



Instituto Politecnico Nacional
Escuela Superior de Cómputo



Bocanegra Heziquio Yestlanezi

Practica 2

Arreglo de objetos

2. Reservaciones de asiento en un camión

30 de abril de 2021

2CM13

Programación Orientada a Objetos



Contenido

Introducción	3
Arreglo de objetos	3
Que es un array en Java.....	3
Inicialización de Arrays en Java.....	4
Desarrollo	5
Conclusiones	10
Bibliografía	11
Código 1 Código Práctica 2	7
Imagen 1 Compilación de código desde CMD	8
Imagen 2 Tablero de 4x11	8
Imagen 3 Reservando asientos	9

Introducción

Arreglo de objetos

Los **arrays dinámicos** son muy potentes porque permiten crear colecciones de tamaño variable que podemos agrandar o empequeñecer en función de nuestras necesidades [1].

cuando se conoce el número de elementos en una colección y éste va a ser invariable, será más eficiente utilizar una colección de tamaño fijo a la que denominamos array estático, arreglo estático, formación o vector. Si utilizamos el término array o arreglo, a secas, entenderemos que hacemos alusión a un array estático. El uso de arrays estáticos tiene ventajas e inconvenientes:

VENTAJAS	INCONVENIENTES
<ul style="list-style-type: none">- Acceso y operación con elementos más eficiente.- Permiten almacenar tanto objetos como tipos primitivos directamente	<ul style="list-style-type: none">- Rigidez. No se pueden redimensionar (aunque sí copiar a otros arrays estáticos de mayor tamaño).

La numeración de los índices de los arrays va desde cero hasta (número de elementos – 1). Tener en cuenta que la variable que es el nombre del array, p.ej. misNombres, lo que contiene es un puntero o referencia al objeto que es en sí el array. Un array en Java puede considerarse un “objeto especial”. Se crea con la sentencia new como el resto de objetos, pero sin embargo no hay una clase específica en Java que defina el tipo de los arrays. Dada una declaración del tipo `int [] cochesPorHora = new int [24];`, hay ciertos errores habituales frente a los que hay que estar atentos:

- Pensar que los índices van de 0 a 24. Falso: van de 0 a 23.
- Pensar que el número de elementos total es 23. Falso: son 24.
- Usar `cochesPorHora [24]`. El índice 24 no existe y el uso de esa expresión daría lugar a un error en tiempo de ejecución del tipo “`ArrayIndexOutOfBoundsException`” [1].

Que es un array en Java

Un array Java es una estructura de datos que nos permite almacenar una ristra de datos de un mismo tipo. El tamaño de los arrays se declara en un primer momento y no puede cambiar en tiempo de ejecución como puede producirse en otros lenguajes [2]. La declaración de un array en Java y su inicialización se realiza de la siguiente manera:

```
tipo_dato nombre_array[];  
nombre_array = new tipo_dato[tamano];
```


Por ejemplo, podríamos declarar un array de caracteres e inicializarlo de la siguiente manera:

```
char arrayCaracteres[];  
arrayCaracteres = new char[10];
```

Los arrays [Java](#) se numeran desde el elemento cero, que sería el primer elemento, hasta el tamaño-1 que sería el último elemento. Es decir, si tenemos un array de diez elementos, el primer elemento sería el cero y el último elemento sería el nueve. Para acceder a un elemento específico utilizaremos los corchetes de la siguiente forma. Entendemos por acceso, tanto el intentar leer el elemento, como asignarle un valor [2].

```
arrayCaracteres[numero_elemento];
```

Inicialización de Arrays en Java

Existe una forma de inicializar un array en [Java](#) con el contenido, amoldándose su tamaño al número de elementos a los que le inicialicemos. Para inicializar un array [Java](#) utilizaremos las llaves de la siguiente forma:

```
tipo_dato array[] = {elemento1,elemento2,...,elementoN};
```

Así, por ejemplo, podríamos inicializar un array [Java](#) o una matriz [Java](#):

```
// Tenemos un array de 5 elementos.
```

```
char array[] = {'a','b','c','d','e'};
```

```
// Tenemos un array de 4x4 elementos.
```

```
int array[][] = { {1,2,3,4}, {5,6,7,8}};
```

Desarrollo

2. Reservaciones de asiento en un camión

- ♡ Tablero de botones de 4x11 (numerar de forma consecutiva los asientos, es decir que la etiqueta de cada botón sea un número).
- ♡ Reservar el asiento cuando se haga clic sobre el botón que lo representa y cambiar el color y etiqueta de dicho botón.
- ♡ Se pulsa un botón especial (el 45 = $44 + 1$) para mostrar cuales asientos se han reservado.
- ♡ Los números de los asientos reservados se pueden almacenar en una variable de tipo String la cuál hará el papel de acumulador. Cada que se hace clic en un botón se puede agregar el número de asientos a la cadena.

```

//Librerias principales
import java.awt.event.*;
import java.awt.*;
import java.io.*;
import javax.swing.*;
import java.util.*;

public class Asientos extends JFrame implements ActionListener {
    JButton botones[]=new JButton[45];
    Button conta;
    int contador;
    Label etiq;
    JLabel todos;
    String test, mostra, aux;
    JPanel Superior, Inferior;
    Container c = getContentPane();

    //Constructor
    public Asientos(){

        c.setLayout(new GridLayout(2,1));

        Superior = new JPanel();
        Superior.setLayout(new GridLayout(12,4));
        contador=0;
        for(int i=1; i<45; i++){
            Superior.add(botones[i]=new JButton(""+i));
            botones[i].addActionListener(this);
            botones[i].setBackground(java.awt.Color.pink);
        }

        Superior.add(conta = new Button("Reservados"));
        conta.addActionListener(this);
        Superior.add(etiq = new Label("0"));

        Inferior = new JPanel();
        todos = new JLabel("Ocupados");
        Font auxFont=todos.getFont();
        todos.setFont(new Font(auxFont.getFontName(), auxFont.getStyle(), 12));
        Inferior.add(todos);
        mostra = " ";
        aux = " ";

        c.add(Superior); c.add(Inferior);

        Superior.setSize(400,400);
    }
}

```

```
Inferior.setSize(400,40);

setSize(350, 450); setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if(e.getSource()==conta){
        String resultado = Integer.toString(contador);
        etiq.setText(resultado);
        mostra = mostra + aux;
        todos.setText(" " + mostra);
        aux = " "; test = " ";
    }
    else{
        contador++;
        JButton btn=(JButton)e.getSource();
        btn.setEnabled(false);
        btn.setBackground(java.awt.Color.white);
        test = btn.getText().toString();
        aux = test + "," + aux;
        contador = contador++;
    }
}

//Inicio main
public static void main(String s[]){
    new Asientos();
}
}
```

Código 1 Código Práctica 2


```
Símbolo del sistema - java Asientos
D:\POO\Practica 2>dir
El volumen de la unidad D es DATA
El número de serie del volumen es: 7060-3731

Directorio de D:\POO\Practica 2
29/04/2021 04:51 p. m. <DIR> .
29/04/2021 04:51 p. m. <DIR> ..
29/04/2021 05:00 p. m. 3,274 Asientos.class
29/04/2021 05:01 p. m. 2,462 Asientos.java
29/04/2021 04:48 p. m. 3,268 camion.class
29/04/2021 04:48 p. m. 1,845 camion.java
4 archivos 10,849 bytes
2 dirs 752,300,277,760 bytes libres

D:\POO\Practica 2>javac Asientos.java
D:\POO\Practica 2>java Asientos
```

Imagen 1 Compilación de código desde CMD

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
Reservados	0		
Ocupados			

Imagen 2 Tablero de 4x11

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
Reservados	3		

22,9,1,

Imagen 3 Reservando asientos

Al momento de hacer clic en los asientos que quieres reservar, estos cambian de color, pasan de ser rosa a ser blancos, lo que indica entonces que los asientos seleccionados ya han sido reservados, ya que también se indica al lado del botón “reservados” cuantos asientos están en esa categoría y debajo muestra el numero de los asientos reservados.

Conclusiones

Con base en lo aprendido en las sesiones de clases impartidas mediante zoom, pude notar que la complejidad de cada practica va aumentando conforme avanza el curso, a veces los conocimientos impartidos en clase no son suficientes, lo que significa que no debemos conformarnos solo con lo que aprendemos en la escuela si queremos seguir avanzando y creciendo como programadores, debemos expandir nuestros recursos, tales como estudiar por nuestra cuenta de otras maneras, en cursos, libros, investigaciones o incluso repasando los apuntes tomados en clases.

Respecto a la práctica, puedo concluir que fue un poco similar con lo que se hacia en estructura de datos con los arreglos en C, ya que en este caso se podía almacenar el valor en una variable de tipo String que sería la que permitirá almacenar el valor de cada asiento que se iba reservando, así que viéndolo de esa manera, fue un poco más fácil trabajar con ello, aunque la practica tiene un grado de dificultad mayor para mí, espero seguir esforzándome en las practicas siguientes y poder lograr concluir el trabajo final.



Bibliografía

- [1] A. Rodríguez, «Aprenderaprogramar.com,» Copyright 2006-2021 aprenderaprogramar.com, [En línea]. Available: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=635:arrays-o-arreglos-java-ejemplos-de-objetos-y-tipos-primitivos-length-tipos-de-colecciones-cu00669b&catid=68&Itemid=188. [Último acceso: 30 abril 2021].
- [2] ManualWeb, «ManualWeb,» 2021. [En línea]. Available: <http://www.manualweb.net/java/arrays-java/>. [Último acceso: 30 abril 2021].