**Implement the following projects which focus on different aspects of Java programming, including:**

Object-Oriented Programming principles

Exception handling

File operations

Data structures

Input validation

User interface design

9. Number Guessing Game

Create a game where users guess a random number with hints using object oriented programming in Java.

Key points:

- Use Random class

- Track number of attempts

- Implement difficulty levels

import java.util.Random;

import java.util.Scanner;

// Game difficulty levels

enum Difficulty {

EASY(1, 50, 10),

MEDIUM(1, 100, 7),

HARD(1, 200, 5);

final int min;

final int max;

final int maxAttempts;

Difficulty(int min, int max, int maxAttempts) {

this.min = min;

this.max = max;

this.maxAttempts = maxAttempts;

}

}

// Custom exception for invalid inputs

class InvalidInputException extends Exception {

public InvalidInputException(String message) {

super(message);

}

}

// Main game logic class

class Game {

private final int targetNumber;

private final Difficulty difficulty;

private int attempts;

private boolean isGameWon;

private final Random random;

public Game(Difficulty difficulty) {

this.difficulty = difficulty;

this.random = new Random();

this.targetNumber = random.nextInt(difficulty.max - difficulty.min + 1) + difficulty.min;

this.attempts = 0;

this.isGameWon = false;

}

public String makeGuess(int guess) throws InvalidInputException {

// Validate input

if (guess < difficulty.min || guess > difficulty.max) {

throw new InvalidInputException(

String.format("Please enter a number between %d and %d",

difficulty.min, difficulty.max)

);

}

attempts++;

if (guess == targetNumber) {

isGameWon = true;

return "Congratulations! You've guessed the number!";

}

if (attempts >= difficulty.maxAttempts) {

return String.format("Game Over! The number was %d", targetNumber);

}

String hint = guess < targetNumber ? "higher" : "lower";

int remainingAttempts = difficulty.maxAttempts - attempts;

return String.format("Try a %s number. %d attempts remaining.",

hint, remainingAttempts);

}

public boolean isGameOver() {

return isGameWon || attempts >= difficulty.maxAttempts;

}

public int getAttempts() {

return attempts;

}

public String getGameStats() {

return String.format("""

Game Statistics:

Difficulty: %s

Total Attempts: %d

Result: %s

""",

difficulty,

attempts,

isGameWon ? "Won" : "Lost"

);

}

}

// Main class to run the game

public class NumberGuessingGame {

private static final Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

while (true) {

playGame();

System.out.print("\nWould you like to play again? (yes/no): ");

String playAgain = scanner.nextLine().trim().toLowerCase();

if (!playAgain.equals("yes")) {

System.out.println("Thanks for playing! Goodbye!");

break;

}

}

scanner.close();

}

private static void playGame() {

// Select difficulty

System.out.println("\nSelect Difficulty:");

System.out.println("1. EASY (1-50, 10 attempts)");

System.out.println("2. MEDIUM (1-100, 7 attempts)");

System.out.println("3. HARD (1-200, 5 attempts)");

Difficulty difficulty;

while (true) {

try {

System.out.print("Enter your choice (1-3): ");

int choice = Integer.parseInt(scanner.nextLine());

difficulty = switch (choice) {

case 1 -> Difficulty.EASY;

case 2 -> Difficulty.MEDIUM;

case 3 -> Difficulty.HARD;

default -> throw new InvalidInputException("Please enter a number between 1 and 3");

};

break;

} catch (NumberFormatException e) {

System.out.println("Please enter a valid number");

} catch (InvalidInputException e) {

System.out.println(e.getMessage());

}

}

// Create and start game

Game game = new Game(difficulty);

System.out.println("\nGame Started! Guess the number!");

while (!game.isGameOver()) {

try {

System.out.printf("\nEnter your guess (%d-%d): ",

difficulty.min, difficulty.max);

int guess = Integer.parseInt(scanner.nextLine());

String result = game.makeGuess(guess);

System.out.println(result);

} catch (NumberFormatException e) {

System.out.println("Please enter a valid number");

} catch (InvalidInputException e) {

System.out.println(e.getMessage());

}

}

// Display game statistics

System.out.println(game.getGameStats());

}

}

o/p

Select Difficulty:

1. EASY (1-50, 10 attempts)

2. MEDIUM (1-100, 7 attempts)

3. HARD (1-200, 5 attempts)

Enter your choice (1-3): 2

Game Started! Guess the number!

Enter your guess (1-100): 45

Try a higher number. 6 attempts remaining.

Enter your guess (1-100): 22

Try a higher number. 5 attempts remaining.

Enter your guess (1-100): 80

Try a higher number. 4 attempts remaining.

Enter your guess (1-100): 95

Try a higher number. 3 attempts remaining.

Enter your guess (1-100): 99

Try a lower number. 2 attempts remaining.

Enter your guess (1-100): 97

Congratulations! You've guessed the number!

Game Statistics:

Difficulty: MEDIUM

Total Attempts: 6

Result: Won

Would you like to play again? (yes/no): yes

Select Difficulty:

1. EASY (1-50, 10 attempts)

2. MEDIUM (1-100, 7 attempts)

3. HARD (1-200, 5 attempts)

Enter your choice (1-3):