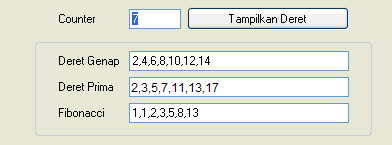
|  |
| --- |
| **Soal Test - Departemen IT** |

|  |  |
| --- | --- |
| **Nama** | Hanas Bayu Pratama |
| **Hari & Tanggal** | Selasa, 27 Juni 2023 |

1. Buatlah program untuk menampilkan deret sesuai form di bawah ini:



Ketika tombol ‘Tampilkan Deret’ di klik, masing-masing editbox di bagian bawah harus menampilkan deret yang sesuai dengan yang diminta sebanyak jumlah counter yang di entry pada editbox di bagian atasnya. Pada contoh di atas masing-masing ditampilkan 7 bilangan Genap, Prima, dan Fibonacci.

package trans7;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author hanas

\*/

public class Palindrome {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

int counter = 8;

List<Integer> deretGenap = generateDeretGenap(counter);

List<Integer> deretPrima = generateDeretPrima(counter);

List<Integer> deretFibonacci = generateDeretFibonacci(counter);

System.out.println("Deret Bilangan Genap:");

for (int num : deretGenap) {

System.out.print(num + " ");

}

System.out.println("\nDeret Bilangan Prima:");

for (int num : deretPrima) {

System.out.print(num + " ");

}

System.out.println("\nDeret Fibonacci:");

for (int num : deretFibonacci) {

System.out.print(num + " ");

}

}

private static List<Integer> generateDeretGenap(int counter) {

List<Integer> deret = new ArrayList<>();

int num = 2;

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

deret.add(num);

num += 2;

}

return deret;

}

private static List<Integer> generateDeretPrima(int counter) {

List<Integer> deret = new ArrayList<>();

int num = 2;

while (deret.size() < counter) {

if (isPrima(num)) {

deret.add(num);

}

num++;

}

return deret;

}

private static List<Integer> generateDeretFibonacci(int counter) {

List<Integer> deret = new ArrayList<>();

int num1 = 0;

int num2 = 1;

deret.add(num1);

deret.add(num2);

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

int nextNum = num1 + num2;

deret.add(nextNum);

num1 = num2;

num2 = nextNum;

}

return deret;

}

private static boolean isPrima(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

}

1. Buatlah program untuk mengecek input String apakah termasuk Palindrome atau tidak, serta menghitung jumlah huruf konsonan dan vokal.

Contoh kalimat palindrome:

* 1. KATAK
  2. UBI IBU

Ilustrasi 1:

INPUT : UBI IBU

HASIL:

Kalimat “UBI IBU” termasuk palindrome

Jumlah Konsonan: 2

Jumlah vocal : 4

Ilustrasi 2:

INPUT : MUKA KAMU

HASIL:

Kalimat “MUKA KAMU” tidak termasuk palindrome

Jumlah konsonan: 4

Jumlah vocal : 4

Tampilan program bebas, hanya ada Kolom Input, Tombol Proses, dan Hasil.

NB: tidak CASE SENSITIVE

import java.util.Scanner;

public class Program {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Masukkan kalimat: ");

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

boolean isPalindrome = isPalindrome(input);

int consonantCount = countConsonants(input);

int vowelCount = countVowels(input);

System.out.println("Hasil:");

System.out.println("Kalimat \"" + input + "\" " + (isPalindrome ? "termasuk palindrome" : "tidak termasuk palindrome"));

System.out.println("Jumlah Konsonan: " + consonantCount);

System.out.println("Jumlah Vokal: " + vowelCount);

}

public static boolean isPalindrome(String input) {

StringBuilder reversed = new StringBuilder();

for (int i = input.length() - 1; i >= 0; i--) {

char ch = input.charAt(i);

if (Character.isLetter(ch)) {

reversed.append(ch);

}

}

return input.equals(reversed.toString());

}

public static int countConsonants(String input) {

int count = 0;

for (int i = 0; i < input.length(); i++) {

char ch = input.charAt(i);

if (Character.isLetter(ch) && !isVowel(ch)) {

count++;

}

}

return count;

}

public static int countVowels(String input) {

int count = 0;

for (int i = 0; i < input.length(); i++) {

char ch = input.charAt(i);

if (Character.isLetter(ch) && isVowel(ch)) {

count++;

}

}

return count;

}

public static boolean isVowel(char ch) {

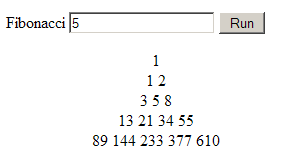
ch = Character.toLowerCase(ch);

return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';

}

}

1. Buatlah program untuk menampilkan deret Bilangan Fibonacci sesuai form di bawah ini:



Ketika tombol Run di klik, hasil di bagian bawah harus menampilkan deret fibbonaci dengan pola seperti diatas, dalam contoh diminta 5 baris deret fibbonaci, dengan ketentuan baris 1 jumlah deret 1 angka, baris 2 jumlah deret 2 angka, baris 3 jumlah deret 3 angka, dst.

public class Fibonacci {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Masukkan jumlah baris deret Fibonacci: ");

int rowCount = scanner.nextInt();

int[] fibonacciSeries = generateFibonacciSeries(rowCount);

System.out.println("Hasil:");

int startIndex = 0;

for (int row = 1; row <= rowCount; row++) {

for (int i = startIndex; i < startIndex + row; i++) {

System.out.print(fibonacciSeries[i] + " ");

}

System.out.println();

startIndex += row;

}

}

public static int[] generateFibonacciSeries(int n) {

int[] series = new int[n];

series[0] = 0;

series[1] = 1;

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

series[i] = series[i - 1] + series[i - 2];

}

return series;

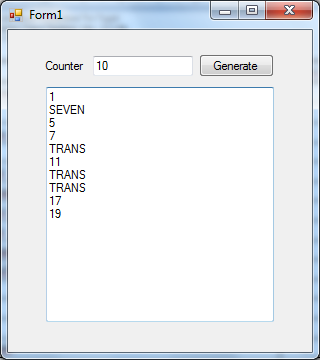
}

}

1. Tampilkan deret bilangan ganjil sebanyak n kali dengan kriteria:

- setiap angka yang habis dibagi 3 **atau** mengandung bilangan 3, diganti menjadi kata 'TRANS'.

- setiap angka yang habis dibagi 3 **dan** mengandung bilangan 3, diganti menjadi kata 'SEVEN'.



public class Deret {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Masukkan jumlah deret: ");

int n = scanner.nextInt();

printOddSeries(n);

}

public static void printOddSeries(int n) {

int count = 0;

int number = 1;

while (count < n) {

if (isOdd(number)) {

String result = transformNumber(number);

System.out.print(result + " ");

count++;

}

number++;

}

}

public static boolean isOdd(int number) {

return number % 2 != 0;

}

public static String transformNumber(int number) {

if (number % 3 == 0 || String.valueOf(number).contains("3")) {

if (number % 3 == 0 && String.valueOf(number).contains("3")) {

return "SEVEN";

} else {

return "TRANS";

}

} else {

return String.valueOf(number);

}

}

}

1. Buatlah program untuk membuat encrypter dan decrpyter, dengan ketentuan sbb:

* Hanya huruf vokal yang di encrypt dan decrypt
* Urutan vokal sebagai berikut a,i,u,e,o dengan index 0,1,2,3,4
* Sistem encrypt menggunakan shifting modulus angka prima, sebagai contoh: “reog”,

huruf **e** dengan index 3 + prima pertama 2 = 5 modulus 5 = 0 => a

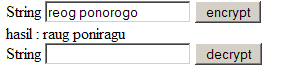
huruf **o** dengan index 4 + prima kedua 3 = 7 modulus 5 = 2 => u

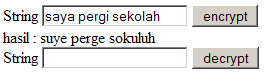
hasil encrypt menjadi “raug”

* Sistem decrypt menggunakan cara kebalikan dari sistem encrypt, jika dimasukkan “raug” maka hasilnya harus “reog”

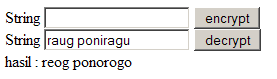
Cth tampilan:

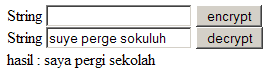
**Encrypt**





**Decrypt**





import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class VowelEncrypterDecrypter {

private static final char[] VOWELS = {'a', 'i', 'u', 'e', 'o'};

private static final int[] PRIMES = {2, 3, 5, 7, 11};

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Masukkan teks: ");

String text = scanner.nextLine();

String encryptedText = encrypt(text);

System.out.println("Hasil Enkripsi: " + encryptedText);

String decryptedText = decrypt(encryptedText);

System.out.println("Hasil Dekripsi: " + decryptedText);

}

public static String encrypt(String text) {

StringBuilder encryptedText = new StringBuilder();

for (char ch : text.toCharArray()) {

if (isVowel(ch)) {

int vowelIndex = getVowelIndex(ch);

int primeIndex = vowelIndex % PRIMES.length;

int shiftedIndex = (vowelIndex + PRIMES[primeIndex]) % VOWELS.length;

encryptedText.append(VOWELS[shiftedIndex]);

} else {

encryptedText.append(ch);

}

}

return encryptedText.toString();

}

public static String decrypt(String text) {

StringBuilder decryptedText = new StringBuilder();

for (char ch : text.toCharArray()) {

if (isVowel(ch)) {

int vowelIndex = getVowelIndex(ch);

int primeIndex = vowelIndex % PRIMES.length;

int shiftedIndex = (vowelIndex - PRIMES[primeIndex] + VOWELS.length) % VOWELS.length;

decryptedText.append(VOWELS[shiftedIndex]);

} else {

decryptedText.append(ch);

}

}

return decryptedText.toString();

}

public static boolean isVowel(char ch) {

ch = Character.toLowerCase(ch);

return ch == 'a' || ch == 'i' || ch == 'u' || ch == 'e' || ch == 'o';

}

public static int getVowelIndex(char ch) {

ch = Character.toLowerCase(ch);

for (int i = 0; i < VOWELS.length; i++) {

if (ch == VOWELS[i]) {

return i;

}

}

return -1;

}

}