Basit Bir tahmin etme oyunu:

**Explanation:**

1. **Main Class**:
   * The entry point of the game. It creates a Player object and starts the game by creating a GuessingGame object.
2. **GuessingGame Class**:
   * This class contains the logic for the number guessing game.
   * A random target number is generated between 1 and 100.
   * The player is given up to 10 attempts to guess the number.
   * Feedback is provided after each guess: if it's too high, too low, or correct.
   * After the game ends (either by guessing correctly or using all attempts), stats are displayed.
3. **Player Class**:
   * Stores player data, including the name, the number of games played, and the number of wins.
   * Includes methods to increment the games played and wins.

**Features:**

* **OOP principles**: The code uses encapsulation (e.g., Player class storing player data and behavior) and abstraction (e.g., the GuessingGame class handling game logic).
* **Game logic**: The game provides feedback and tracks the number of attempts.

This setup can be easily extended with more features or rules. Let me know if you'd like any modifications!

**Açıklama:**

1. **Ana Sınıf**:
   * Oyunun giriş noktasıdır. Bir Player nesnesi oluşturur ve bir GuessingGame nesnesi oluşturarak oyunu başlatır.
2. **GuessingGame Sınıfı**:
   * Bu sınıf, sayı tahmin etme oyununun mantığını içerir.
   * Rastgele bir hedef sayı 1 ile 100 arasında oluşturulur.
   * Oyuncuya sayıyı tahmin etmesi için en fazla 10 deneme hakkı verilir.
   * Her tahminden sonra geri bildirim sağlanır: çok yüksek, çok düşük veya doğru olup olmadığı.
   * Oyun sona erdikten sonra (doğru tahmin ederek veya tüm denemeleri kullanarak), istatistikler görüntülenir.
3. **Oyuncu Sınıfı**:
   * İsim, oynanan oyun sayısı ve galibiyet sayısı dahil olmak üzere oyuncu verilerini depolar.
   * Oynanan oyunları ve kazançları artırmak için yöntemler içerir.

**Özellikler**

* **OOP ilkeleri**: Kod, kapsülleme (örneğin, oyuncu verilerini ve davranışını depolayan Oyuncu sınıfı) ve soyutlama (örneğin, oyun mantığını ele alan GuessingGame sınıfı) kullanır.
* **Oyun mantığı**: Oyun geri bildirim sağlar ve deneme sayısını takip eder.

Bu kurulum daha fazla özellik veya kuralla kolayca genişletilebilir. Herhangi bir değişiklik isterseniz bana bildirin!

*Kodlar:*

// Main.java

public class Main {

public static void main(String[] args) {

Player player = new Player("Player1");

GuessingGame game = new GuessingGame(player);

game.start();

}

}

// GuessingGame.java

import java.util.Scanner;

import java.util.Random;

public class GuessingGame {

private Player player;

private int targetNumber;

private int maxAttempts;

public GuessingGame(Player player) {

this.player = player;

this.maxAttempts = 10; // Maximum attempts allowed

Random random = new Random();

this.targetNumber = random.nextInt(100) + 1; // Random number between 1 and 100

}

public void start() {

Scanner scanner = new Scanner(System.in);

int attempts = 0;

boolean isCorrect = false;

System.out.println("Welcome to the Number Guessing Game!");

while (attempts < maxAttempts && !isCorrect) {

System.out.println(player.getName() + ", guess the number (1-100): ");

int guess = scanner.nextInt();

attempts++;

if (guess == targetNumber) {

isCorrect = true;

System.out.println("Congratulations, " + player.getName() + "! You guessed the correct number.");

player.incrementWins();

} else if (guess < targetNumber) {

System.out.println("Too low! Try again.");

} else {

System.out.println("Too high! Try again.");

}

}

if (!isCorrect) {

System.out.println("Sorry, " + player.getName() + ". You've used all " + maxAttempts + " attempts. The correct number was " + targetNumber + ".");

}

player.incrementGamesPlayed();

displayStats();

}

public void displayStats() {

System.out.println("Game over! Here are your stats:");

System.out.println("Games played: " + player.getGamesPlayed());

System.out.println("Wins: " + player.getWins());

}

}

// Player.java

public class Player {

private String name;

private int gamesPlayed;

private int wins;

public Player(String name) {

this.name = name;

this.gamesPlayed = 0;

this.wins = 0;

}

public String getName() {

return name;

}

public int getGamesPlayed() {

return gamesPlayed;

}

public int getWins() {

return wins;

}

public void incrementGamesPlayed() {

gamesPlayed++;

}

public void incrementWins() {

wins++;

}

}