Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II

Arreglos (Arrays): Búsquedas y Ordenamientos

Nombre: Jhonatan Benjamin Mamani Céspedes **CUI: 20232188**

Link del repositorio GitHub: https://github.com/JBenjamin01/fp2-24b.git

Ejercicio 1: Crear un arreglo de n Personas,almacenando datos que ingresemos para cada una.Luego que los muestre.

- Primero se crea la clase Persona para manejar los objetos de esa clase.

```java
public class Persona {
    private String nombre;
    private int edad;
    private char genero;

    public Persona(String nombre, int edad, char genero) {
        this.nombre = nombre;
        this.edad = edad;
        this.genero = genero;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

    public void setgenero(char genero) {
        this.genero = genero;
    }


    public String getNombre() {
        return nombre;
    }

    public int getEdad() {
        return edad;
    }

    public char getGenero() {
        return genero;
    }

    @Override
    public String toString() {
        return "Nombre: " + nombre +
                ", Edad: " + edad +
                ", Genero: " + genero;
    }
}
```

- Ahora se realiza el pedido del valor de n, se inicializa el arreglo de personas y se guardan a partir del constructor de la misma clase, finalmente se imprimen usando el método toString() dentro de un ciclo iterador for.

```java
import java.util.*;
public class Ejercicio1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingresa el número de personas: ");
        int n = sc.nextInt();

        Persona[] personas = new Persona[n];

        for (int i = 0; i < n; i++) {
            sc.nextLine();
            System.out.println("Persona " + (i + 1) + ":");

            System.out.print("Nombre: ");
            String nombre = sc.nextLine();

            System.out.print("Edad: ");
            int edad = sc.nextInt();

            System.out.print("Género (M/F): ");
            char genero = sc.next().charAt(0);

            personas[i] = new Persona(nombre, edad, genero);
        }

        System.out.println("\nLista de personas ingresadas:");
        for (int i = 0; i < n; i++)
            System.out.println("- " + personas[i]);
    }
}
```

Ejemplo de salida en la consola tras la compilación del programa:

```
Ingresa el número de personas: 3
Persona 1:
Nombre: Jose
Edad: 10
Género (M/F): M
Persona 2:
Nombre: Carla
Edad: 15
Género (M/F): F
Persona 3:
Nombre: Jhonatan
Edad: 18
Género (M/F): M

Lista de personas ingresadas:
- Nombre: Jose, Edad: 10, Genero: M
- Nombre: Carla, Edad: 15, Genero: F
- Nombre: Jhonatan, Edad: 18, Genero: M
```

Ejercicio 2: Mostrar los datos del amigo más joven y del mayor de todos.

- Reutilizando el código del ejercicio 1 para crear el arreglo con los objetos Persona, ahora se toman dos objetos (amigo mayor y menor) que se actualizan tras cada comparación al iterar sobre todas las personas, finalmente, estos se imprimen.

```java
import java.util.*;
public class Ejercicio2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingresa el número de personas: ");
        int n = sc.nextInt();

        Persona[] personas = new Persona[n];

        for (int i = 0; i < n; i++) {
            sc.nextLine();
            System.out.println("Persona " + (i + 1) + ":");

            System.out.print("Nombre: ");
            String nombre = sc.nextLine();

            System.out.print("Edad: ");
            int edad = sc.nextInt();

            System.out.print("Género (M/F): ");
            char genero = sc.next().charAt(0);

            personas[i] = new Persona(nombre, edad, genero);
        }

        // EJERCICIO 2:

        Persona masJoven = personas[0];
        Persona masMayor = personas[0];

        for (int i = 1; i < n; i++) {
            if (personas[i].getEdad() < masJoven.getEdad())
                masJoven = personas[i];
            if (personas[i].getEdad() > masMayor.getEdad())
                masMayor = personas[i];
        }

        System.out.println("\nEl amigo más joven es:");
        System.out.println(masJoven);

        System.out.println("\nEl amigo más viejo es:");
        System.out.println(masMayor);
    }
}
```

Ejemplo de salida en la consola tras la compilación del programa:

```
Ingresa el número de personas: 4
Persona 1:
Nombre: Jose
Edad: 13
Género (M/F): M
Persona 2:
Nombre: Carlos
Edad: 10
Género (M/F): m
Persona 3:
Nombre: Rodrigo
Edad: 17
Género (M/F): M
Persona 4:
Nombre: Maria
Edad: 21
Género (M/F): F

El amigo más joven es:
Nombre: Carlos, Edad: 10, Genero: m

El amigo más viejo es:
Nombre: Maria, Edad: 21, Genero: F
```

Ejercicio 3: Hacer un programa que cree un arreglo con capacidad para 100 contactos (al menos nombre y celular), que permita ingresar los datos requeridos hasta que el usuario ingrese "q" en nombre. Luego, que imprima todos los datos de los contactos ingresados. Finalmente, que el usuario ingrese un nombre y el programa muestre su celular correspondiente. Verificar que el número de celular tenga sólo dígitos.

- Para empezar, se crea la clase Contacto para el ejercicio con los atributos de nombre y celular, su constructor y su método sobreescrito.

```java
public class Contacto {
    private String nombre;
    private String celular;

    public Contacto(String nombre, String celular) {
        this.nombre = nombre;
        this.celular = celular;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public void setCelular(String celular) {
        this.celular = celular;
    }

    public String getNombre() {
        return nombre;
    }

    public String getCelular() {
        return celular;
    }

    @Override
    public String toString() {
        return "Nombre: " + nombre + ", Celular: " + celular;
    }
}
```

- Con la clase Contacto se procede a trabajar con objetos de dicha clase, en el ejercicio se usa un arreglo de 100 contactos y se sigue una lógica similar a los trabajos anteriores.

```java
import java.util.*;
public class Ejercicio3 {
    private static Contacto[] contactos = new Contacto[100];
    private static int c = 0;

    public static void ingresarContactos() {
        Scanner sc = new Scanner(System.in);
        String nombre, celular;

        while (true) {
            System.out.print("Ingresa el nombre del contacto (o 'q' para salir): ");
            nombre = sc.nextLine();

            if (nombre.equalsIgnoreCase("q"))
                break;

            do {
                System.out.print("Ingresa el número de teléfono: ");
                celular = sc.nextLine();

                if (esNumeroValido(celular))
                    break;
                else
                    System.out.println("Número inválido. Debe contener exactamente 9 dígitos.");
            } while (true);

            contactos[c++] = new Contacto(nombre, celular);
        }
    }

    public static boolean esNumeroValido(String numero) {
        if (numero.length() != 9) return false;

        for (int i = 0; i < numero.length(); i++)
            if (!Character.isDigit(numero.charAt(i)))
                return false;
        return true;
    }

    public static void imprimirContactos() {
        System.out.println("\nContactos ingresados:");
        for (int i = 0; i < c; i++)
            System.out.println(contactos[i]);
    }
```

```java

    public static void buscarContacto(String nombre) {
        for (int i = 0; i < c; i++)
            if (contactos[i].getNombre().equalsIgnoreCase(nombre)) {
                System.out.println("El número de " + nombre + " es: " + contactos[i].getCe
                return;
            }
        System.out.println("No se encontró un contacto con el nombre: " + nombre);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        ingresarContactos();

        imprimirContactos();

        System.out.print("\nIngresa el nombre del contacto para buscar su número: ");
        String nombreBuscado = sc.nextLine();
        buscarContacto(nombreBuscado);

        sc.close();
    }
```

Ejemplo de salida en la consola tras la compilación del programa:

```
Ingresa el nombre del contacto (o 'q' para salir): jose
Ingresa el número de teléfono: 977123456
Ingresa el nombre del contacto (o 'q' para salir): mario
Ingresa el número de teléfono: 4567891235
Número inválido. Debe contener exactamente 9 dígitos.
Ingresa el número de teléfono: q
```