



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

Guía de Ejercicios – Punteros

Algoritmos y Estructura de datos

Curso Ing. Gustavo Schmidt

Nivel 1: Conceptos Básicos

1. **Hola Mundo:** Escribe un programa en Java que imprima "Hola Mundo" en la consola.
2. **Variables y Tipos de Datos:** Declara variables de diferentes tipos (int, double, boolean, String) e imprime sus valores.
3. **Operaciones Aritméticas:** Escribe un programa que tome dos números enteros como entrada y muestre la suma, resta, multiplicación, división y el módulo de esos números.
4. **Entrada del Usuario:** Crea un programa que solicite al usuario su nombre y edad, y luego imprima un mensaje personalizado, por ejemplo, "Hola, [nombre]. Tienes [edad] años."
5. **Condicionales Simples:** Escribe un programa que solicite al usuario un número entero y verifique si es positivo, negativo o cero.

Nivel 2: Estructuras de Control

6. **Condicionales Múltiples:** Escribe un programa que pida al usuario una calificación (número entero) y muestre si la calificación es "Excelente" (90-100), "Bueno" (80-89), "Regular" (70-79) o "Insuficiente" (0-69).
7. **Ciclo for:** Crea un programa que imprima los números del 1 al 10 utilizando un ciclo for.
8. **Ciclo while:** Escribe un programa que solicite al usuario un número positivo y calcule la suma de todos los números naturales hasta ese número (inclusive) utilizando un ciclo while.
9. **Ciclo do-while:** Escribe un programa que siga solicitando al usuario un número hasta que ingrese un número negativo.
10. **Cálculo de Factorial:** Escribe un programa que calcule el factorial de un número dado por el usuario.

Nivel 3: Arrays y Métodos

11. **Array de Números:** Crea un programa que solicite 5 números al usuario, los almacene en un array y luego imprima esos números en orden inverso.
12. **Promedio de un Array:** Escribe un programa que tome un array de números enteros y calcule el promedio de esos números.
13. **Métodos en Java:** Crea un método que reciba dos números y devuelva su suma. Utiliza este método en tu programa principal.
14. **Método para Invertir una Cadena:** Escribe un método que tome una

cadena como parámetro y devuelva la cadena invertida.

15. **Búsqueda en un Array:** Crea un programa que solicite al usuario un número y luego verifique si ese número está presente en un array de 10 números.

Nivel 4: Clases y Objetos

16. **Clase Persona:** Define una clase Persona con atributos nombre, edad, y dni. Implementa métodos para mostrar los detalles de la persona y para verificar si es mayor de edad.
17. **Clase Cuenta Bancaria:** Crea una clase CuentaBancaria con atributos saldo y titular. Implementa métodos para depositar y retirar dinero, asegurándote de que no se pueda retirar más dinero del que hay en la cuenta.
18. **Clase Rectángulo:** Crea una clase Rectangulo con atributos base y altura. Implementa métodos para calcular el área y el perímetro del rectángulo.
19. **Clase Círculo:** Define una clase Circulo con un atributo radio. Implementa un método para calcular el área del círculo y otro para calcular el perímetro. Considera la constante PI como Math.PI.

Uso de String

1. **Concatenación de Cadenas:** Escribe un programa que tome dos cadenas de texto del usuario y las concatene en una sola, luego imprima el resultado.
2. **Longitud de una Cadena:** Crea un programa que solicite al usuario una cadena y muestre su longitud.
3. **Comparación de Cadenas:** Escribe un programa que compare dos cadenas ingresadas por el usuario, verificando si son iguales o diferentes (considerando mayúsculas y minúsculas).
4. **Extracción de Subcadena:** Crea un programa que tome una cadena y dos números enteros como entrada, y extraiga la subcadena correspondiente a esos índices.
5. **Reemplazo de Caracteres:** Escribe un programa que reemplace todas las ocurrencias de un carácter en una cadena por otro carácter ingresado por el usuario.

Uso de LocalDate

6. **Fecha Actual:** Escribe un programa que muestre la fecha actual utilizando la clase LocalDate.
7. **Diferencia entre Fechas:** Crea un programa que solicite dos fechas al

usuario (en formato yyyy-mm-dd) y calcule la diferencia en días entre ambas fechas.

8. **Agregar Días a una Fecha:** Escribe un programa que tome una fecha y un número de días como entrada, y muestre la fecha resultante después de agregar esos días.
9. **Validar Fecha:** Crea un programa que solicite una fecha al usuario y verifique si es una fecha válida, es decir, si existe en el calendario.
10. **Día de la Semana:** Escribe un programa que solicite una fecha y muestre qué día de la semana corresponde a esa fecha.

Manejo de Archivos

11. **Crear y Escribir en un Archivo:** Escribe un programa que cree un archivo de texto y escriba en él una serie de líneas de texto proporcionadas por el usuario.
12. **Leer desde un Archivo:** Crea un programa que lea y muestre en la consola el contenido de un archivo de texto existente.
13. **Contar Palabras en un Archivo:** Escribe un programa que lea un archivo de texto y cuente cuántas palabras contiene.
14. **Buscar Palabra en un Archivo:** Crea un programa que busque una palabra específica en un archivo de texto y muestre cuántas veces aparece.
15. **Copiar Contenido de un Archivo a Otro:** Escribe un programa que copie el contenido de un archivo de texto en otro archivo.

Combinación de Clases

16. **Formato de Fechas en Archivos:** Crea un programa que lea un archivo de texto que contiene fechas en formato yyyy-mm-dd, convierta cada fecha a un formato más legible (por ejemplo, dd/MM/yyyy), y guarde el resultado en otro archivo.
17. **Procesar Cadenas de Fechas:** Escribe un programa que lea un archivo donde cada línea contiene una fecha en formato yyyy-mm-dd y un texto. Luego, el programa debe leer las fechas, calcular cuántos días han pasado desde cada una, y escribir el resultado en un nuevo archivo.
18. **Buscar y Reemplazar en Archivos:** Crea un programa que busque una palabra específica en un archivo de texto y la reemplace por otra palabra, guardando el resultado en un nuevo archivo.
19. **Analizar Fechas en Texto:** Escribe un programa que lea un archivo de texto, identifique todas las cadenas que se asemejan a fechas en el formato yyyy-mm-dd, y las convierta en objetos LocalDate para luego calcular cuántos días han pasado desde cada fecha hasta hoy.
20. **Generar un Archivo de Resumen:** Crea un programa que lea varios

archivos de texto (que contengan fechas, nombres y mensajes) y genere un resumen en un nuevo archivo, donde se listan las fechas, los nombres de las personas, y el mensaje más reciente de cada una.