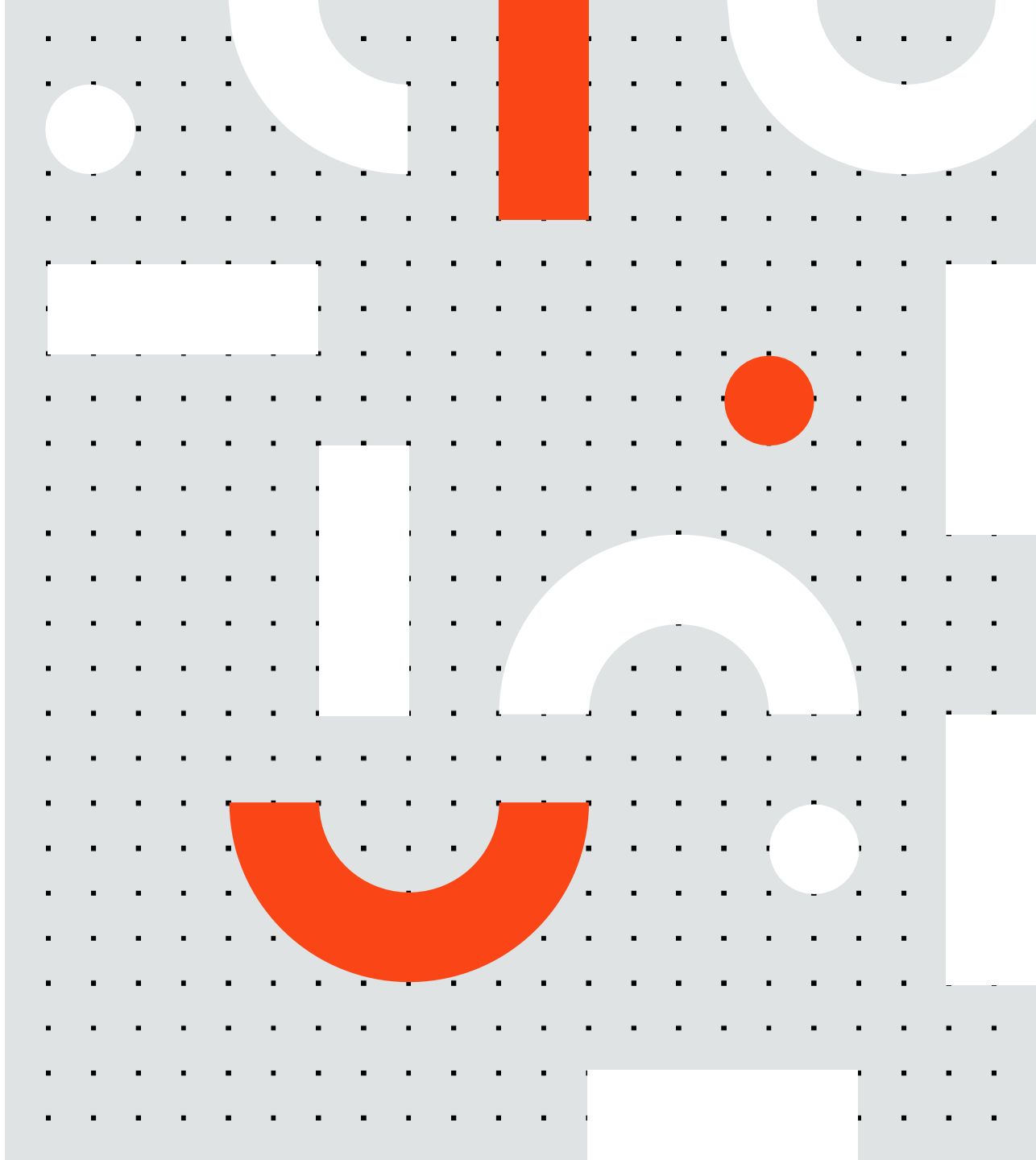
Step 14: Create a new xaml file as **Sequence**.
Step 15: Name the new file **“Write extracted data into csv”.**
Step 16: Create the **input argument** containing the data table extracted previously e.g. in_ExtractedInfoDT.
Step 17: Drag the “**Write CSV**” activity from the Activity panel.
Step 18: Create an in argument that will store the path to the .csv file e.g. CSVFilePath. Provide the argument to “Write CSV” activity.
Step 19: The input argument containing the data table will represent the Input for “Write CSV” activity.



Check the invoices issued to bankruptcy clients

1. Check the invoices issued to bankruptcy clients

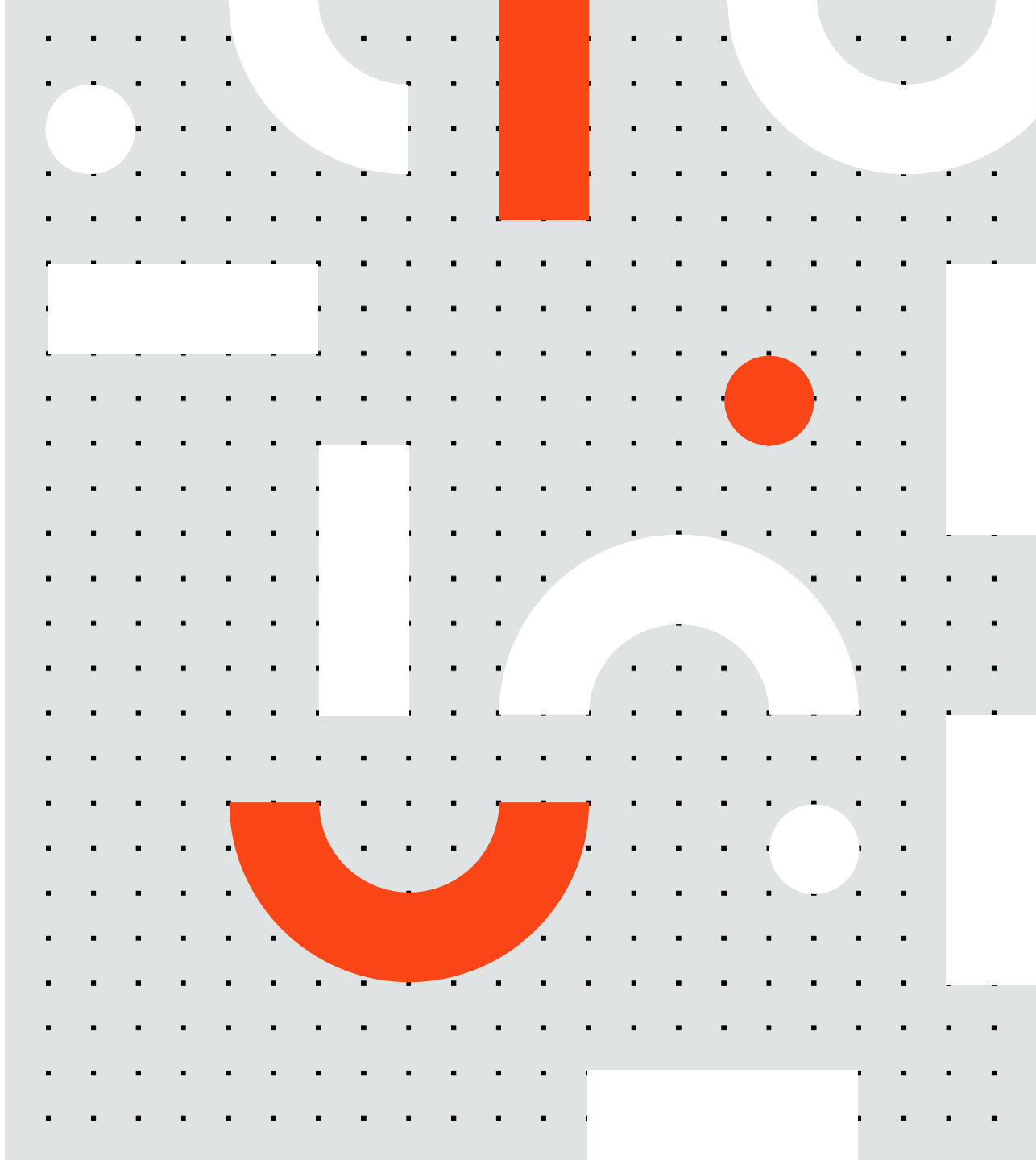
Objective: Check which of the invoices in an Excel file were issued to clients in bankruptcy and calculate the sum of the invoices to be recorded as loss.
Input: excel & csv files.
Output: excel file.



Check the invoices issued to bankruptcy clients

Algorithm:

Step 1: Use a **'Read Range'** activity to read the .xlsx file and store it in a newly created **DataTable variable** ("dt_Invoices");
Step 2: Use a **'Read CSV'** activity to read the .csv file and store the content in a newly created **DataTable variable** ("dt_Clients")
Step 3: Add a **'Join Data Tables'** activity to bring together the data from both variables to a new one ("dt_Results"). **Input Data Table 1** should be "dt_Clients" and **Input Data Table 2** should be "dt_Invoices". Use the **"ClientName" columns** in both DataTables as the Join criterion and the **'Left' Join Type** (so that we keep only the invoices that were issued for clients in the second DataTable).
Step 4: Check which invoices in "dt_Results" are issued to companies that are bankrupt, by using a **'Filter Data Table'** activity. For this, select the **'Keep' radio button** and use **'IsBankrupt'** as the **filtering** criterion (value "TRUE"). You can use this activity to remove some of the columns that you don't need - go to the **'Output Columns'** tab, check the **'Remove'** button and write down the names of the **columns you don't need** ("ClientName" and "ClientId").



Check the invoices issued to bankruptcy clients

Algorithm:

Step 5: To calculate the loss from the companies that are bankrupt, **loop** through the rows in the filtered Data Table and add the content of the **“InvoiceValue” column**, converted to Integer, to a variable in which we will store the sum - “TotalLoss” of Integer type. Use a **'For Each Row'** and an **'Assign'** activity in the Body with the following method: **TotalLoss = TotalLoss+Cint(row("InvoiceValue"))**;
Step 6: Write the filtered Data Table to a new Excel file using the **'Write Range'** activity. Additionally, you can add the total loss information to a cell that is outside the main table, for example the H4 cell, using the **'Write Cell'** activity.)