

Introducción a Ciencias de la Computación 2020-2

Práctica 5: Arreglos

Pedro Ulises Cervantes González
confundeme@ciencias.unam.mx

Emmanuel Cruz Hernández
emmanuel_cruzh@ciencias.unam.mx

Adriana Sánchez del Moral
adrisanchez@ciencias.unam.mx

Víctor Zamora Gutiérrez
agua@ciencias.unam.mx

Fecha límite de entrega: 3 de Mayo de 2020.
Hora límite de entrega: 23:59.

1. Objetivo

Aprender a declarar arreglos de diversas formas, así como aprovechar cada una de sus propiedades, como lo es acceder a sus elementos, obtener la longitud de un arreglo, conocer su comportamiento, entre otros.

Además, se espera que sepan utilizar arreglos para resolver ciertos problemas cuya solución es más sencilla utilizando arreglos.

2. Actividad

Crea una clase llamada *Arreglos* que implemente la interfaz *InterfazArreglos* y realiza las actividades que se solicitan.

2.1. Actividad 1 (1 punto)

Implementa el método *espejo* que regresa el espejo de una matriz.
Por ejemplo:

$$\text{espejo}\left(\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}\right) \rightarrow \begin{pmatrix} 4 & 3 & 2 & 1 \\ 8 & 7 & 6 & 5 \\ 12 & 11 & 10 & 9 \\ 16 & 15 & 14 & 13 \end{pmatrix}$$

$$espejo\left(\begin{pmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \end{pmatrix}\right) \rightarrow \begin{pmatrix} 3 & 2 & 1 \\ 7 & 6 & 5 \end{pmatrix}$$

$$espejo\left(\begin{pmatrix} 1 & 2 \\ 5 & 6 \\ 6 & 8 \end{pmatrix}\right) \rightarrow \begin{pmatrix} 2 & 1 \\ 6 & 5 \\ 8 & 6 \end{pmatrix}$$

2.2. Actividad 2 (2 puntos)

Implementa el método *gato* que verifique si un tablero del juego *gato* tiene una partida ganada.

Por ejemplo:

$$gato\left(\begin{pmatrix} - & - & - \\ X & O & X \\ - & X & O \end{pmatrix}\right) \rightarrow N$$

$$gato\left(\begin{pmatrix} X & - & O \\ X & - & O \\ X & - & O \end{pmatrix}\right) \rightarrow E$$

$$gato\left(\begin{pmatrix} X & - & - \\ X & O & O \\ X & - & O \end{pmatrix}\right) \rightarrow X$$

$$gato\left(\begin{pmatrix} O & - & X \\ X & O & O \\ X & - & O \end{pmatrix}\right) \rightarrow O$$

2.3. Actividad 3 (1 punto)

Implementa el método *construyeFrase* que crea una cadena a partir de un arreglo bidimensional de caracteres.

Por ejemplo:

$$construyeFrase\left(\begin{pmatrix} h & o & l & a \\ m & u & n & d & o \\ l & o & c & o \end{pmatrix}\right)$$

→ "Hola mundo loco"

$$\text{construyeFrase}\left(\begin{pmatrix} e & s & t & a \\ e & s \\ u & n & a \\ f & r & a & s & e \\ c & o & n \\ c & a & r & a & c & t & e & r & e & s \end{pmatrix}\right)$$

→ "esta es una frase con caracteres"

2.4. Actividad 4 (1 punto)

Implementa el método *separaNumeros* que separe números impares de pares.

$$\text{separaNumeros}((1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7)) \rightarrow \begin{pmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 & 7 \end{pmatrix}$$

$$\text{separaNumeros}((5 \ 5 \ 2 \ 4 \ 5 \ 8 \ 12)) \rightarrow \begin{pmatrix} 2 & 4 & 8 & 12 \\ 5 & 5 & 5 \end{pmatrix}$$

$$\text{separaNumeros}((30 \ 8 \ 2 \ 4)) \rightarrow \begin{pmatrix} 30 & 8 & 2 & 4 \end{pmatrix}$$

2.5. Actividad 5 (1 punto)

Implementa el método *masRepetido* que encuentre el número más repetido en un arreglo.

$$\text{masRepetido}((3 \ 3 \ 10 \ 13 \ 24 \ 3)) \rightarrow 3$$

$$\text{masRepetido}((1 \ 2 \ 3 \ 4 \ 5 \ 6)) \rightarrow 1$$

$$\text{masRepetido}((2 \ 2 \ 4 \ 4)) \rightarrow 2$$

2.6. Actividad 6 (1 punto)

Implementa el método *estaContenido* que permite saber si todos los elementos de un arreglo están contenidos en otro.

$$\text{estaContenido}((2 \ 7 \ 11 \ 3 \ 5), (5 \ 5 \ 2 \ 4 \ 5 \ 8 \ 12)) \rightarrow false$$

$$\text{estaContenido}((2 \ 7 \ 5), (7 \ 5 \ 2 \ 4 \ 5)) \rightarrow true$$

$$\text{estaContenido}((2 \ 7 \ 12 \ 8), (7 \ 5 \ 2)) \rightarrow false$$

$$\text{estaContenido}((2 \ 7 \ 7 \ 7), (7 \ 5 \ 2)) \rightarrow true$$

2.7. Actividad 7 (2 puntos)

Implementa el método *pascal* que construye tantos niveles del triángulo de Pascal como se soliciten.

$$pascal(4) \rightarrow \begin{pmatrix} 1 \\ 1 & 1 \\ 1 & 2 & 1 \\ 1 & 3 & 3 & 1 \end{pmatrix}$$

$$pascal(5) \rightarrow \begin{pmatrix} 1 \\ 1 & 1 \\ 1 & 2 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

$$pascal(10) \rightarrow \begin{pmatrix} 1 \\ 1 & 1 \\ 1 & 2 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 4 & 6 & 4 & 1 \\ 1 & 5 & 10 & 10 & 5 & 1 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \\ 1 & 7 & 21 & 35 & 35 & 21 & 7 & 1 \\ 1 & 8 & 28 & 56 & 70 & 56 & 28 & 8 & 1 \\ 1 & 9 & 36 & 84 & 126 & 126 & 84 & 36 & 9 & 1 \end{pmatrix}$$

2.8. Actividad 8 (1 punto)

Implementa el método *colapsa* que permite eliminar los elementos repetidos de un arreglo.

$$colapsa((2 \ 2 \ 3 \ 3 \ 5)) \rightarrow (2 \ 3 \ 5)$$

$$colapsa((2 \ 1 \ 3 \ 2 \ 5)) \rightarrow (2 \ 1 \ 3 \ 5)$$

$$colapsa((2 \ 1 \ 2 \ 2 \ 5 \ 1 \ 2 \ 3 \ 2 \ 5 \ 1)) \rightarrow (2 \ 1 \ 5 \ 3)$$

3. Nota importante

Pon a prueba todos los métodos de la práctica en el método main de la clase *Arreglos*. Imprime en terminal las entradas y los resultados de cada método. **Usar el Scanner es opcional.**

4. Punto Extra

Crea una clase llamada *Ordenador* que implemente todos los métodos de la interfaz *InterfazOrdenador* y en el método *main* pon a prueba los métodos, mostrando las entradas y las salidas de cada método.

El formato de entrega del punto extra es el siguiente:

- *Practica05ParternoNombre*
 - src
 - Arreglos.java
 - InterfazArreglos.java
 - extra
 - InterfazOrdenador.java
 - Ordenador.java
 - Readme.txt

5. Materiales para consultar

1. Arreglos unidimensionales: <https://youtu.be/6zP9Au02gsM>
2. Recorrido sobre arreglos: <https://youtu.be/4x-qHiftnRA>
3. Arreglos bidimensionales: https://youtu.be/_tUncS0AsNE
4. Arreglos con Minecraft: <https://youtu.be/k24J92Hod50>
5. For each: https://youtu.be/DMV1DZVW_9M
6. Insertion Sort: <https://youtu.be/bB8Px8D9QdQ>
7. Selection Sort: <https://youtu.be/KCvr7eHXEHE>
8. Baile Insertion Sort: <https://youtu.be/R0a1U37913U>
9. Baile Selection Sort: <https://youtu.be/Ns4TPTC8whw>

6. Reglas Importantes

- Cumple con los lineamientos de entrega.
- Todos los archivos deberán contener nombre y número de cuenta.
- Tu código debe estar comentado. Esto abarca clases, atributos, métodos y comentarios extra.
- Para cada clase solicitada, crea un nuevo archivo.

- Utiliza correctamente las convenciones para nombrar variables, constantes, clases y métodos.
- Sólo se permite el uso de la biblioteca Scanner.
- **SÍ** se reciben prácticas con retraso. Por cada día se restará 1 punto.
- En caso de no cumplirse alguna de las reglas especificadas, se restará 0.5 puntos en tu calificación obtenida.