



UNIDAD 4. ARREGLOS

MATERIA: Programación orientada a objetos I



3 DE JUNIO DE 2020

Guadalupe García Vázquez
MATRICULA: ES1921008556
Desarrollo de Software

Docente: Claudia Erika González Gómez

Actividad 1. Arreglos unidimensionales y multidimensionales

Contenido

Introducción.....	3
Investigación.....	4
Desarrollo Caso de estudio.....	9
Proyecto	10
Ejecución del programa	13
Conclusiones.....	15
Bibliografía.....	16

Introducción

En el siguiente trabajo se abordará el tema de los arreglos en Java.

Los arreglos en Java, también conocidos como arrays, agrupan valores del mismo tipo guardados en una misma variable y se puede acceder de manera independiente.

Estos Arrays pueden ser unidimensionales o multidimensionales, estos arrays son muy beneficios para almacenar varios valores en una sola variable, además su uso en Java es muy sencillo.

De acuerdo a lo anterior en un vector podemos almacenar elementos de cualquier tipo de dato (int, char, float, double etc.).

La declaración anterior de un vector en memoria se vería algo parecido a la siguiente imagen.

miArreglo[0]	miArreglo[1]	miArreglo[2]	miArreglo[3]	miArreglo[4]	miArreglo[5]
0	1	2	3	4	5

Las posiciones de los arreglos en java comienzan desde 0 hasta que ascienden en número según se necesite, pues las posiciones de un array contienen un valor determinado, ya que comienzan desde el elemento 0 (primer elemento del array) hasta los que se necesiten.

Función de los arreglos

Los arrays se utilizan para guardar objetos de la misma clase, este es el medio donde se alojan, es posible acceder a cada elemento individual del arreglo, estos son objetos en java que para crearlos se establece como primer paso la declaración del arreglo que consiste en la forma primera con que se va a armar el arreglo, por consiguiente se procede a la creación del arreglo Java para darle vida o cabida en la programación que se está haciendo, luego se le da inicio al arreglo para ponerlo en marcha y por último es cuando se va a proceder con el uso del arreglo.

Durante el uso de los arreglos se puede notar que dentro de ellos se puede guardar más de una variable, es decir que se pueden guardar múltiples variables de acuerdo a su tamaño o capacidad y es importante no olvidar que las variables que se van a guardar dentro del objeto siempre deben ser del mismo tipo; si las variables son numéricas, se guardan números, lo mismo sucede si las variables son caracteres o strings.

Investigación

1.- Investiga e identifica los diferentes tipos y usos que tienen los arreglos en Java.

Antes de identificar los tipos de arreglos es necesario definir que es un arreglo.

Un arreglo es una estructura de datos que contiene una serie de variables que se encuentran en un solo lugar (ese lugar es el arreglo) a las que se accede mediante índices, o posiciones del arreglo.

.

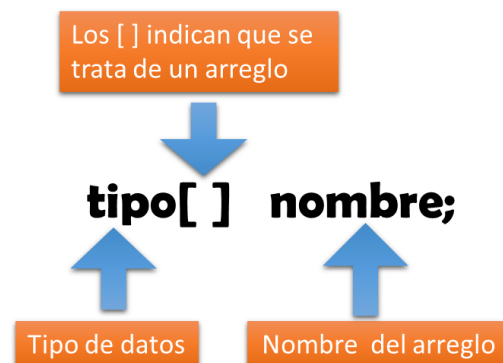
Declaración:

La estructura de declaración de un arreglo es la siguiente:

```
tipo_dedato [] nombre_variable; //tambien
puede declararse "tipo_dedato
nombre_variable[;"
```

ejemplos:

- 1 String array[]; //donde array es un arreglo que guardara variables tipo Strings
- 2 double array2[]; //donde array2 es un arreglo que guardara variabls tipo double
- 3 int [] array3; //donde array3 es un arreglo que guardara variables tipo int(numericas)



Definir tamaño de arreglos:

Para asignar a un arreglo su tamaño o capacidad, se hace de la siguiente forma:

```
arreglo = new tipo_dedato[capacidad];
```

ejemplo:

```
1 array = new String[10]; //el arreglo array guardara hasta un maximo de 10  
variables de tipo String
```

Para realizar una declaración de un arreglo y definir su tamaño en una misma línea se puede usar la siguiente estructura:

```
1 String array[] = new String[10]; //se declara un arreglo y se asigna su tamaño en  
una misma línea
```

Al arreglo también se le conoce con el nombre de vector o array. Según Deitel & Deitel:

“Un arreglo es un grupo de posiciones de memoria contiguas, todas las cuales tienen el mismo nombre y el mismo tipo”

Propiedades de los arrays

- Los arrays se utilizan como contenedores para almacenar datos relacionados (en vez de declarar variables por separado para cada uno de los elementos del array).
- Todos los datos incluidos en el array son del mismo tipo. Se pueden crear arrays de enteros de tipo `int` o de reales de tipo `float`, pero en un mismo array no se pueden mezclar datos de tipo `int` y datos de tipo `float`.
- El tamaño del array se establece cuando se crea el array (con el operador `new`, igual que cualquier otro objeto).

Tipos de arrays en Java

Existen dos tipos de arrays, los llamados multidimensionales y los unidimensionales son los más fáciles y sencillos de comprender además de ser bastante útiles, ambos sirven para diversas aplicaciones; de forma general a estas dimensiones se les denominan matrices. La forma a la que se accede a los arrays son los corchetes y se entiende por acceso, tanto el intentar leer el elemento, como asignarle un valor. Para mostrar el contenido de un arreglo Java es importante tener en cuenta que para mostrar el contenido hay que utilizar dos bucles, es decir es necesario conocer el número de columnas y filas totales.

Arreglos unidimensionales

El rango de un arreglo también se conoce como las dimensiones del arreglo. Un arreglo con un rango de 1 se denomina arreglo unidimensional.

Un array (o arreglo) unidimensional consta de una fila, también otros la llaman vectores de una dimensión.

Si queremos guardar las calificaciones de un conjunto alumnos podríamos utilizar un array unidimensional.

Sintaxis

tipo nombre[];

Cuando declaramos un array unidimensional y queremos recorrer este array para obtener un valor guardado debemos recorrer de 0 a n-1, “n” es igual al tamaño de nuestro array en Java.

Arreglos multidimensional

Un arreglo con un rango mayor que 1 se llama un arreglo multidimensional

Estos arrays manejan dos o más dimensiones.

Array multidimensional java

Para entender mejor imagínense un cubo de Ruby donde cada cuadradito almacena un dato, donde el cubo Ruby sería un array de 3x3x3.

Sintaxis

int notas[][];

2.- Considerando tu investigación, analiza las diferencias entre los tipos de arreglos y redáctalas en un documento.

Arreglos unidimensionales	Arreglos multidimensional
Sólo se manejaban a manera de renglón (o fila).	se manejan a manera de tabla, donde se tendrán filas y columnas, por lo que este tipo de arreglos ofrece un mejor manejo de datos cuando éstos deben manejar dimensiones, como la información

<p>Declaración</p> <pre>tipo [] nombreArray; tipo nombreArray[];</pre> <p>Ejemplos:</p> <pre>int [] ventas; //array de datos de tipo int llamado ventas double [] temperaturas; //array de datos de tipo double llamado temperaturas String [] nombres; //array de datos de tipo String llamado nombres</pre> <div style="border: 1px solid green; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>tipoDeDato[] identificadorDelArray = new tipoDeDato[tamañoDelArray]</pre> </div> <p>Inicialización</p> <p>Al igual que con cualquier variable utilizada, en el lenguaje Java necesitan ser inicializadas a un valor. Bajo esta premisa, cuando se declara (crea) un array en Java se inicializa automáticamente, dependiendo de estos casos:</p> <ol style="list-style-type: none"> 1. false cuando el tipo de dato es boolean. 2. null para un array de objetos de tipo Object. 3. El equivalente a 0 (vacío para String) para cualquier otro tipo de dato 	<p>mostrada en hojas de cálculo o tablas de bases de datos</p> <p>Declaración</p> <p>Los arreglos multidimensionales tienen más de una dimensión. En Java, las dimensiones se manejan por medio de corchetes (muy parecido a los arrays unidimensionales), dentro de los cuales se escribe cada dimensión (cantidad de datos que guardará). Cada dimensión irá dentro de un corchete. La sintaxis general para declarar e instanciar un array es:</p> <pre>Tipo_de_variable[][]... [] Nombre_del_array = new Tipo_de_variable[dimensión1][dimensión2]... [dimensiónN];</pre> <div style="border: 1px solid green; border-radius: 10px; padding: 5px; margin: 10px 0;"> <pre>tipoDeDato[][][]... identificadorDelArray = new tipoDeDato[tamañoDimensión][tamañoDimensión][tamañoDimensión]...</pre> </div> <p>Recorrido</p> <p>El recorrido de los elementos de un array multidimensional es de la misma manera que en un arreglo unidimensional, sólo agregando un corchete por cada dimensión que se añada, considerando que se puede acceder a los valores de los elementos de un arreglo bidimensional a través del nombre del arreglo y dos subíndices.</p>
--	--

<p>Operaciones</p> <p>Al referirse a operaciones con arrays no es en el sentido clásico de operaciones aritméticas sobre ellos (aunque se pueden hacer y se ejemplificará más adelante), sino a cómo se obtienen sus elementos, haciendo un profundo énfasis en la importancia que tienen los índices para ello</p>	<p>Operaciones</p> <p>Como los arrays unidimensionales y multidimensionales son tan parecidos en su método de acceso (que de hecho es idéntico), se ejemplifica los arrays multidimensionales de la misma manera que se hizo con los arrays unidimensionales, sólo modificando sus dimensiones.</p>
---	---

Desarrollo Caso de estudio

Una empresa que se dedica a la venta de automóviles seminuevos, necesita crear un programa que le permita almacenar la marca, modelo, el valor comercial y el tipo del automóvil (F->Familiar, P->Publico o C->Carga) de “n” automóviles. El programa permitirá agregar los datos de un nuevo automóvil.

Para resolver el problema debe imprimir un listado de los automóviles almacenados con sus datos, además calcular y mostrar su valor final con base en: Para automóviles que su valor es menor a 150,000 pesos se le cobrara el IVA de 16%, para los demás el IVA es del 20%.

Datos:

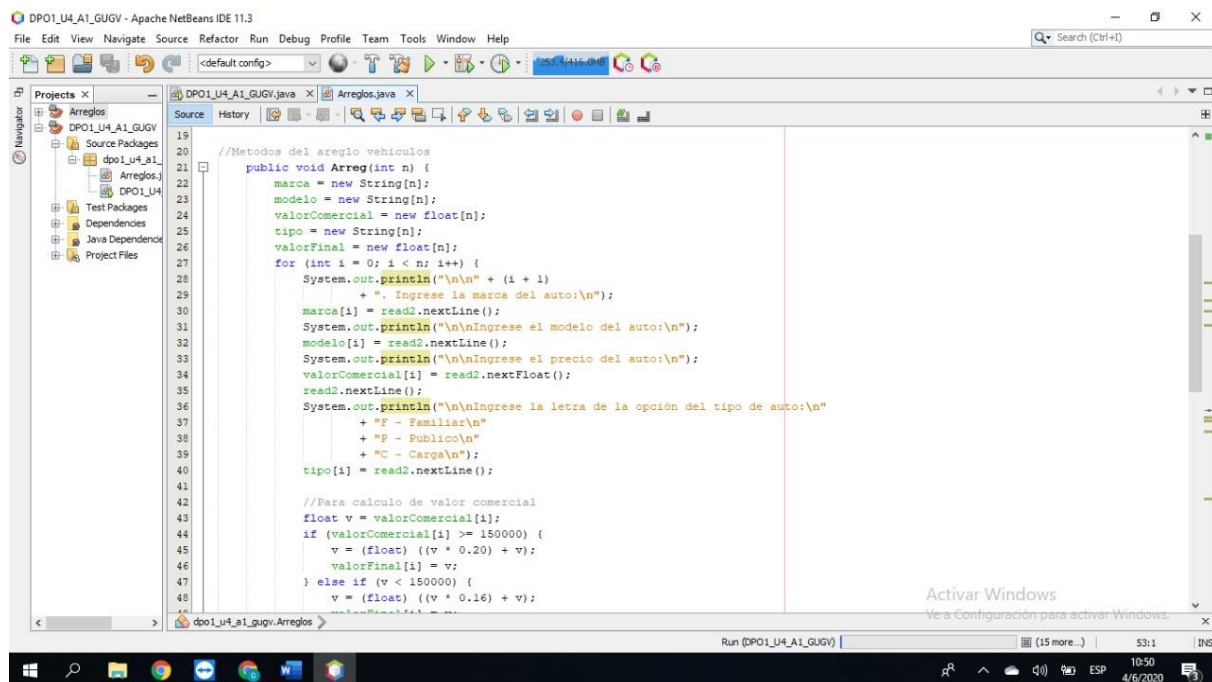
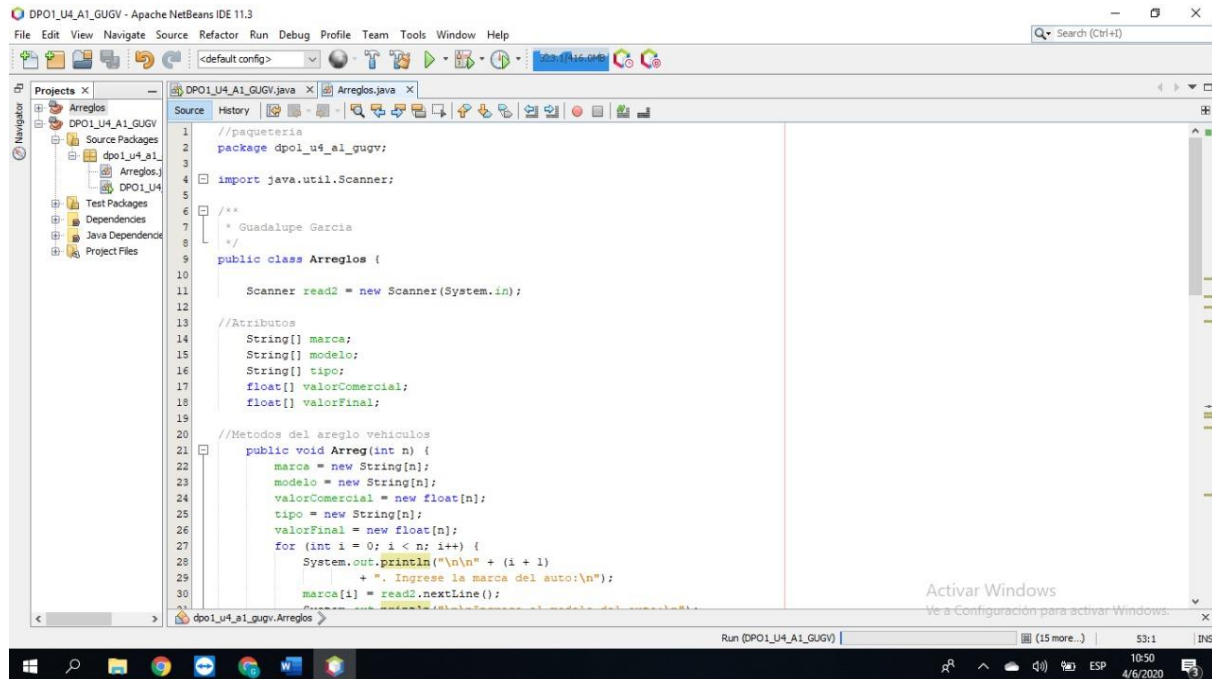
- Marca (es una cadena de caracteres)
- Modelo (es una cadena de caracteres)
- Valor Comercial (es un valor numérico con decimales)
- Tipo del automóvil (F-Familiar, P-Publico o C-Carga, una inicial)
- Valor final (es un valor para precio monetario)

Analiza y desarrollo el código para establecer la solución del caso de estudio anterior, debes de utilizar arreglos en dicha solución.

El programa deberá hacer el recorrido de los arreglos, imprime en pantalla lo que este almacenando en ellos.

Proyecto

Clase arreglos, en ella se desarrolla el código de los elementos del objeto de vehículos, todas las características que tendrá el arreglo.



Dentro de éste también se hace el manejo de excepciones.

```

42 //Para calculo de valor comercial
43 float v = valorComercial[i];
44 if (valorComercial[i] >= 150000) {
45     v = (float) ((v * 0.20) + v);
46     valorFinal[i] = v;
47 } else if (v < 150000) {
48     v = (float) ((v * 0.16) + v);
49     valorFinal[i] = v;
50 }
51 }
52 }
53 public void Lista(int n) {
54     System.out.println("\n\nListado de autos: \n");
55     System.out.println("\n");
56     for (int i = 0; i < n; i++) {
57         System.out.println(i + 1 + ". Marca: " + marca[i]
58             + ", Modelo: " + modelo[i]
59             + ", Precio: " + valorComercial[i]
60             + ", Tipo: " + tipo[i]
61             + ", Precio Final: " + valorFinal[i] + "\n");
62     }
63     pressAnyKeyToContinue();
64 }
65
66 //Manejo de excepciones
67 public static void pressAnyKeyToContinue() {
68     String seguir;
69     Scanner teclado = new Scanner(System.in);
70     System.out.println("Press Enter key to continue...");
71     try {

```

En la clase main principal, se realizó la instancia el arreglo codificado, así como los métodos para el ingreso de los datos por parte del usuario.

```

1 //nombre del paquete
2 package dpo1_u4_a1_gugv;
3
4 //librerías a utilizar
5 import java.util.Scanner;
6
7 /**
8  * Guadalupe Garcia
9  */
10 public class DPO1_U4_A1_GUGV {
11
12     public static void main(String[] args) {
13
14         //instanciar objeto-arreglo
15         Arreglos arr = new Arreglos();
16
17         Scanner read = new Scanner(System.in);
18         int nRegistros;
19         int i = 0;
20         int opcion;
21
22         System.out.println("\n\n....."
23             + "\n Bienvenido al portal de registro de automoviles\n");
24         System.out.println("¿Cuántos registro de automóvil, desea ingresar?\n");
25         nRegistros = read.nextInt();
26         read.nextLine();
27
28         do {
29             System.out.println("\n\n....."
30                 + "\n Seleccione la opción que desee\n");

```

```

13 //instanciar objeto-arreglo
14 Arreglos arr = new Arreglos();
15
16 Scanner read = new Scanner(System.in);
17 int nRegistros;
18 int i = 0;
19 int opcion;
20
21
22 System.out.println("\n.....")
23 + "\n Bienvenido al portal de registro de automoviles\n");
24 System.out.println("¿Cuántos registro de automóvil, desea ingresar?\n");
25 nRegistros = read.nextInt();
26 read.nextLine();
27
28 do {
29     System.out.println("\n.....")
30     + "\n\n Seleccione la opción que desea\n"
31     + "(Número de la opción deseada:)\n\n"
32     + "1.Registro de datos de autos\n"
33     + "2.Ver la lista de autos registrados\n"
34     + "3.Salir\n\n");
35
36     opcion = read.nextInt();
37     read.nextLine();
38
39 //Switch- El resultado se compara con cada <valor> consecutivamente y se ejecutan las instrucciones.
40 switch (opcion) {
41     case 1:
42         //...
43     case 2:
44         //...
45     case 3:
46         //...
47     }
48 } while (opcion != 3);
49
50 }
51
52
53
54
55
56
57
58
59
60
61
62

```

Para determinar el tamaño del arreglo (los espacios guardados en memoria para el arreglo) se le pide al usuario que ingrese el número de datos registros que desea hacer, con lo cual el tamaño es establecido. Para indicar al sistema que el proceso lo va a realizar hasta que se a completen los registros se utiliza la sentencia **Switch**, para evaluar el siguiente proceso a ejecutar por el sistema de acuerdo a la opción elegida por el usuario.

```

37 read.nextLine();
38
39 //Switch- El resultado se compara con cada <valor> consecutivamente y se ejecutan las instrucciones.
40 switch (opcion) {
41     case 1:
42         if (i < nRegistros) {
43             arr.Arreg(nRegistros);
44             i++;
45         } else {
46             System.out.println("Registros completos\n");
47         }
48         break;
49
50     case 2:
51         arr.Lista(nRegistros);
52         break;
53
54     case 3:
55         System.out.println("\n Proceso completado\n");
56         break;
57 }
58 } while (opcion != 3);
59
60 }
61
62

```

Ejecución del programa

The image displays two sequential screenshots of the Apache NetBeans IDE 11.3, showing the execution of a Java program named 'DPO1_U4_A1_GUGV'. The 'Output - Run (DPO1_U4_A1_GUGV)' window shows the program's output, which includes prompts for user input and the program's responses.

First Screenshot:

```

--- exec-maven-plugin:1.5.0:exec (default-cli) @ DPO1_U4_A1_GUGV ---

Bienvenido al portal de registro de automoviles

¿Cuántos registro de automóvil, desea ingresar?
2

.....
Seleccione la opción que desea
(Número de la opción deseada:)

1.Registro de datos de autos
2.Ver la lista de autos registrados
3.Salir

1

1. Ingrese la marca del auto:
NISSAN VERSA

Ingrese el modelo del auto:
Sense
  
```

Second Screenshot:

```

1. Ingrese la marca del auto:
NISSAN VERSA

Ingrese el modelo del auto:
Sense

Ingrese el precio del auto:
267,400

Ingrese la letra de la opción del tipo de auto:
F - Familiar
P - Publico
C - Carga
F

2. Ingrese la marca del auto:
NISSAN VERSA

Ingrese el modelo del auto:
  
```

PROGRAMACIÓN ORIENTADA A OBJETOS | Desarrollo de Software

DPO1_U4_A1_GUGV - Apache NetBeans IDE 11.3

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Projects: Arreglos, DPO1_U4_A1_GUGV, Source Packages, dpo1_u4_a1, Arreglos, DPO1_U4, Test Packages, Dependencies, Java Dependencies, Project Files

Source: dpo1_u4_a1_gugv.DPO1_U4_A1_GUGV

Output - Run (DPO1_U4_A1_GUGV)

```
Ingrese la letra de la opción del tipo de auto:
F - Familiar
P - Publico
C - Carga
F

-----
Seleccione la opción que desea
(Número de la opción deseada:)

1.Registro de datos de autos
2.Ver la lista de autos registrados
3.Salir

2

Listado de autos:

1. Marca: NISSAN VERSA, Modelo: Sense, Precio: 267400.0, Tipo: F, Precio Final: 320980.0
2. Marca: NISSAN VERSA, Modelo: Exclusive, Precio: 317600.0, Tipo: F, Precio Final: 381120.0
Press Enter key to continue...
```

Activar Windows
Ve a Configuración para activar Windows.

Run (DPO1_U4_A1_GUGV) | (14 more...) | 61:2 | INS

Windows taskbar: 11:01 4/6/2020

DPO1_U4_A1_GUGV - Apache NetBeans IDE 11.3

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Projects: Arreglos, DPO1_U4_A1_GUGV, Source Packages, dpo1_u4_a1, Arreglos, DPO1_U4, Test Packages, Dependencies, Java Dependencies, Project Files

Source: dpo1_u4_a1_gugv.DPO1_U4_A1_GUGV

Output - Run (DPO1_U4_A1_GUGV)

```
Ingrese la letra de la opción del tipo de auto:
F - Familiar
P - Publico
C - Carga
F

-----
Seleccione la opción que desea
(Número de la opción deseada:)

1.Registro de datos de autos
2.Ver la lista de autos registrados
3.Salir

2

Listado de autos:

1. Marca: NISSAN VERSA, Modelo: Sense, Precio: 267400.0, Tipo: F, Precio Final: 320980.0
2. Marca: NISSAN VERSA, Modelo: Exclusive, Precio: 317600.0, Tipo: F, Precio Final: 381120.0
Press Enter key to continue...
```

Activar Windows
Ve a Configuración para activar Windows.

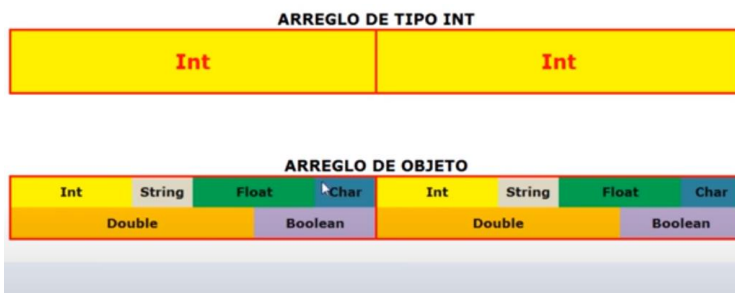
Run (DPO1_U4_A1_GUGV) | (14 more...) | 61:2 | INS

Windows taskbar: 11:01 4/6/2020

Conclusiones

En este siguiente tema hemos abordamos el tema de los arreglos, conocidos también como Arrays.

Cómo vi los arrays se utilizan para guardar objetos de la misma clase, haciendo énfasis a “objetos”, ya que a largo de la materia hemos visto lo que es la



programación orientada a objetos y éste tema también me ha permitido entender las ventajas de trabajar de esta manera. La imagen que muestro a lado, generaliza el beneficio que tenemos, en los Arrays se puede confundir el

concepto de que solo se puede utilizar un tipo de datos, y en un arreglo como, ilustrado en la imagen superior podemos observar que está definido solo por el tipo Int, pero cuándo hacemos un arreglo de un objeto, la composición del objeto será determinado por la declaración de los atributos y sus tipos de datos que tendrá el objeto.

Por ejemplo, en el ejercicio presentado, los atributos contenidos en el objeto fueron, Marca, Modelo, Valor comercial, Tipo de automóvil, Costo final, permitiendo ingresar diferentes tipos de datos y guardarlos en un arreglo.

Bibliografía

- Berzal, F. (Consultado 03/06/2020). *Vectores y matrices*. Obtenido de Vectores y matrices - Java: Obtenido de: <https://elvex.ugr.es/decsai/java/pdf/6A-Arrays.pdf>
- Creatividad Codificada. (Consultado el: 03/06/2020). *Arrays o arreglos en Java*. Obtenido de Obtenido de_ <https://creatividadcodificada.com/java/arrays-o-arreglos-en-java/>
- David Hackro. (2014). *Registro de alumnos(arreglo de objetos) Resuelto con JAVA*. Obtenido de Consultado 04/06/2020 Obtenido de: <https://www.youtube.com/watch?v=Z4k582CoCvs>
- Grupo Codesi. (Consultado 03/06/2020). *Arreglos en Java*. Obtenido de Ventajas de utilizar Arreglos en Java: Obtenido de: <http://www.buscaminegocio.com/cursos-de-java/arreglos-java.html>
- Universidad Abierta y a Distancia de México. (Consultado el 03/06/2020). *Unidad 4. Arreglos*. Obtenido de Programación orientada a objetos I: Obtenido de: https://ceit.unadmexico.mx/contenidos/DCEIT/BLOQUE2/DS/02/DPO1/U4/descargables/DPO1_U4_Contenido.pdf
- V, J. (s.f.). *Arreglos en java*. Obtenido de Consultado 03/06/2020: Obtenido de: <http://codigoprogramacion.com/cursos/java/96-arreglos-en-java.html>