

# Ejercicios y Repaso

## Problemas

Elizabeth León Guzmán, Ph.D.  
eleonguz@unal.edu.co

Jonatan Gómez Perdomo, Ph. D.  
jgomezpe@unal.edu.co

Arles Rodríguez, Ph.D.  
aerodriguezp@unal.edu.co

Camilo Cubides, Ph.D. (c)  
eccubidesg@unal.edu.co

Carlos Andres Sierra, M.Sc.  
casierrav@unal.edu.co

Research Group on Data Mining – Grupo de Investigación en Minería de Datos – (Midas)  
Research Group on Artificial Life – Grupo de Investigación en Vida Artificial – (Alife)  
Computer and System Department  
Engineering School  
Universidad Nacional de Colombia

# Agenda

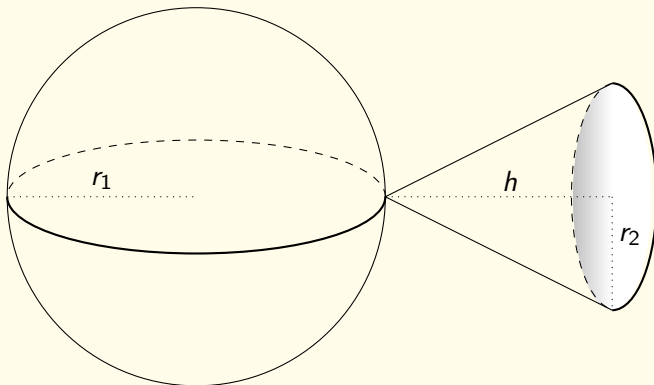
- 1 Funciones
- 2 Condicionales
- 3 Estructuras Cíclicas
- 4 Estructuras Ciclicas II
- 5 Cadenas
- 6 Arreglos y Matrices
- 7 Clases y Objetos
- 8 Herencia y Polimorfismo
- 9 Relaciones Objetos-Clases



# Volumen de un sólido I

## Problema

Para el sólido que se presenta a continuación,



# Volumen de un sólido II

## Problema (continuación)

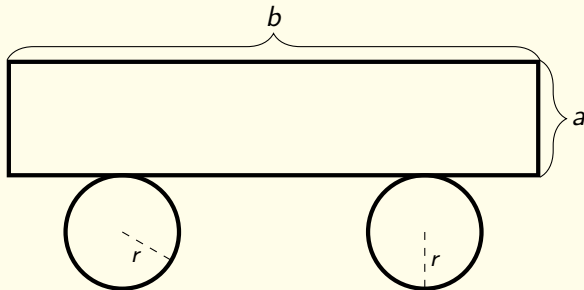
- 1 Establezca el modelo matemático (función matemática) que permita calcular el volumen del sólido anteriormente mostrado.
- 2 Escriba una función en Java que implemente la función anteriormente modelada, en la cual se invoque la constante matemática  $\pi$  de la clase `Math`
- 3 Para los valores  $r_1 = 3$ ,  $h = 9/2$  y  $r_2 = 4$ , calcule (a mano o con calculadora) el volumen del sólido y compárelo con el resultado obtenido a partir de la evaluación de la función anteriormente implementada. ¿Qué pasa si se invoca la función con los mismos valores, pero  $h$  se calcula como la expresión  $h = 9/2$ ?



# Área lateral de un vagón I

## Problema

Para el vagón que se presenta a continuación,



# Área lateral de un vagón II

## Problema (continuación)

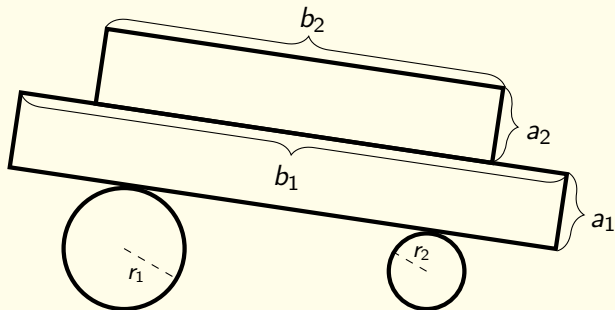
- 1 Establezca el modelo matemático (función matemática) que permita calcular el área lateral del vagón.
- 2 Escriba una función en Java que implemente la función anteriormente modelada, en la cual se invoque la constante matemática  $\pi$  de la clase Math.



# Área lateral de un carro l

## Problema

Para el carro que se presenta a continuación,



# Área lateral de un carro II

## Problema (continuación)

- 1 Establezca el modelo matemático (función matemática) que permita calcular el área lateral del carro.
- 2 Escriba una función en Java que implemente la función matemática previamente modelada, en la cual se utilice la composición de las funciones de suma de números reales, *area\_circulo* y *area\_rectangulo* codificadas previamente.





# Problemas varios

## Problemas

- ① Diseñe una función que calcule la cantidad de carne de aves en kilos si se tienen  $N$  gallinas,  $M$  gallos y  $K$  pollitos cada uno pesando 6 kilos, 7 kilos y 1 kilo respectivamente.
- ② Mi mamá me manda a comprar  $P$  panes a \$ 300 cada uno,  $M$  bolsas de leche a \$ 3300 cada una y  $H$  huevos a \$ 350 cada uno. Hacer un programa que me diga las vueltas (o lo que quedo debiendo) cuando me da un billete de  $B$  pesos.
- ③ Si pido prestados  $P$  cantidad de pesos para pagarlos en dos meses, si el interés del préstamo es del 3%. ¿Cuánto se debe pagar al final del segundo mes si el interés es compuesto mensualmente?
- ④ El número de contagiados de Covid-19 en el país de NuncaLandia se duplica cada día. Hacer un programa que diga el número total de personas que se han contagiado cuando pasen  $D$  días a partir de hoy, si el número de contagiados actuales es  $C$ .

# Agenda

- 1 Funciones
- 2 **Condicionales**
- 3 Estructuras Cíclicas
- 4 Estructuras Ciclicas II
- 5 Cadenas
- 6 Arreglos y Matrices
- 7 Clases y Objetos
- 8 Herencia y Polimorfismo
- 9 Relaciones Objetos-Clases



# Problemas varios I

## Problemas

- 1 Dado un número entero, determinar si ese número corresponde al código ASCII de una vocal minúscula.
- 2 Dado un carácter, determine si el código ASCII es par o no.
- 3 Dado un carácter, construya un programa en Java para determinar si el carácter es un dígito o no.
- 4 Dado un número real  $x$ , construya una función que permita determinar si el número es positivo, negativo o cero. Para cada caso de debe imprimir el texto que se especifica a continuación:

**Positivo:** "El número  $x$  es positivo"

**Negativo:** "El número  $x$  es negativo"

**Cero (0):** "El número  $x$  es el neutro para la suma"



# Problemas varios II

## Problemas (continuación)

- 5 Dado el centro y el radio de un círculo, determinar si un punto de  $\mathbb{R}^2$  pertenece o no al interior del círculo.
- 6 Dado un punto de  $\mathbb{R}^2$  retorne el cuadrante en forma de cadena ("I", "II", "III" o "IV") al cual pertenece el punto, en caso de que la abscisa o la ordena sean nulas, retorne la cadena "Ejes".
- 7 Dadas tres longitudes positivas, determinar si con esas longitudes se puede construir un triángulo.



# Agenda

- 1 Funciones
- 2 Condicionales
- 3 Estructuras Cíclicas**
- 4 Estructuras Cíclicas II
- 5 Cadenas
- 6 Arreglos y Matrices
- 7 Clases y Objetos
- 8 Herencia y Polimorfismo
- 9 Relaciones Objetos-Clases



# Problemas varios I

## Problemas

- 1 Imprimir un listado con los números del 1 al 100 cada uno con su respectivo cuadrado.
- 2 Imprimir un listado con los números impares desde 1 hasta 999 y seguidamente otro listado con los números pares desde 2 hasta 1000.
- 3 Imprimir los números pares en forma descendente hasta 2 que son menores o iguales a un número natural  $n \geq 2$  dado.
- 4 En 2022 el país *A* tendrá una población de 25 millones de habitantes y el país *B* de 18.9 millones. Las tasas de crecimiento anual de la población serán de 2% y 3% respectivamente. Desarrollar un algoritmo para informar en qué año la población del país *B* superará a la de *A*.



# Problemas varios II

## Problemas

- 5 Diseñar una función que permita calcular el épsilon de la máquina. El épsilon de máquina es el número decimal más pequeño que sumado a 1 se puede representar de manera precisa en la máquina (que no es redondeado), es decir, retorna un valor diferente de 1, éste da una idea de la precisión o número de cifras reales que pueden ser almacenadas en la máquina. La idea es realizar un ciclo en el cual se realiza la operación  $1 + \epsilon$  para potencias de 2 desde  $\epsilon = 2^0$  y continuando con potencias decrecientes de 2 ( $\epsilon = 2^{-1}, \epsilon = 2^{-2}, \epsilon = 2^{-3}, \epsilon = 2^{-4}, \dots$ ) hasta obtener que el resultado de la suma  $1 + \epsilon$  no se altere.



# Agenda

- 1 Funciones
- 2 Condicionales
- 3 Estructuras Cíclicas
- 4 Estructuras Cíclicas II**
- 5 Cadenas
- 6 Arreglos y Matrices
- 7 Clases y Objetos
- 8 Herencia y Polimorfismo
- 9 Relaciones Objetos-Clases





# Problemas varios I

## Problemas

- 1 Imprimir un listado con los números del 1 al 100 cada uno con su respectivo cuadrado.
- 2 Imprimir un listado con los números impares desde 1 hasta 999 y seguidamente otro listado con los números pares desde 2 hasta 1000.
- 3 Imprimir los números pares en forma descendente hasta 2 que son menores o iguales a un número natural  $n \geq 2$  dado.
- 4 Imprimir los números de 1 hasta un número natural  $n$  dado, cada uno con su respectivo factorial.
- 5 Calcular el valor de 2 elevado a la potencia  $n$ .
- 6 Leer un número natural  $n$ , leer otro dato de tipo real  $x$  y calcular  $x^n$ .
- 7 Diseñe un programa que muestre las tablas de multiplicar del 1 al 9.



# Problemas varios II

## Problemas

- 8 Diseñar una función que permita calcular una aproximación de la función exponencial alrededor de 0 para cualquier valor  $x \in \mathbb{R}$ , utilizando los primeros  $n$  términos de la serie de Maclaurin

$$\exp(x, n) \approx \sum_{i=0}^n \frac{x^i}{i!}.$$

- 9 Diseñar una función que permita calcular una aproximación de la función seno alrededor de 0 para cualquier valor  $x \in \mathbb{R}$  ( $x$  dado en radianes), utilizando los primeros  $n$  términos de la serie de Maclaurin

$$\sin(x, n) \approx \sum_{i=0}^n \frac{(-1)^i x^{2i+1}}{(2i+1)!}.$$



# Problemas varios III

## Problemas

- 10 Diseñar una función que permita calcular una aproximación de la función coseno alrededor de 0 para cualquier valor  $x \in \mathbb{R}$  ( $x$  dado en radianes), utilizando los primeros  $n$  términos de la serie de Maclaurin

$$\cos(x, n) \approx \sum_{i=0}^n \frac{(-1)^i x^{2i}}{(2i)!}.$$

- 11 Diseñar una función que permita calcular una aproximación de la función logaritmo natural alrededor de 0 para cualquier valor  $x \in \mathbb{R}^+$ , utilizando los primeros  $n$  términos de la serie de Maclaurin

$$\ln(x, n) \approx \sum_{i=0}^n \frac{1}{2i+1} \left( \frac{x^2 - 1}{x^2 + 1} \right)^{2i+1}.$$



# Problemas varios IV

## Problemas

- 12 Diseñar una función que permita calcular una aproximación de la función arco tangente para cualquier valor  $x \in [-1, 1]$ , utilizando los primeros  $n$  términos de la serie de Maclaurin (al evaluar esta función el resultado que se obtiene está expresado en radianes)

$$\arctan(x, n) \approx \sum_{i=0}^n \frac{(-1)^i x^{2i+1}}{(2i+1)}.$$



# Agenda

- 1 Funciones
- 2 Condicionales
- 3 Estructuras Cíclicas
- 4 Estructuras Ciclicas II
- 5 Cadenas**
- 6 Arreglos y Matrices
- 7 Clases y Objetos
- 8 Herencia y Polimorfismo
- 9 Relaciones Objetos-Clases



# Problemas varios I

## Problemas

- 1 Elabore un programa que dada una letra cuente cuantas ocurrencias de esta letra hay.
- 2 Elabore un programa que dada una cadena diga si todos los símbolos de la cadena son letras.
- 3 Elabore un programa que dada una cadena cuente las consonantes en dicha cadena.



# Problemas varios II

## Problemas

- ④ Desarrollar un algoritmo que retorne un valor booleano que indique si dos cadenas son iguales, esto es, que tienen la misma longitud con los mismos símbolos en la mismas posiciones.
- ⑤ Desarrollar un algoritmo que permita concatenar dos cadenas (colocar la segunda inmediatamente después de la primera), hay que tener en cuenta que la cadena resultante tendrá un tamaño mayor que cualquiera de la cadenas operandos si son distintas de la cadena vacía (la cadena resultante debe quedar guardada en una variable aparte).
- ⑥ Desarrollar un algoritmo que invierta una cadena de caracteres (la cadena invertida debe quedar guardada en una variable aparte).



# Problemas varios III

## Problemas

- 4 Desarrollar un algoritmo que determine si una cadena de caracteres es palíndromo. Una cadena se dice palíndromo si al invertirla es igual a ella misma. Ejemplos:
- “ala” es palíndromo
  - “anita lava la tina” No es palíndromo, pues al invertirla con espacios no es exactamente igual a la original.
  - “los estudiantes de programación leyeron toda la guía” no es palíndromo.
  - “robas ese sabor” es palíndromo





# Agenda

- 1 Funciones
- 2 Condicionales
- 3 Estructuras Cíclicas
- 4 Estructuras Ciclicas II
- 5 Cadenas
- 6 Arreglos y Matrices**
- 7 Clases y Objetos
- 8 Herencia y Polimorfismo
- 9 Relaciones Objetos-Clases



# Problemas varios I

## Problemas

- 1 Modele mediante una función matemática y desarrolle un algoritmo en Java con ciclos que permita hallar la posición del mínimo de un vector de números reales.
- 2 Modele mediante una función matemática y desarrolle un algoritmo con ciclos en Java que calcule el mínimo de un vector de números reales.



# Problemas varios II

## Problemas

- 4 Modele mediante una función matemática y desarrolle un algoritmo en Java que calcule el producto por escalar de una constante con un vector de números reales. Sea  $\alpha \in \mathbb{R}$  y  $v = (v_1, v_2, \dots, v_n)$ , el producto por escalar de  $\alpha$  con  $v$  (notado  $\alpha v$ ) es el vector dado por la expresión

$$\alpha v = (\alpha v_1, \alpha v_2, \dots, \alpha v_n)$$



# Agenda

- 1 Funciones
- 2 Condicionales
- 3 Estructuras Cíclicas
- 4 Estructuras Ciclicas II
- 5 Cadenas
- 6 Arreglos y Matrices
- 7 Clases y Objetos**
- 8 Herencia y Polimorfismo
- 9 Relaciones Objetos-Clases



# Problemas I

## Problemas

- 1 Crear la clase `Persona` que tenga como atributos `Nombre`, `Edad`, `Genero` (H hombre, M mujer), `peso` y `altura`. El constructor debe recibir todos los parámetros para su inicialización.

La clase debe tener los siguientes métodos:

`calcularIMC()`: Calculará y retornará el Índice de Masa Corporal del objeto.

`esMayorDeEdad()`: Devolverá `true` si es mayor de edad, `false` en caso contrario.

`toString()`: Devolverá toda la información del objeto en un `String`.



# Problemas II

## Problemas (continuación)

- ② Crear la clase Contraseña que tenga como atributos Longitud (por defecto será 8) y Contraseña (String). El constructor debe recibir la longitud y generar una contraseña aleatoria con dicha longitud. La clase debe tener los siguientes métodos:

**esFuerte():** Devuelve un booleano si es fuerte o no. Una contraseña se considera fuerte si tiene al menos una mayúscula, una minúscula, y más de cinco números.

**cambiarContraseña(String nuevaContraseña):** Cambiará la contraseña actual por la indicada en los parámetros.

**toString():** Devolverá toda la información del objeto en un String.



# Problemas III

## Triki I

### Problemas (continuación)

- 3 El objetivo será crear una clase Triki con el que se pueda jugar. Considere la siguiente matriz

|   |   |   |
|---|---|---|
| a | a | a |
| a | a | a |
| a | a | a |

Cree la clase Triki que tenga como único atributo la matriz anteriormente contemplada.

El constructor no recibirá parámetros de entrada, y allí se inicializará la matriz con a's que indican que la casilla en cuestión está vacía.



# Problemas IV

## Triki II

### Problemas (continuación)

La clase tendrá los siguientes métodos:

**marcarCasilla(String simbolo, int fila, int columna):**

Marcará el símbolo indicado (lo ideal es que en el parámetro simbolo se pase un único carácter (X ó O). Dicho carácter se escribirá en la posición de la matriz (fila, columna).

**verificarGanador():** Verificará si hay un ganador. Retornará el ganador (si lo hay) del juego, devolviendo el carácter que usó para ganar. Si no hay ganador aún se devolverá una "a" indicando que no hay un ganador.

**verificarCasilla(int fila, int columna):** Devolverá el carácter que hay en la casilla que está en la posición (fila, columna) de la matriz.



# Problemas V

## PacMan I

### Problemas (continuación)

- 4 Considere el videojuego PacMan (1980). Se creará una simulación del personaje y el tablero. Para ello es necesario crear dos clases:
- Clase PacMan: Tendrá la información del personaje.
  - Clase Tablero: Tendrá la información del tablero.



# Problemas VI

## PacMan II

### Problemas (continuación)

Para la clase PacMan se deben definir como atributos el color (por defecto "Amarillo"), la puntuacion (por defecto 0) y las vidasRestantes (por defecto 3). El constructor debe recibir estos tres parámetros e inicializarlos.

Dicha clase tendrá los siguientes métodos:

**sumarPuntuacion():** Cada vez que se llame este método, se sumará +1 a la puntuación.

**obtenerPuntuación():** Retornará la puntuación actual del personaje.

**restarVida():** Cada vez que se llame este método, se restará -1 a las vidas.

**sigueVivo():** Devolverá un booleano indicando si aún le quedan vidas para seguir jugando.

# Problemas VII

## PacMan III

### Problemas (continuación)

Para la clase `Tablero` se requerirán dos atributos: el personaje (será un objeto de tipo `PacMan`) y el `nivel`. El constructor debe recibir el personaje (el cual debe haber sido creado anteriormente y pasarse como un parámetro en el constructor de la clase) e inicializar el `nivel` en 0.

Dicha clase tendrá el siguiente método:

**`comprobarNivelActual()`**: El funcionamiento es simple: cada 25 puntos del personaje equivalen a 1 nivel en el tablero. Este método obtendrá la puntuación del PacMan y fijará el nivel actual dependiendo de la cantidad de puntos que haya encontrado. (Por ejemplo, 57 puntos equivalen al nivel 2, 75 puntos al nivel 3, etc).



# Agenda

- 1 Funciones
- 2 Condicionales
- 3 Estructuras Cíclicas
- 4 Estructuras Ciclicas II
- 5 Cadenas
- 6 Arreglos y Matrices
- 7 Clases y Objetos
- 8 Herencia y Polimorfismo**
- 9 Relaciones Objetos-Clases



# Problemas varios I

## Problemas

- ① Modele mediante el uso de clases y herencia la organización al interior de una universidad (profesores, alumnos, directivos, empleados, ...) Utilice la siguiente clase como base para las demás:

```
public class Persona {  
    private String nombre;  
    private String apellidos;  
    private int edad;  
    //Constructor  
    public Persona (String nombre, String apellidos,  
                    int edad) {  
        this.nombre = nombre;  
        this.apellidos = apellidos;  
        this.edad = edad;  
    }  
}
```



# Problemas varios II

## Problemas

```
//Métodos  
public String getNombre () { return nombre; }  
public String getApellidos () { return apellidos; }  
public int getEdad () { return edad; }  
} //Cierre de la clase
```

- ② Usando el concepto de herencia, diseñe una superclase que sirva como base para los elementos que contiene una biblioteca. **Pista:** Piense en que características tienen en comun los libros, revistas, articulos y demas elementos presentes en cualquier biblioteca.



# Problemas varios III

## Problemas

- 3 Desarrolle la jerarquía de figuras geométricas de la diapositiva 7 en Java. Cambie el método pintar por toString e imprima en el información relevante de los objetos.
- 4 Modele una clase "Calculadora" que realice las cuatro operaciones básicas (suma, resta, multiplicación y división) Para ello haga uso del concepto de interfaces y diseñe dos interfaces, una que contenga las operaciones y otra que contenga constantes de utilidad para cálculos, como pi o e.



# Agenda

- 1 Funciones
- 2 Condicionales
- 3 Estructuras Cíclicas
- 4 Estructuras Ciclicas II
- 5 Cadenas
- 6 Arreglos y Matrices
- 7 Clases y Objetos
- 8 Herencia y Polimorfismo
- 9 Relaciones Objetos-Clases**

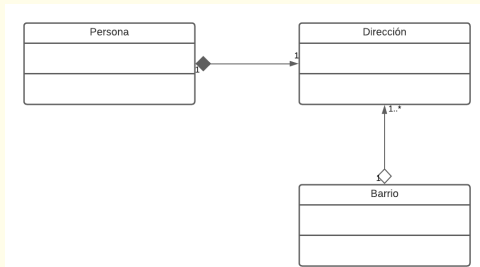




# Problema I

## Problema

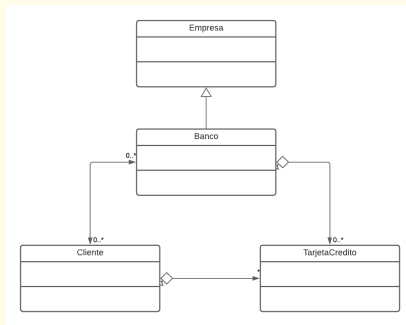
Implementar el siguiente diagrama de clases. Los atributos y métodos se dejan abiertos a la implementación. Nótese que la relación de Persona y Dirección es una **relación fuerte**, mientras que la relación de Dirección y Barrio es una **relación débil**.



# Problema II

## Problema

Implementar el siguiente diagrama de clases. Los atributos y métodos se dejan abiertos a la implementación. Tenga en cuenta las relaciones y las multiplicidades entre las clases.



# Problema III

## Red social

### Problema

Red social LinkedIn es una red social orientada al uso empresarial. Es reconocida por su sistema de *contactos*.

- Puedes agregar contactos de **primer grado**, que serán las personas con las que estés conectado directamente.
- Los contactos de **segundo grado** serán las personas que estén conectadas con tus contactos de primer grado.
- Los contactos de **tercer grado** serán las personas que estén conectadas con tus contactos de segundo grado.



# Problema III

## Red social

### Problema (continuación)

Red social Desarrollar un diagrama de clases recursivo, donde una persona pueda tener una lista de contactos, quienes a su vez serán personas que tienen contactos, etc.

Realizar la implementación del diagrama de clases en Java (véase el ejemplo de los directorios).



Imagen tomada desde <https://cartasdepresentacion.net/wp-content/uploads/2020/02/reglas-red-contactos-1024x745.jpg>

