P. prudhvi Ramana
SE22USE188
CSE-3

Documentation

1) Project Overview

The FLAMES Game is a Java-based application designed to compute and display the relationship compatibility between two individuals based on their names. It implements the classic FLAMES algorithm and provides an interactive graphical user interface (GUI) for user input and result visualization.

2) Project Structure

The project is modular, following Object-Oriented Programming (OOP) principles. Below is the detailed description of each component:
Classes and Their Responsibilities:

1)FLAMESGame.java:
->Purpose: Implements the FLAMES game logic.
->Extends: Game (abstract class)
Key Functionalities:
->play(String name1,String name2)
Calculates the FLAMES result by reducing the string "FLAMES" based on the unique character count between two names.
->displayResult()
Maps the final character in the string "FLAMES" to a specific relationship type:

- F: Friends
- L: Lovers
- A: Affectionate
- M: Marriage
- E: Enemies
- S: Siblings

2) Game.java:
->Purpose: Abstract class that defines the general structure for games.
Key Methods:
->play(String name1,String name2)
Abstract method for implementing the game's logic.
->displayResult()
Abstract method for displaying results.

3)GUIManager.java
->Purpose: Handles the graphical user interface for the FLAMES game.
->Key Features:
->Input Fields:
Your Name: Text box for entering the first name.
Partner's Name: Text box for entering the second name.
->Play Button:
Executes the FLAMES logic on click.
->Result Label:
Displays the output of the game or error messages.

P. prudhvi Ramana
SE22USE188
CSE-3

->Design Elements:
Styled using Java Swing with pastel and user-friendly color schemes.
->Key Functionality:
Integrates FLAMESGame for computation.
Uses NameValidator to validate user inputs before computation.

4)Main.java:
->Purpose:The entry point for the application.
->Key Functionality:
Initializes and starts the GUI using the GUIManager class.

5)NameValidator.java:
->Purpose: Ensures user inputs are valid.
->Implements: Validator interface.
->Key Functionality:
validateInput(String name1, String name2):
Checks if both names are non-null and non-empty.

6)RelationshipCalculator.java:
->Purpose:Calculates the compatibility score between two names.
->Key Algorithm:
Converts names to lowercase and removes spaces.
Matches and removes common characters between the names.
Appends unmatched characters from one name to the other and returns the length of the resultant string.

7)Validator.java
->Purpose: Interface to enforce validation rules.
->validateInput(String name1, String name2):Abstract method to validate inputs.

3) Execution Flow

GUI Initialization:
->The Main class starts the application by calling .
Input Collection:
->Users enter their names into the input fields.
Validation:
->The NameValidator class validates inputs to ensure they are non-empty.
Relationship Calculation:
->RelationshipCalculator computes the unique character count between the names.
FLAMES Algorithm Execution:
->FLAMESGame.play() reduces the string "FLAMES" based on the computed count to a single character.
Result Display:
->The final result is mapped to a relationship type and displayed on the GUI.

P. prudhvi Ramana
SE22USE188
CSE-3

4) RESULTS