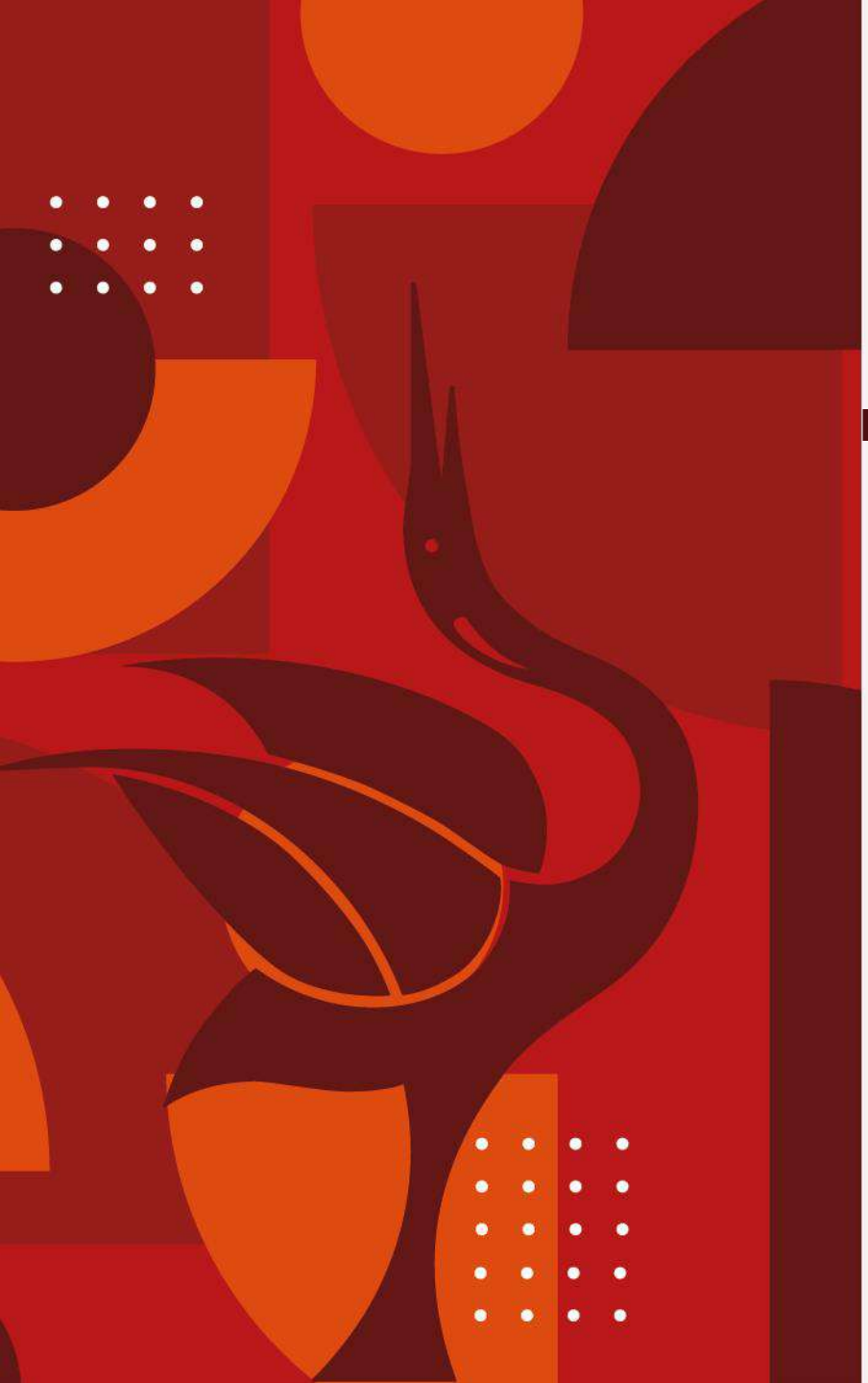


Área Académica de Computación y Electrónica
Licenciatura en Ciencias Computacionales
Programación Orientada a Objetos

Arreglos dinámicos en Java

Elaboraron:

M.C.C. Iliana Castillo Pérez

Dra. Verónica Martínez Lazcano

Dra. Yira Muñoz Sánchez

Fecha de elaboración: noviembre 2024

Abstract

An array or collection is a structure that allows a set of elements to be stored under the same name, allowing the management of a greater amount of information. In the Java language there is a library that allows generating dynamic arrays, giving the advantage of not pre-establishing a fixed number of elements when the array is declared. This material shows the declaration and use of the most common methods for its manipulation.

Keywords:

Dynamic arrays, ArrayList, dynamic collections.

Resumen

Un arreglo o colección es una estructura que permite almacenar un conjunto de elementos bajo un mismo nombre, permitiendo la gestión de un mayor número de información. En el lenguaje Java existe una librería que permite generar arreglos dinámicos, otorgando la ventaja de no preestablecer un número fijo de elementos al declarar el arreglo. En este material se muestra la declaración y uso de los métodos más comunes para su manipulación.

Palabras clave:

Arreglos dinámicos, arreglos, colecciones dinámicas.

Objetivo de Aprendizaje

Implementar el uso de ArrayList de acuerdo a sus características y métodos, bajo el paradigma orientado a objetos, con la intención de generar y manipular colecciones dinámicas de objetos.

Arreglos dinámicos en Java

Introducción

Los arreglos dinámicos permiten almacenar y administrar una colección de elementos de una forma flexible, permitiendo que dichos elementos se agreguen o eliminen en el momento, en función de las necesidades.

Arreglos dinámicos en Java

Introducción

El lenguaje de programación Java, proporciona la clase ArrayList para la creación y manejo de arreglos dinámicos. Dicha clase se localiza en el paquete java.util. Para utilizarla es necesario importarla, por lo que se debe agregar la siguiente línea de código al inicio del archivo .java:

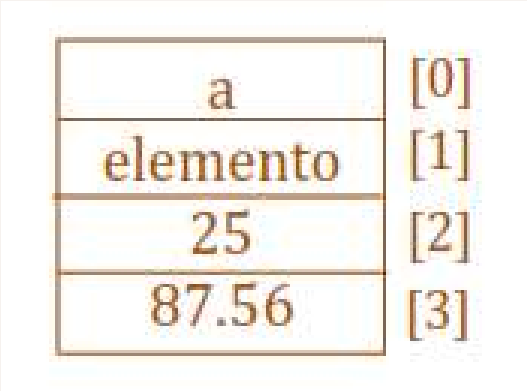
```
import java.util.ArrayList;
```

Clase ArrayList

La clase ArrayList permite declarar un arreglo utilizando la siguiente sintaxis:

```
ArrayList nombreArray = new ArrayList();
```

La instrucción anterior genera un ArrayList vacío en el que se podrá almacenar cualquier tipo de dato.



Clase ArrayList

Para que un ArrayList almacene sólo un tipo de objetos, la declaración debe seguir la siguiente sintaxis:

```
ArrayList<tipo> nombreArray = new ArrayList();
```

Donde **tipo** representa la clase de objetos que se almacenarán en el arreglo dinámico.

Ejemplo:

```
ArrayList<Persona> listaPersonas = new ArrayList();
```

Preámbulo

Se describirán los métodos más utilizados para manipular los arreglos dinámicos, para ello tomar en consideración la siguiente declaración:

```
public class Persona{
    private String nombre;
    private int edad;
    private String sexo;

    public Persona(String n, int e, String s){
        this.nombre = n;
        this.edad = e;
        this.sexo = s;
    } // Constructor explícito con argumentos
    … //continúa el resto de la clase
}
```

Métodos de ArrayList

Método	Descripción
add(x)	Agrega el elemento x al final del arreglo. Devuelve true.

Ejemplo:

```
ArrayList<Persona> listaPersonas = new ArrayList();
Persona persona1 = new Persona("Ana",21,"Femenino");
listaPersonas.add(persona1);
```

También es posible utilizar la siguiente instrucción:

```
listaPersonas.add(new Persona("Ian",22,"Masculino"));
```

Métodos de ArrayList

Método	Descripción
add(posición,x)	Agrega el elemento x en la posición indicada. Desplaza el elemento que se encuentra actualmente en esa posición.

Ejemplo:

```
Persona persona2 = new Persona("Luna",19,"Femenino");
listaPersonas.add(1,persona2);
```

También es posible utilizar la siguiente instrucción:

```
listaPersonas.add(1, new Persona("Luna",19,"Femenino"));
```

Métodos de ArrayList

Método	Descripción
clear()	Elimina todos los elementos de la lista.

Ejemplo:

```
listaPersonas.clear();
```

La lista estará vacía después de que se ejecute la llamada al método indicado.

Métodos de ArrayList

Método	Descripción
get(posición)	Devuelve el elemento que está en la posición indicada.

Ejemplo:

```
Persona persona3 = new Persona();
persona3 = listaPersonas.get(1);
```

La instrucción anterior, devuelve el objeto almacenado en la posición `1` del arreglo `listaPersonas` y lo asigna a `persona3`.

Métodos de ArrayList

Método	Descripción
set(índice,x)	Reemplaza el elemento que se encuentra en la posición indicada, por el objeto x. Devuelve el elemento sustituido.

Ejemplo:

```
Persona persona4 = new Persona("Gael",21,"Masculino");
listaPersonas.set(1, persona4);
```

También es posible utilizar la siguiente instrucción:

```
listaPersonas.set(1, new Persona("Gael",21,"Masculino"));
```

La instrucción anterior insertará el nuevo objeto `persona` en la posición `1` del arreglo `listaPersonas`.

Métodos de ArrayList

Método	Descripción
isEmpty()	Devuelve true si el arreglo está vacío.

Ejemplo:

```
ArrayList<Persona> listaPersonas2 = new ArrayList();
if(listaPersonas2.isEmpty()){
    System.out.println("No hay elementos");
}
```

El código anterior, imprimirá en pantalla el mensaje No hay elementos ya que `listaPersonas2` está vacío.

Métodos de ArrayList

Método	Descripción
size()	Devuelve el número de elementos en el arreglo (int).

Ejemplo:

```
listaPersonas2.add(new Persona("Karla",20,"Femenino");
listaPersonas2.add(new Persona("Hugo",22,"Masculino");
int num = listaPersonas2.size();
System.out.println("Hay " + num + " elementos.");
```

El código anterior, imprimirá en pantalla el mensaje Hay 2 elementos. .

Métodos de ArrayList

Método	Descripción
remove(índice)	Elimina el elemento en la posición especificada en la lista. Desplaza cualquier elemento posterior hacia la izquierda (resta uno de sus índices).

Ejemplo:

```
listaPersonas2.remove(0);
```

La instrucción anterior elimina el objeto almacenado en la posición `0` del arreglo `listaPersonas2`.

Bibliografía

- Deitel, H. y Deitel, P. (2016). Como Programar en Java, 9na. Edición, Pearson Prentice Hall.
- GeeksforGeeks. (2024). ArrayList in Java. Obtenido de GeeksforGeeks: https://www.geeksforgeeks.org/arraylist-in-java/
- Joyanes, L. (2011), Programación en Java 6. Algoritmos, programación orientada a objetos e interfaz gráfica de usuarios, McGraw-Hill.
- Schildt, H. (2017). Java: The Complete Reference, Tenth Edition, McGraw-Hill.
- W3Schools. (2024). Java ArrayList. Obtenido de W3Schools: https://www.w3schools.com/java/java_arraylist.asp

Datos de contacto

M.C.C. Iliana Castillo Pérez
Profesora Investigadora
Área Académica de Computación y Electrónica
Instituto de Ciencias Básicas e Ingeniería
Universidad Autónoma del Estado de Hidalgo
Correo-e: ilianac@uaeh.edu.mx

Dra. Verónica Martínez Lazcano
Profesora Investigadora
Área Académica de Computación y Electrónica
Instituto de Ciencias Básicas e Ingeniería
Universidad Autónoma del Estado de Hidalgo
Correo-e: vlazcano@uaeh.edu.mx

Dra. Yira Muñoz Sánchez
Profesora Investigadora
Escuela Superior de Cd. Sahagún
Universidad Autónoma del Estado de Hidalgo
Teléfono: 77171 72000 ext. 5300
Correo-e: yira@uaeh.edu.mx

UAEH®