HW-2

1. Write a Java program that takes an integer as input from the user. It should calculate the sum and

product of all input digits. For example, if the user inputs 5143, the sum of digits is (5+1+4+3=13),

and the product of digits is (5\*1\*4\*3=60).

Answer:

import java.util.Scanner;

public class SumAndProductOfDigits {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter an integer: ");

int number = sc.nextInt();

// Initialize variables for sum and product

int sum = 0;

int product = 1;

// Process each digit in the number

int tempNumber = Math.abs(number); // Ensure positive value

while (tempNumber > 0) {

int digit = tempNumber % 10; // Extract the last digit

sum += digit; // Add digit to the sum

product \*= digit; // Multiply digit to the product

tempNumber /= 10; // Remove the last digit

}

System.out.println("Sum of digits: " + sum);

System.out.println("Product of digits: " + product);

}

}

1. It reads an integer input from the user.
2. It initializes variables for the sum and product of digits.
3. Using a while loop, it extracts each digit from the input number, adds it to the sum, and multiplies it to the product.
4. Finally, it prints the calculated sum and product.

2. In the Fibonacci series, each number (other than the first two) is the sum of the previous two  numbers. Here, the first two numbers are 0 and 1. So, the series looks like the below: ***0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ....***

Write a Java program that inputs an integer n and calculates the nth Fibonacci number. For  example,

*User input: 1, output: 0*

*User input: 4, output: 2*

*User input: 9, output: 21*

*User input: 21, output: 6765*

Answer:

import java.util.Scanner;

public class FibonacciCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a positive integer n: ");

int n = scanner.nextInt();

int fibN = calculateFibonacci(n);

System.out.println("The " + n + "th Fibonacci number is: " + fibN);

}

public static int calculateFibonacci(int n) {

if (n <= 1) {

return n;

}

int prev1 = 0;

int prev2 = 1;

int fibN = 0;

for (int i = 2; i <= n; i++) {

fibN = prev1 + prev2;

prev1 = prev2;

prev2 = fibN;

}

return fibN;

}

}

.

You can compile and run this program, and it will prompt you to enter a positive integer n. It will then calculate and display the nth Fibonacci number. For example:

* User input: 1, output: **0**
* User input: 4, output: **2**
* User input: 9, output: **21**
* User input: 21, output: **6765**

3. Write a Java program that takes an input of an integer number. Then, it checks whether this  number is divisible by 4, 6, and 10 or some or none. If multiple of them divide your number, you  must mention all of them. For example,

*user inputs: 24, output: 24 is divisible by 4 and 6*

*user inputs: 30, output: 30 is divisible by 6 and 10*

*user inputs: 60, output: 60 is divisible by 4,6, and 10*

*user inputs: 22, output: 22 is not divisible by 4,6, or 10*

answer:

import java.util.Scanner;

public class DivisibilityChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer number: ");

int number = scanner.nextInt();

scanner.close();

boolean divisibleBy4 = (number % 4 == 0);

boolean divisibleBy6 = (number % 6 == 0);

boolean divisibleBy10 = (number % 10 == 0);

if (divisibleBy4 && divisibleBy6 && divisibleBy10) {

System.out.println(number + " is divisible by 4, 6, and 10.");

} else if (divisibleBy4 && divisibleBy6) {

System.out.println(number + " is divisible by 4 and 6.");

} else if (divisibleBy6 && divisibleBy10) {

System.out.println(number + " is divisible by 6 and 10.");

} else if (divisibleBy4) {

System.out.println(number + " is divisible by 4.");

} else if (divisibleBy6) {

System.out.println(number + " is divisible by 6.");

} else if (divisibleBy10) {

System.out.println(number + " is divisible by 10.");

} else {

System.out.println(number + " is not divisible by 4, 6, or 10.");

}

}

}

4. Write a Java program that takes electricity consumption in KWH as input (from the user) and  calculates the total electricity bill according to the given condition:

*0 - 50 units: 10 cents/unit*

*51 - 150 units: 16 cents/unit*

*151 - 250 units: 22 cents/units*

*251 units or higher: 30 cents/unit*

If the total consumption is less than 251 KWH, an additional surcharge of $7 is added to the bill.  Else, the surcharge amount will be $12.

import java.util.Scanner;

public class ElectricityBillCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input: Electricity consumption in KWH

System.out.print("Enter electricity consumption in KWH: ");

double consumption = scanner.nextDouble();

// Constants for unit rates and surcharges

final double RATE\_0\_TO\_50 = 0.10;

final double RATE\_51\_TO\_150 = 0.16;

final double RATE\_151\_TO\_250 = 0.22;

final double RATE\_ABOVE\_250 = 0.30;

final double SURCHARGE\_LESS\_THAN\_251 = 7.00;

final double SURCHARGE\_251\_OR\_HIGHER = 12.00;

double totalBill = 0.0;

if (consumption >= 0 && consumption <= 50) {

totalBill = consumption \* RATE\_0\_TO\_50;

} else if (consumption <= 150) {

totalBill = 50 \* RATE\_0\_TO\_50 + (consumption - 50) \* RATE\_51\_TO\_150;

} else if (consumption <= 250) {

totalBill = 50 \* RATE\_0\_TO\_50 + 100 \* RATE\_51\_TO\_150 + (consumption - 150) \* RATE\_151\_TO\_250;

} else {

totalBill = 50 \* RATE\_0\_TO\_50 + 100 \* RATE\_51\_TO\_150 + 100 \* RATE\_151\_TO\_250 + (consumption - 250) \* RATE\_ABOVE\_250;

}

// Apply surcharge

if (consumption < 251) {

totalBill += SURCHARGE\_LESS\_THAN\_251;

} else {

totalBill += SURCHARGE\_251\_OR\_HIGHER;

}

System.out.println("Total electricity bill: $" + totalBill);

}

}

5.The dates for each season in the northern hemisphere are:

*Spring: March 20 - June 20*

*Summer: June 21 - September 21*

*Autumn: September 22 - December 20*

*Winter: December 21 - March 19*

Write a Java program that inputs a date and outputs the date’s season in the northern  hemisphere. The input is a string representing the month and an int representing the day. First,  you must check if the string and int are valid (an actual month and day). For example, **July 23** is  valid, while **March 53** or **Covid 19** is invalid.

import java.util.Scanner;

public class SeasonIdentifier {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input month as a string (e.g., "March")

System.out.print("Enter the month (e.g., March): ");

String month = scanner.nextLine().toLowerCase();

// Input day as an integer (e.g., 23)

System.out.print("Enter the day (e.g., 23): ");

int day = scanner.nextInt();

// Validate the input month and day

if (isValidMonth(month) && isValidDay(day)) {

String season = getSeason(month, day);

System.out.println("The season for " + month + " " + day + " is " + season);

} else {

System.out.println("Invalid input. Please enter a valid month and day.");

}

}

// Check if the input month is valid

private static boolean isValidMonth(String month) {

return month.equals("january") || month.equals("february") || month.equals("march") ||

month.equals("april") || month.equals("may") || month.equals("june") ||

month.equals("july") || month.equals("august") || month.equals("september") ||

month.equals("october") || month.equals("november") || month.equals("december");

}

// Check if the input day is valid

private static boolean isValidDay(int day) {

return day >= 1 && day <= 31;

}

// Determine the season based on the input month and day

private static String getSeason(String month, int day) {

if ((month.equals("march") && day >= 20) || (month.equals("april") || month.equals("may")) ||

(month.equals("june") && day <= 20)) {

return "Spring";

} else if ((month.equals("june") && day >= 21) || (month.equals("july") || month.equals("august")) ||

(month.equals("september") && day <= 21)) {

return "Summer";

} else if ((month.equals("september") && day >= 22) || (month.equals("october") || month.equals("november")) ||

(month.equals("december") && day <= 20)) {

return "Autumn";

} else {

return "Winter";

}

}

}