

A decorative graphic consisting of a teal circle on the left side of the slide. A horizontal bar, colored with a teal-to-white gradient, spans across the middle of the slide. The title 'Práctica #6 - Arreglos' is centered on this bar. Large, stylized square brackets are positioned on either side of the bar, with the left one being dark teal and the right one being light blue.

Práctica #6 - Arreglos

Arreglos unidimensionales

Arreglos bidimensionales

Composición

Arreglos de tipos primitivos

Clases con arreglos como atributos de instancia

Ciclos

Arreglos 1 dimensión

- Crea una clase de prueba con los siguientes métodos estáticos.
- En el método principal, main, invoca cada método codificado e imprime en pantalla los resultados que regresan.
- Para la prueba de los métodos, declara varios arreglos unidimensionales con datos incluidos.
- Imprime, antes de cada salida, un mensaje indicando qué arreglo(s) y qué método estás probando. Ejemplo:

```
Arreglo 1
5 10 6 12 12 2 13 12 14 12
Arreglo 2
14 0 7 0 6 11 8 5 13 1
Arreglo de suma
19 10 13 12 18 13 21 17 27 13
Arreglo 1
La suma de los nones es: 18
```

[Arreglos 1 dimensión]

- `public static int datoMayor(int[] arr)`
 - Regresa el dato mayor del arreglo que se recibe de parámetro
- `public static int datoMenor(int[] arr)`
 - Regresa el dato mayor del arreglo que se recibe de parámetro
- `public static double promedio(int[] arr)`
 - Regresa el promedio de los datos del arreglo
- `public static int[] datosMayores(int[] arr1, int[] arr2)`
 - Regresa un nuevo arreglo con los datos mayores de ambos arreglos. Es decir, si `arr1[0] > arr2[0]` entonces `arr3[0] = arr1[0]`, y así sucesivamente

[Arreglos 1 dimensión]

- `public static int cuentaMenores(int[] arr, int dato)`
 - Regresa el total de datos en *arr* que son menores al parámetro *dato*
- `public static int[] invierteArreglo(int[] arr)`
 - Regresa un nuevo arreglo con los datos de *arr* invertidos
- `public static int[] multiplosDe(int[] arr, int dato)`
 - Regresa un nuevo arreglo con los datos de *arr* que son múltiplos de *dato*.

Matrices

	0	1	2	3	4
0	20	10	9	3	1
1	0	2	8	55	4
2	99	28	30	91	6
3	5	12	13	14	7
4	22	39	21	40	67

Matrices

- Implementa en java una clase **Matriz** que permita crear matrices bidimensionales de enteros y que implemente operaciones de matrices de dos dimensiones.
- El único atributo de dicha clase Matriz es:
 - Un arreglo bidimensional, *mat*.
- La clase Matriz tiene como *constantes de clase* las siguientes constantes de tipo entero:
 - REN = 5
 - COL = 5

Matrices – Métodos de instancia

- Constructor de default
 - Que cree un arreglo bidimensional de REN renglones y COL columnas.
- Constructor
 - Que reciba dos atributos, *ren* y *col*, y cree el arreglo bidimensional, *mat*, de tamaño *ren* y *col*.
- sumaMatrices
 - Que cree y regrese un nuevo objeto Matriz con el resultado de la suma de dos objetos Matriz. Debes validar que ambas matrices a sumar sean del mismo tamaño.

Matrices – Métodos de instancia

- restaMatrices
 - Que cree y regrese un nuevo objeto Matriz con el resultado de la resta de dos objetos Matriz. Debes validar que ambas matrices a restar sean del mismo tamaño.
- setElement
 - Que actualice de los datos de un objeto Matriz. Debes recibir el renglón y columna a actualizar así como el valor a asignar. Valida que el renglón y columna sean válidos según el tamaño del arreglo bidimensional, *mat*, del objeto Matriz.
- toString
 - Regresa una cadena, String, que imprima el arreglo *mat* como una matriz cuadrada

Matrices – Métodos de instancia

- mayorMatriz
 - Que regrese el valor mayor del objeto Matriz.
- menorMatriz
 - Que regrese el valor menor del objeto Matriz.
- promedioMatriz
 - Que regrese el promedio, como un *double*, de los datos del objeto Matriz.

Matrices

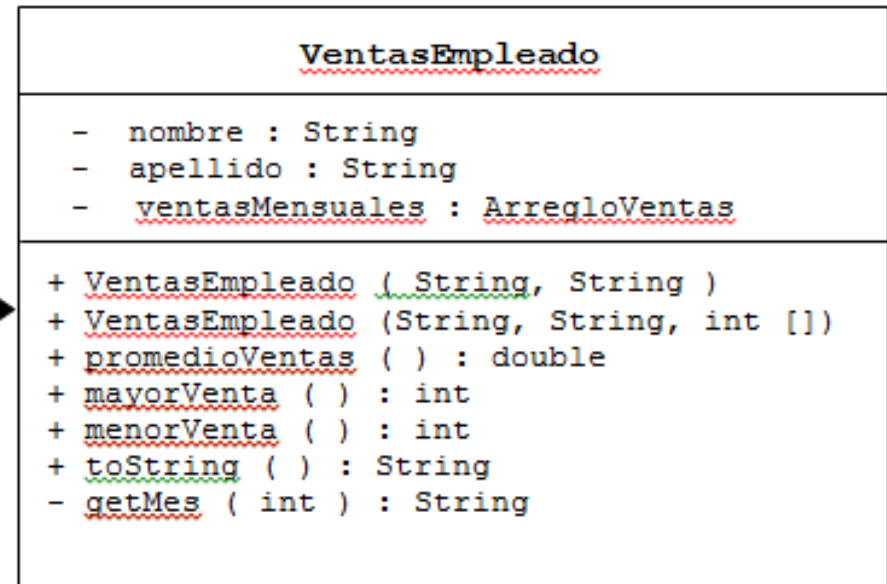
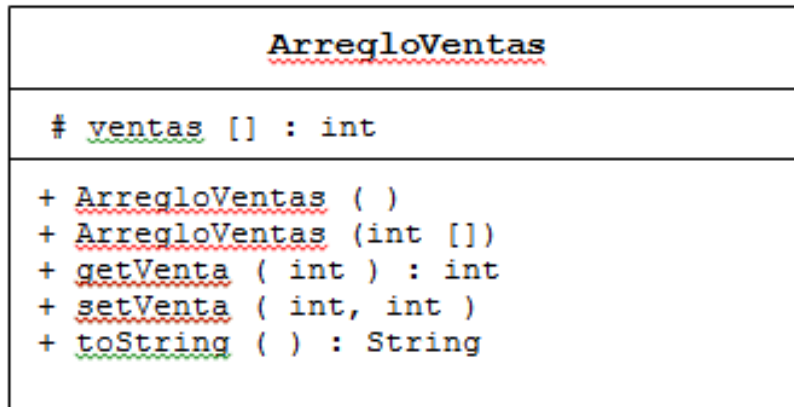
- Prueba tu clase con el programa **PruebaMatriz.java** que te proporciona tu profesora.
- Observa que en la clase de prueba falta verificar el buen funcionamiento de los métodos:
 - mayorMatriz
 - menorMatriz
 - promedioMatriz
- Así que deberás **AGREGAR** a la clase de prueba la correspondiente llamada a estos métodos para verificar que funcionan correctamente.

Ventas Empleados





[Ventas empleados





[ArregloVentas

ArregloVentas	
Atributo o Método	Descripción
ventas	Arreglo de tipo entero que representas las ventas anuales, mes por mes, de un empleado
<u>ArregloVentas</u> ()	Constructor que crea el arreglo <i>ventas</i> como un arreglo de enteros de tamaño 12 (representando 12 meses de ventas)
<u>ArregloVentas</u> (int [] arr)	Constructor que crea el arreglo <i>ventas</i> como un arreglo de enteros de tamaño 12 y, posteriormente, copia los elementos del arreglo <i>arr</i> , que se recibe de parámetro, uno a uno.
int <u>getVenta</u> (int pos)	Regresa el dato en el arreglo <i>ventas</i> que se encuentra en la posición <i>pos</i> , que se recibe de parámetro. Caso especial: - Si <i>pos</i> está fuera del rango del arreglo <i>ventas</i> deberá regresar -1
<u>setVenta</u> (int pos, int dato)	Método que cambia el dato que se encuentra en la posición <i>pos</i> , del arreglo de <i>ventas</i> , por el dato que se recibe de parámetro. Caso especial: - Si <i>pos</i> está fuera del rango del arreglo <i>ventas</i> no realizará cambio alguno.
String <u>toString</u> ()	Crea y regresa una cadena que contiene los datos del arreglo <i>ventas</i> , mes a mes. Por ejemplo: Enero: 12 Febrero: 20 Marzo: 45 y así sucesivamente para el resto de los meses



[VentaEmpleados

<u>VentasEmpleado</u>	
Atributo o Método	Descripción
nombre apellido <u>ventasMensuales</u>	Atributos de la clase <u>VentasEmpleado</u>
<u>VentasEmpleado</u> (String nombre, String apellido)	Constructor de <u>VentasEmpleado</u> que inicializa los atributos nombre y apellido con los parámetros <i>nombre</i> y <i>apellido</i> . El atributo <u>ventasMensuales</u> se inicializa utilizando el constructor de default de <u>ArregloVentas</u>
<u>VentasEmpleado</u> (String nombre, String apellido, int [] arr)	Constructor de <u>VentasEmpleado</u> que inicializa los atributos nombre y apellido con los parámetros <i>nombre</i> y <i>apellido</i> . El atributo <u>ventasMensuales</u> se inicializa utilizando el constructor de <u>ArregloVentas</u> que recibe un arreglo de ventas.
double <u>promedioVentas()</u>	Método que regresa el promedio de las ventas realizadas por el empleado.



[VentaEmpleados

String mayorVenta()	Método que regresa el nombre del mes con mayores ventas del empleado. Utiliza el método de utilidad getMes , explicado más adelante.
String menorVenta()	Método que regresa el nombre del mes con menores ventas del empleado. Utiliza el método de utilidad getMes , explicado más adelante.
String toString ()	Crea y regresa una cadena que contiene las ventas anuales del empleado. Utiliza el método toString() de ArregloVentas para resolver este método. Por ejemplo: <i>Las ventas de Marco Arellano son:</i> <i>Enero: 12</i> <i>Febrero: 20</i> <i>Marzo: 45</i> y así sucesivamente para el resto de los meses
String getMes (int mes)	Método que recibe un número entero y regresa al mes correspondiente. 0 = Enero 1 = Febrero Y así sucesivamente hasta el mes 11 = Diciembre Para cualquier otro caso regresa "Mes invalido!"

[VentaEmpleados]

void setVenta(int pos, int dato)

Actualiza las ventas del empleado en el mes, pos, por el valor, dato. Debes invocar al método setVenta del atributo ventasMensuales.

int getVenta(int pos)

Regresa las ventas del empleado del mes, pos. Debes invocar al método getVenta del atributo ventasMensuales.

[Prueba y salida]

- Prueba tus clases con el programa **PruebaVentas.java** que te proporciona tu profesora.
- La salida debe ser idéntica al archivo que te proporciona tu profesora: **Salida2 201711.txt**

¿Qué debes entregar?

- Enviar un correo electrónico con los siguientes documentos:

- Vectores.java
- Matriz.java
- PruebaMatriz.java
- VentaEmpleados.java
- ArregloVentas.java



Nombre del archivo .zip :
Practica06.zip

- Cada documento .java que generes o modifiques DEBE incluir, en la parte superior la siguiente documentación (con la información correcta):

```
/* Clase <nombre clase>.java  
  Autor 1: <nombre alumno 1> <matrícula alumno 1>  
  Autor 2: <nombre alumno 2> <matrícula alumno 2>  
  Fecha: <fecha de entrega>  
  Práctica # 6 – Arreglos  
*/
```

- Subject del correo: POO – Práctica #6 – Arreglos
- En el cuerpo del correo los nombres de los integrantes del equipo