# Ejercicios prácticos

|  |
| --- |
| **Asignatura**: Programación |
| **Nombre**: Actividad práctica 1 |
| **Unidades**: 1, 2, 3, 4 y 5 |

## Guía

* Se plantean diferentes problemas que tratan aspectos de las unidades 1, 2, 3, 4 y 5.
* Para aquellos ejercicios que se pida el pseudocódigo se debe poner el texto dentro del recuadro y para los diagramas poner una imagen o captura de pantalla.
* Los diagramas de flujo se pueden realizar con la aplicación: <https://www.drawio.com/>
* Para aquellos ejercicios que se pida código fuente se debe poner todo el código de cada clase en formato texto dentro del recuadro.
* Los ejercicios pueden tener más de una solución, buscar la óptima, teniendo en cuenta el número de instrucciones, sentencias y estructuras, etc.
* Es recomendable utilizar de apoyo la documentación oficial de java de Oracle. <https://docs.oracle.com/en/java/>

## Ejemplo

Realiza una solución que cumpla lo siguiente: solicite el nombre de una persona y lo salude.

**Pseudocódigo**

|  |
| --- |
| Inicio  Cadena nombre  Imprimir (“Introduce un nombre”)  Leer (nombre)  Imprimir (“Hola “ + nombre)  Fin |

**Diagrama**

Diagrama

Descripción generada automáticamente

**Código fuente**

|  |
| --- |
| **import** java.util.Scanner;  **public** **class** Ejemplo {  **public** **static** **void** main(String[] args) {  String nombre;  System.***out***.println("Introduce un nombre:");  Scanner entrada = **new** Scanner(System.***in***);  nombre = entrada.next();  System.***out***.println("Hola " + nombre);  }  } |

## Enunciado de los ejercicios

## Ejercicio 1.

Realiza el pseudocódigo, diagrama y código java de un programa que pida un número por teclado e indique si es un número primo o no.

Un número primo es aquel solo puede dividirse entre 1 y sí mismo. Por ejemplo: 25 no es primo, ya que 25 es divisible entre 5, sin embargo, 17 si es primo. Un buen truco, calcular la raíz cuadrada del número e ir comprobando que si es divisible desde ese número hasta 1.

NOTA: Si se introduce un número menor o igual que 1, directamente es no primo.

**Pseudocódigo**

|  |
| --- |
| Inicio  Leer numero  Si numero <= 1  Escribir "El número no es primo"  Sino  Establecer esPrimo = verdadero  Para i = 2 hasta raíz cuadrada de numero  Si numero mod i == 0  Establecer esPrimo = falso  Salir del bucle  Fin Si  Fin Para  Si esPrimo  Escribir "El número es primo"  Sino  Escribir "El número no es primo"  Fin Si  Fin Si  Fin |

**Diagrama**

|  |
| --- |
|  |

**Código fuente**

|  |
| --- |
| **import java.util.Scanner;**  **public class Ejercicio1 {**  **public static void main(String[] args) {**  **Scanner scanner = new Scanner(System.in);**  **System.out.print("Ingrese un número: ");**  **int numero = scanner.nextInt();**  **if (numero <= 1) {**  **System.out.println("El número no es primo.");**  **} else {**  **boolean esPrimo = true;**  **for (int i = 2; i <= Math.sqrt(numero); i++) {**  **if (numero % i == 0) {**  **esPrimo = false;**  **break;**  **}**  **}**  **if (esPrimo) {**  **System.out.println("El número es primo.");**  **} else {**  **System.out.println("El número no es primo.");**  **}**  **}**  **scanner.close();**  **}**  **}** |

## Ejercicio 2.

Escribe un programa que simule un pequeño menú interactivo para realizar operaciones de suma. El programa debe seguir las siguientes reglas:

* El menú debe tener cuatro opciones:
  + Opción 1: Introducir el primer número.
  + Opción 2: Introducir el segundo número.
  + Opción 3: Mostrar la suma de los números introducidos en las opciones 1 y 2.
  + Opción 4: Salir
* El programa debe iniciar mostrando el menú y permitir al usuario elegir una de las opciones.
* El usuario solo puede seleccionar la Opción 2 después de haber seleccionado la Opción 1, y solo puede seleccionar la Opción 3 después de haber seleccionado tanto la Opción 1 como la Opción 2.
* Una vez que el usuario ha introducido los dos números (Opciones 1 y 2), el programa debe calcular la suma y mostrar el resultado cuando se elige la Opción 3.
* Cuando se muestre el resultado desde la Opción 3, se tiene que volver a reiniciar los pasos y empezar desde la Opción 1.
* El programa debe continuar mostrando el menú hasta que el usuario decida salir, eligiendo la Opción 4.

Asegúrate de que el programa pueda manejar posibles errores, como que se introduzca una opción no válida o la introducción de caracteres no numéricos al ingresar los números. Además, que cumpla con las reglas descritas.

|  |
| --- |
| import java.util.Scanner;  public class Ejercicio2 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  int opcion = 0;  int numero1 = 0;  int numero2 = 0;  while (opcion != 4) {  System.out.println("Menú interactivo");  System.out.println("1. Introducir el primer número");  System.out.println("2. Introducir el segundo número");  System.out.println("3. Mostrar la suma de los números introducidos");  System.out.println("4. Salir");  System.out.print("Seleccione una opción: ");  opcion = scanner.nextInt();  switch (opcion) {  case 1:  System.out.print("Introduzca el primer número: ");  numero1 = scanner.nextInt();  break;  case 2:  if (numero1 == 0) {  System.out.println("Debe introducir el primer número primero.");  } else {  System.out.print("Introduzca el segundo número: ");  numero2 = scanner.nextInt();  }  break;  case 3:  if (numero1 == 0 || numero2 == 0) {  System.out.println("Debe introducir ambos números primero.");  } else {  int suma = numero1 + numero2;  System.out.println("La suma de los números es: " + suma);  numero1 = 0;  numero2 = 0;  }  break;  case 4:  System.out.println("Saliendo del programa...");  break;  default:  System.out.println("Opción no válida. Intente nuevamente.");  break;  }  }  scanner.close();  }  } |

## Ejercicio 3.

El siguiente código tiene errores de compilación y de ejecución.

**public** **class** Ejercicio3 {

**public** **static** **void** main(String[] args) {

**int** numero = 5;

// Se realiza la suma de cada número hasta llegar al número de la variable

// por ejemplo, de 3 sería 1 + 2 + 3 = 6

System.***out***.println("Sumamos cada número hasta llegar al " + numero);

**int** temp = 0; resultado = 0;

**while**(temp <= numero) {

resultado += temp;

temp = temp - temp++;

}

System.***out***.println("El resultado es: " + resultado);

// Se hace la cuenta atras desde el número que hay en la variable número

System.***out***.println("\nMostramos la cuenta atras desde: " + numero);

**int** contador = numero;

**do** {

System.***out***.println(contador);

contador--;

} **while**(contador<0);

// Se muestra por cada número si es par o impart hasta llegar al número que hay en la variable número

System.***out***.println("\nIndicamos los números que son pares e impares hasta el número " + numero);

**for** (**int** i = ++contador; i <= numero; temp++) {

**if**(i%2 == 1) {

System.***out***.println("El número " + i + " es par");

} **else** {

System.***out***.println("El número " + i + " es impar");

}

}

}

}

Corrige los errores, pon el código fuente en el recuadro y rellena la tabla indicando en que número de línea ocurre y que corrección se ha realizado.

NOTA: Los errores se pueden ver a simple vista al analizar el código y con la ayuda del IDE, aun así una gran ayuda puede ser el modo depuración y los puntos de interrupción.

|  |  |  |
| --- | --- | --- |
| **Error** | **Número de línea** | **Corrección** |
| 10 int temp = 0; resultado = 0; int temp = 0, resultado = 0; | 10 int temp = 0; resultado = 0; int temp = 0, resultado = 0; | 10 int temp = 0; resultado = 0; int temp = 0, resultado = 0; |
| 13 temp = temp - temp++; temp++; | 13 temp = temp - temp++; temp++; | 13 temp = temp - temp++; temp++; |
| 20 while(contador<0); while(contador > 0); | 20 while(contador<0); while(contador > 0); | 20 while(contador<0); while(contador > 0); |
| 24 for (int i = ++contador; i <= numero; temp++) { for (int i = 1; i <= numero; i++) { | 24 for (int i = ++contador; i <= numero; temp++) { for (int i = 1; i <= numero; i++) { | 24 for (int i = ++contador; i <= numero; temp++) { for (int i = 1; i <= numero; i++) { |
| 25 if(i%2 == 1) { if(i % 2 == 0) { | 25 if(i%2 == 1) { if(i % 2 == 0) { | 25 if(i%2 == 1) { if(i % 2 == 0) { |

## Ejercicio 4.

Realiza un programa para validar un DNI (Documento Nacional de Identidad). El programa debe ser capaz de pedir por teclado primero los números que confirman el DNI y después la letra que le corresponde. Finalmente, el programa mostrará si es correcto no.

Hay que tener en cuenta que un DNI cumple las siguientes características:

* Está formado por un total de 8 dígitos y una letra.
* Los dígitos pueden tomar el valor de 0 a 9
* La letra puede ser T, R, W, A, G, M, Y, F, P, D, X, B, N, J, Z, S, Q, V, H, L, C, K, E.
* Para saber que letra le corresponde al DNI se realiza una división de la parte numérica por 23 y con el resto se determina la letra correspondiente. La siguiente tabla muestra el valor del resto y la letra correspondiente.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RESTO** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| **LETRA** | T | R | W | A | G | M | Y | F | P | D | X | B | N | J | Z | S | Q | V | H | L | C | K | E |

|  |
| --- |
| package ejercicio4.DNI;  import java.util.Scanner;  public class ValidadorDNI {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  // Pedir por teclado los números del DNI  System.out.print("Ingresa los 8 dígitos del DNI: ");  int numerosDNI = scanner.nextInt();  // Pedir por teclado la letra del DNI  System.out.print("Ingresa la letra del DNI: ");  char letraDNI = scanner.next().toUpperCase().charAt(0);  // Validar el DNI  if (validarDNI(numerosDNI, letraDNI)) {  System.out.println("El DNI ingresado es válido.");  } else {  System.out.println("El DNI ingresado no es válido.");  }  scanner.close();  }  // Función para validar el DNI  private static boolean validarDNI(int numerosDNI, char letraDNI) {  // Array con las letras posibles  char[] letrasPosibles = { 'T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q',  'V', 'H', 'L', 'C', 'K', 'E' };  // Calcular el resto de la división de los números del DNI por 23  int resto = numerosDNI % 23;  // Verificar si la letra ingresada coincide con la calculada  return letraDNI == letrasPosibles[resto];  }  } |

## Ejercicio 5.

Busca y explica con algún ejemplo para que sirven las clases *Date* y *SimpleDateFormat.* Luegorealiza un programa que pida una fecha de nacimiento en el formato “dd/mm/yyyy” y muestre la edad que tiene.

|  |
| --- |
| import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Date;  import java.util.Scanner;  public class Ejercicio5 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  // Pedimos por teclado la fecha de nacimiento  System.out.print("Ingrese la fecha de nacimiento (dd/MM/yyyy): ");  String fechaNacimientoStr = scanner.next();  // Crear un objeto SimpleDateFormat con el patrón "dd/MM/yyyy"  SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");  try {  // Parseamos la cadena de fecha a un objeto Date  Date fechaNacimiento = formatoFecha.parse(fechaNacimientoStr);  // Calculamos la edad  int edad = calcularEdad(fechaNacimiento);  // Mostramos la edad  System.out.println("La edad es: " + edad + " años.");  } catch (ParseException e) {  System.out.println("Formato de fecha incorrecto. Por favor, ingrese la fecha en el formato especificado.");  }  scanner.close();  }  // Función para calcular la edad a partir de la fecha de nacimiento  private static int calcularEdad(Date fechaNacimiento) {  // Obtener la fecha actual  Date fechaActual = new Date();  // Calculamos la diferencia en milisegundos  long diferenciaEnMillis = fechaActual.getTime() - fechaNacimiento.getTime();  // Calculamos la edad en años  int edadEnAnios = (int) (diferenciaEnMillis / (1000 \* 60 \* 60 \* 24 \* 365.25));  return edadEnAnios;  }  } |