**TASK 1:** **NUMBER GAME**

import java.util.Random;

import java.util.Scanner;

public class NumberGame {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Random random = new Random();

int min = 1;

int max = 100;

int maxAttempts = 5;

int score = 0;

boolean playAgain;

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

do {

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

int attemptsLeft = maxAttempts;

boolean isGuessedCorrectly = false;

System.out.println("\nI'm thinking of a number between " + min + " and " + max + ".");

System.out.println("You have " + maxAttempts + " attempts to guess it.");

while (attemptsLeft > 0) {

System.out.print("Enter your guess: ");

int userGuess;

// Validate input

if (!scanner.hasNextInt()) {

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

scanner.next(); // discard invalid input

continue;

}

userGuess = scanner.nextInt();

attemptsLeft--;

if (userGuess == targetNumber) {

System.out.println("Correct! Congratulations!! You guessed the number.");

score++;

isGuessedCorrectly = true;

break;

} else if (userGuess < targetNumber) {

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

} else {

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

}

System.out.println("Attempts left: " + attemptsLeft);

}

if (!isGuessedCorrectly) {

System.out.println("Out of attempts! The number was: " + targetNumber);

}

System.out.print("\nDo you want to play another round? (yes/no): ");

playAgain = scanner.next().equalsIgnoreCase("yes");

} while (playAgain);

System.out.println("\nThanks for playing! Your total score: " + score);

scanner.close();

}

}

**TASK 2 : STUDENT GRADE CALCULATOR**

import java.util.Scanner;

public class StudentGradeCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("=== Student Marks & Grade Calculator ===");

System.out.print("Enter the number of subjects: ");

int numSubjects = scanner.nextInt();

// Validate number of subjects

if (numSubjects <= 0) {

System.out.println("Invalid number of subjects.");

return;

}

int[] marks = new int[numSubjects];

int totalMarks = 0;

// Input marks for each subject

for (int i = 0; i < numSubjects; i++) {

System.out.print("Enter marks for Subject " + (i + 1) + " (out of 100): ");

int mark = scanner.nextInt();

// Validate marks

while (mark < 0 || mark > 100) {

System.out.print("Invalid input! Enter marks between 0 and 100: ");

mark = scanner.nextInt();

}

marks[i] = mark;

totalMarks += mark;

}

// Calculate average percentage

double average = (double) totalMarks / numSubjects;

// Determine grade

String grade;

if (average >= 90) {

grade = "A+";

} else if (average >= 80) {

grade = "A";

} else if (average >= 70) {

grade = "B";

} else if (average >= 60) {

grade = "C";

} else if (average >= 50) {

grade = "D";

} else {

grade = "F (Fail)";

}

// Display results

System.out.println("\n--- Result ---");

System.out.println("Total Marks: " + totalMarks + "/" + (numSubjects \* 100));

System.out.printf("Average Percentage: %.2f%%\n", average);

System.out.println("Grade: " + grade);

scanner.close();

}

}

**TASK 3 : ATM INTERFACE**

import java.util.Scanner;

// Step 1: Define an interface for ATM operations

interface BankOperations {

void withdraw(double amount);

void deposit(double amount);

void checkBalance();

}

// Step 2: Create the BankAccount class

class BankAccount {

private double balance;

public BankAccount(double initialBalance) {

this.balance = Math.max(initialBalance, 0); // Ensure non-negative balance

}

public double getBalance() {

return balance;

}

public boolean withdrawAmount(double amount) {

if (amount > 0 && amount <= balance) {

balance -= amount;

return true;

}

return false;

}

public boolean depositAmount(double amount) {

if (amount > 0) {

balance += amount;

return true;

}

return false;

}

}

// Step 3: ATM class that implements BankOperations

class ATM implements BankOperations {

private BankAccount account;

public ATM(BankAccount account) {

this.account = account;

}

@Override

public void withdraw(double amount) {

if (account.withdrawAmount(amount)) {

System.out.println("Withdrawal successful. Remaining balance: ₹" + account.getBalance());

} else {

System.out.println("Withdrawal failed. Either insufficient balance or invalid amount.");

}

}

@Override

public void deposit(double amount) {

if (account.depositAmount(amount)) {

System.out.println("Deposit successful. Current balance: ₹" + account.getBalance());

} else {

System.out.println("Deposit failed. Amount must be greater than 0.");

}

}

@Override

public void checkBalance() {

System.out.println("Current balance: ₹" + account.getBalance());

}

}

// Step 4: Main application class

public class ATMInterfaceApp {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Create account with default initial balance

BankAccount userAccount = new BankAccount(10000);

ATM atm = new ATM(userAccount);

int choice;

do {

System.out.println("\n======= ATM MENU =======");

System.out.println("1. Withdraw");

System.out.println("2. Deposit");

System.out.println("3. Check Balance");

System.out.println("4. Exit");

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

while (!scanner.hasNextInt()) {

System.out.print("Please enter a valid option (1-4): ");

scanner.next();

}

choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.print("Enter amount to withdraw: ₹");

double withdrawAmount = scanner.nextDouble();

atm.withdraw(withdrawAmount);

break;

case 2:

System.out.print("Enter amount to deposit: ₹");

double depositAmount = scanner.nextDouble();

atm.deposit(depositAmount);

break;

case 3:

atm.checkBalance();

break;

case 4:

System.out.println("Thank you for using the ATM. Goodbye!");

break;

default:

System.out.println("Invalid option. Please try again.");

}

} while (choice != 4);

scanner.close();

}

}

**TASK 4 : CURRENCY CONVERTER**

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class SimpleCurrencyConverter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Static exchange rates (base currency: USD)

Map<String, Double> rates = new HashMap<>();

rates.put("USD", 1.0); // US Dollar

rates.put("INR", 83.12); // Indian Rupee

rates.put("EUR", 0.91); // Euro

rates.put("GBP", 0.78); // British Pound

rates.put("JPY", 157.34); // Japanese Yen

System.out.println("=== Currency Converter ===");

System.out.println("Available currencies: USD, INR, EUR, GBP, JPY");

System.out.print("Enter base currency: ");

String baseCurrency = scanner.next().toUpperCase();

System.out.print("Enter target currency: ");

String targetCurrency = scanner.next().toUpperCase();

System.out.print("Enter amount to convert: ");

double amount = scanner.nextDouble();

// Validate currencies

if (!rates.containsKey(baseCurrency) || !rates.containsKey(targetCurrency)) {

System.out.println("Unsupported currency.");

} else {

// Convert: first convert base to USD, then USD to target

double amountInUSD = amount / rates.get(baseCurrency);

double convertedAmount = amountInUSD \* rates.get(targetCurrency);

System.out.printf(" %.2f %s = %.2f %s\n", amount, baseCurrency, convertedAmount, targetCurrency);

}

scanner.close();

}

}