Architecture Used:

Package "baba.enumerate":

Interface "EnumWords": Encapsulates all the words of the game.
Three Enums implement this interface:

- "**EnumNouns**": Contains all possible nouns in the game (Baba, Wall, etc.).
- "**EnumOperators**": Contains all possible operators in the game (only "is" in this version).
- "**EnumProperties**": Contains all possible properties in the game (You, Stop, etc.).

All three have a method to determine if a string describes a noun for "EnumNouns," an operator for "EnumOperators," or a property for "EnumProperties."

Package "baba":

Representation of game elements:
Class "Element": Represents a game element, containing information about its position (positionx and positiony) and its "type" (e.g., element "Baba," a text block "You," etc.), which corresponds to an "EnumWords." It includes methods to determine if an element is above, adjacent, or at the same position as another element.

Representation of game properties:

Interface "Property": Encompasses all game properties.

Abstract class "AbstractProperty": Implements the "Property" interface and contains all the methods that a property should have. Dx and dy represent directions, and the "elements" ArrayList contains all elements with the property. This class is extended by nine classes representing different properties such as "You," "Win," "Stop," and more.

Representation of a game level:
Abstract class "AbstractLevel": Contains all the methods necessary for level creation. It has four hashmaps:

- "**elements**": Contains all game elements except text blocks. Any modification to an element is done through this hashmap.
- "**texts**": Contains only text blocks, used for generating level rules.
- "**properties**": Contains game properties. Any modification to a property is done through this hashmap.
- "**elementsImage**": Contains all game elements for displaying. It helps associate images with their corresponding elements.

Class "Level": Contains methods for configuring levels using commands and generating default properties for levels (e.g., associating "You" with an element if not specified by commands).

Improvements and Corrections Made After the Presentation:

Complete program overhaul.

- Changed the representation of elements from multiple classes to a single class.
- Added a third parameter (type) to the "Element" class to differentiate elements.
- Used a hashmap that contains all elements instead of having an ArrayList for each element.
- Removed the "Elements" interface and the "AbstractElement" class.
- Changed the representation of text blocks, removed the "Text" class, and created a hashmap that groups all text blocks.
- Changed the way game rules are generated.
- Removed the "Image" interface and the "DrawImages" class, replaced with a hashmap "elementsimage" that contains all images and their associated elements, as well as a "drawLevel" function to iterate through this hashmap and display its elements.
- Generated different levels using text files instead of having a class for each level.
- Changed the "elementsimage" hashmap to a LinkedHashMap to ensure that decorative elements are displayed first and avoid display issues (e.g., Baba behind a decorative element).