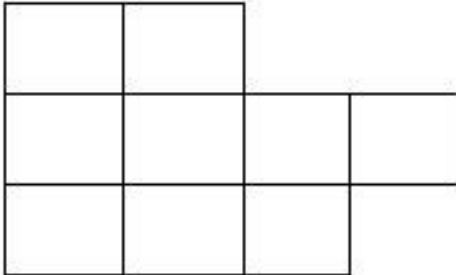


Curso de Java a distancia

Clase 9: Matrices irregulares

Java nos permite crear matrices irregulares. Se dice que una matriz es irregular si la cantidad de elementos de cada fila varía. Luego podemos imaginar una matriz irregular:



Como podemos ver la fila cero tiene reservado dos espacios, la fila uno reserva cuatro espacios y la última fila reserva espacio para tres componentes. Para crear la matriz irregular del gráfico:

La declaración es la misma que para matrices regulares:

```
int [][] mat;
```

Primero creamos la cantidad de filas dejando vacío el espacio que indica la cantidad de columnas:

```
mat=new int[3][];
```

Luego debemos ir creando cada fila de la matriz indicando la cantidad de elementos de la respectiva fila:

```
mat[0]=new int[2];
```

```
mat[1]=new int[4];
```

```
mat[2]=new int[3];
```

Luego la forma para acceder a sus componentes es similar a las matrices regulares, siempre teniendo en cuenta y validando que exista dicha componente:

```
mat[0][0]=120;
```

Dará un error si queremos cargar la tercer componente de la fila cero (esto debido a que no existe):

```
mat[0][2]=230;
```

Luego si queremos saber la cantidad de filas que tiene la matriz:

```
Sytem.out.println(mat.length);
```

Si queremos saber la cantidad de elementos de una determinada fila:

```
Sytem.out.println("Cantidad de elementos de la fila 0:"+mat[0].length);
```

```
Sytem.out.println("Cantidad de elementos de la fila 1:"+mat[1].length);
```

```
Sytem.out.println("Cantidad de elementos de la fila 2:"+mat[2].length);
```

Problema 1:

Confeccionaremos un programa que permita crear una matriz irregular y luego imprimir la matriz en forma completa.

Programa:

```
import java.util.Scanner;
```

```
public class MatrizIrregular1 {
```

```
    private Scanner teclado;
```

```
    private int[][] mat;
```

```
    public void cargar() {
```

```
        teclado=new Scanner(System.in);
```

```
System.out.print("Cuantas fila tiene la matriz:");

int filas=teclado.nextInt();

mat=new int[filas][];

for(int f=0;f<mat.length;f++) {

    System.out.print("Cuantas elementos tiene la fila " + f + ":");

    int elementos=teclado.nextInt();

    mat[f]=new int[elementos];

    for(int c=0;c<mat[f].length;c++) {

        System.out.print("Ingrese componente:");

        mat[f][c]=teclado.nextInt();

    }

}

}

public void imprimir() {

    for(int f=0;f<mat.length;f++) {

        for(int c=0;c<mat[f].length;c++) {

            System.out.print(mat[f][c]+" ");

        }

        System.out.println();

    }

}
```

```

public static void main(String[] ar) {

    MatrizIrregular1 ma=new MatrizIrregular1();

    ma.cargar();

    ma.imprimir();

}
}

```

Primero creamos la cantidad de filas que tendrá la matriz (en los corchetes para las columnas no disponemos valor):

```

System.out.print("Cuántas fila tiene la matriz:");

int filas=teclado.nextInt();

mat=new int[filas][];

```

Dentro del primer for pedimos que ingrese la cantidad de elementos que tendrá cada fila y utilizamos el operador new nuevamente, pero en este caso se están creando cada fila de la matriz (Java trata a cada fila como un vector):

```

for(int f=0;f<mat.length;f++) {

    System.out.print("Cuántas elementos tiene la fila " + f + ":");

    int elementos=teclado.nextInt();

    mat[f]=new int[elementos];
}

```

Dentro del for interno hacemos la carga de las componentes propiamente dicho de la matriz (podemos ir cargando cada fila a medida que las vamos creando):

```

for(int c=0;c<mat[f].length;c++) {

    System.out.print("Ingrese componente:");

    mat[f][c]=teclado.nextInt();

}

```

Luego imprimimos la matriz en forma completa teniendo cuidado las condiciones que disponemos en cada for.

El primer for se repite tantas veces como filas tiene la matriz: `f<mat.length` y el for interno se repite tantas veces como elementos tiene la fila que estamos procesando `c<mat[f].length`:

```
for(int f=0;f<mat.length;f++) {  
  
    for(int c=0;c<mat[f].length;c++) {  
  
        System.out.print(mat[f][c]+" ");  
  
    }  
  
    System.out.println();  
  
}
```

Problemas propuestos

1. Confeccionar una clase para administrar una matriz irregular de 5 filas y 1 columna la primer fila, 2 columnas la segunda fila y así sucesivamente hasta 5 columnas la última fila (crearla sin la intervención del operador)
Realizar la carga por teclado e imprimir posteriormente.
2. Confeccionar una clase para administrar los días que han faltado los 3 empleados de una empresa.
Definir un vector de 3 elementos de tipo String para cargar los nombres y una matriz irregular para cargar los días que han faltado cada empleado (cargar el número de día que faltó)
Cada fila de la matriz representan los días de cada empleado.
Mostrar los empleados con la cantidad de inasistencias.
Cuál empleado faltó menos días.

Solución

```
import java.util.Scanner;

public class MatrizIrregular2 {

    private Scanner teclado;

    private int[][] mat;

    public void cargar() {

        teclado=new Scanner(System.in);

        mat=new int[5][];

        for(int f=0;f<mat.length;f++) {

            mat[f]=new int[f+1];

            for(int c=0;c<mat[f].length;c++) {

                System.out.print("Ingrese componente:");

                mat[f][c]=teclado.nextInt();

            }

        }

    }

    public void imprimir() {

        for(int f=0;f<mat.length;f++) {

            for(int c=0;c<mat[f].length;c++) {

                System.out.print(mat[f][c]+" ");

            }

        }

    }

}
```

```
    }  
  
    System.out.println();  
  
    }  
  
}
```

```
public static void main(String[] ar) {  
  
    MatrizIrregular2 ma=new MatrizIrregular2();  
  
    ma.cargar();  
  
    ma.imprimir();  
  
    }  
}
```

```
import java.util.Scanner;  
  
public class MatrizIrregular3 {  
  
    private Scanner teclado;  
  
    private String[] nombres;  
  
    private int[][] dias;  
  
  
    public void cargar() {  
  
        teclado=new Scanner(System.in);
```

```
nombres=new String[3];

dias=new int[3][];

for(int f=0;f<nombres.length;f++) {

    System.out.print("Ingrese el nombre del empleado:");

    nombres[f]=teclado.next();

    System.out.print("Cuántas días faltó el empleado:");

    int faltas=teclado.nextInt();

    dias[f]=new int[faltas];

    for(int c=0;c<dias[f].length;c++) {

        System.out.print("Ingrese nro de día:");

        dias[f][c]=teclado.nextInt();

    }

}

}

public void inasistencias() {

    for(int f=0;f<nombres.length;f++) {

        System.out.println(nombres[f] + " faltó " + dias[f].length + " días");

    }

}

public void empleadoMensosFaltas() {

    int faltas=dias[0].length;
```



```
String nom=nombres[0];

for(int f=1;f<dias.length;f++) {

    if (dias[f].length<faltas) {

        faltas=dias[f].length;

        nom=nombres[f];

    }

}

System.out.println("El empleado que faltó menos es "+nom+" con "+faltas+" faltas.");

}

public static void main(String[] ar) {

    MatrizIrregular3 ma=new MatrizIrregular3();

    ma.cargar();

    ma.inasistencias();

    ma.empleadoMensosFaltas();

}

}
```

Constructor de la clase

En Java podemos definir un método que se ejecute inicialmente y en forma automática. Este método se lo llama constructor.

El constructor tiene las siguientes características:

- Tiene el mismo nombre de la clase.

- Es el primer método que se ejecuta.
- Se ejecuta en forma automática.
- No puede retornar datos.
- Se ejecuta una única vez.
- Un constructor tiene por objetivo inicializar atributos.

PROBLEMA 1:

Se desea guardar los sueldos de 5 operarios en un vector. Realizar la creación y carga del vector en el constructor.

PROGRAMA:

```
import java.util.Scanner;

public class Operarios {

    private Scanner teclado;

    private int[] sueldos;

    public Operarios()
    {
        teclado=new Scanner(System.in);

        sueldos=new int[5];

        for(int f=0;f<5;f++) {

            System.out.print("Ingrese valor de la componente:");

            sueldos[f]=teclado.nextInt();

        }

    }

    public void imprimir() {

        for(int f=0;f<5;f++) {
```

```
        System.out.println(sueldos[f]);  
    }  
}
```

```
public static void main(String[] ar) {  
  
    Operarios op=new Operarios();  
  
    op.imprimir();  
  
}  
}
```

Como podemos ver es el mismo problema que resolvimos cuando vimos vectores. La diferencia es que hemos sustituido el método cargar con el constructor:

```
public Operarios()  
{  
  
    teclado=new Scanner(System.in);  
  
    sueldos=new int[5];  
  
    for(int f=0;f<5;f++) {  
  
        System.out.print("Ingrese valor de la componente:");  
  
        sueldos[f]=teclado.nextInt();  
  
    }  
  
}
```

Como la clase se llama Operarios el constructor tiene el mismo nombre, no disponemos la palabra clave void ya que el constructor no puede retornar datos.

La ventaja de plantear un constructor en lugar de definir un método con cualquier nombre es que se llamará en forma automática cuando se crea un objeto de esta clase:

```
public static void main(String[] ar) {
```

```
Operarios op=new Operarios();
```

Cuando se crea el objeto op se llama al método constructor.

Finalmente llamamos al método imprimir:

```
op.imprimir();
```

Problema 2:

Plantear una clase llamada Alumno y definir como atributos su nombre y su edad. En el constructor realizar la carga de datos. Definir otros dos métodos para imprimir los datos ingresados y un mensaje si es mayor o no de edad (edad >=18)

Programa:

```
import java.util.Scanner;

public class Alumno {

    private Scanner teclado;

    private String nombre;

    private int edad;

    public Alumno() {

        teclado=new Scanner(System.in);

        System.out.print("Ingrese nombre:");

        nombre=teclado.next();

        System.out.print("Ingrese edad:");

        edad=teclado.nextInt();

    }

    public void imprimir() {
```

```
        System.out.println("Nombre:"+nombre);

        System.out.println("Edad:"+edad);

    }

    public void esMayorEdad() {

        if (edad>=18) {

            System.out.print(nombre+" es mayor de edad.");

        } else {

            System.out.print(nombre+" no es mayor de edad.");

        }

    }

    public static void main(String[] ar) {

        Alumno alumno1=new Alumno();

        alumno1.imprimir();

        alumno1.esMayorEdad();

    }

}
```

Declaramos la clase Persona, sus tres atributos y definimos el constructor con el mismo nombre de la clase:

```
public class Alumno {

    private Scanner teclado;

    private String nombre;
```

```
private int edad;
```

```
public Alumno() {  
  
    teclado=new Scanner(System.in);  
  
    System.out.print("Ingrese nombre:");  
  
    nombre=teclado.next();  
  
    System.out.print("Ingrese edad:");  
  
    edad=teclado.nextInt();  
  
}
```

En la main el constructor se llama en forma automática cuando creamos un objeto de la clase Alumno:

```
public static void main(String[] ar) {  
  
    Alumno alumno1=new Alumno();
```

Los otros dos métodos deben llamarse por su nombre y en el orden que necesitamos:

```
    alumno1.imprimir();  
  
    alumno1.esMayorEdad();
```

Problemas propuestos

1. Confeccionar una clase que represente un empleado. Definir como atributos su nombre y su sueldo. En el constructor cargar los atributos y luego en otro método imprimir sus datos y por último uno que imprima un mensaje si debe pagar impuestos (si el sueldo supera a 3000)
2. Implementar la clase operaciones. Se deben cargar dos valores enteros en el constructor, calcular su suma, resta, multiplicación y división, cada una en un método, imprimir dichos resultados.

Solución

```
import java.util.Scanner;

public class EmpleadoFabrica {

    private Scanner teclado;

    String nombre;

    float sueldo;

    public EmpleadoFabrica() {

        teclado=new Scanner(System.in);

        System.out.print("Ingrese el nombre del empleado:");

        nombre=teclado.next();

        System.out.print("Ingrese su sueldo:");

        sueldo=teclado.nextFloat();

    }

    public void pagaImpuestos() {

        if (sueldo>3000) {

            System.out.print("Debe abonar impuestos");

        } else {

            System.out.print("No paga impuestos");

        }

    }

}

public static void main(String[] ar) {

    EmpleadoFabrica empleado1;

    empleado1=new EmpleadoFabrica();
```

```
        empleado1.pagaImpuestos();  
    }  
}
```

```
import java.util.Scanner;  
  
public class OperacionesCalculo {  
    private Scanner teclado;  
    int valor1, valor2;  
  
    public OperacionesCalculo() {  
        teclado=new Scanner(System.in);  
        System.out.print("Ingrese primer valor:");  
        valor1=teclado.nextInt();  
        System.out.print("Ingrese segundo valor:");  
        valor2=teclado.nextInt();  
    }  
  
    public void sumar() {  
        int suma;  
        suma=valor1+valor2;  
        System.out.println("La suma es:"+suma);  
    }  
}
```



```
public void restar() {  
  
    int resta;  
  
    resta=valor1-valor2;  
  
    System.out.println("La resta es:"+resta);  
  
}  
  
public void multiplicar() {  
  
    int multiplicacion;  
  
    multiplicacion=valor1*valor2;  
  
    System.out.println("La multiplicación es:"+multiplicacion);  
  
}  
  
public void dividir() {  
  
    int division;  
  
    division=valor1/valor2;  
  
    System.out.println("La división es:"+division);  
  
}  
  
public static void main(String[] ar) {  
  
    OperacionesCalculo opera= new OperacionesCalculo();  
  
    opera.sumar();  
  
    opera.restar();  
  
    opera.multiplicar();  
  
    opera.dividir();  
  
}  
}
```

Clase String

La clase String está orientada a manejar cadenas de caracteres. Hasta este momento hemos utilizado algunos métodos de la clase String (equals, compareTo)

Ahora veremos otro conjunto de métodos de uso común de la clase String:

Métodos

boolean equals(String s1)

Como vimos el método equals retorna true si el contenido de caracteres del parámetro s1 es exactamente igual a la cadena de caracteres del objeto que llama al método equals.

boolean equalsIgnoreCase(String s1)

El funcionamiento es casi exactamente igual que el método equals con la diferencia que no tiene en cuenta mayúsculas y minúsculas (si comparamos 'Ana' y 'ana' luego el método equalsIgnoreCase retorna true)

int compareTo(String s1)

Este método retorna un 0 si el contenido de s1 es exactamente igual al String contenido por el objeto que llama al método compareTo. Retorna un valor >0 si el contenido del String que llama al método compareTo es mayor alfabéticamente al parámetro s1.

char charAt(int pos)

Retorna un caracter del String, llega al método la posición del caracter a extraer.

int length()

Retorna la cantidad de caracteres almacenados en el String.

String substring(int pos1,int pos2)

Retorna un substring a partir de la posición indicada en el parámetro pos1 hasta la posición pos2 sin incluir dicha posición.

int indexOf(String s1)

Retorna -1 si el String que le pasamos como parámetro no está contenida en la cadena del objeto que llama al método. En caso que se encuentre contenido el String s1 retorna la posición donde comienza a repetirse.

String toUpperCase()

- Retorna un String con el contenido convertido todo a mayúsculas.

String toLowerCase()

- Retorna un String con el contenido convertido todo a minúsculas.

Problema 1:

Confeccionar una clase que solicite el ingreso de dos String y luego emplee los métodos más comunes de la clase String.

Programa:

```
import java.util.Scanner;

public class Cadena1 {

    public static void main(String[] ar) {

        Scanner teclado=new Scanner(System.in);

        String cad1;

        String cad2;

        System.out.print("Ingrese la primer cadena:");

        cad1=teclado.nextLine();

        System.out.print("Ingrese la segunda cadena:");

        cad2=teclado.nextLine();

        if (cad1.equals(cad2)==true) {

            System.out.println(cad1+" es exactamente igual a "+cad2);

        } else {

            System.out.println(cad1+" no es exactamente igual a "+cad2);

        }

        if (cad1.equalsIgnoreCase(cad2)==true) {
```

```
        System.out.println(cad1+" es igual a "+cad2+" sin tener en cuenta  
mayúsculas/minúsculas");
```

```
    } else {
```

```
        System.out.println(cad1+" no es igual a "+cad2+" sin tener en cuenta  
mayúsculas/minúsculas");
```

```
    }
```

```
    if (cad1.compareTo(cad2)==0) {
```

```
        System.out.println(cad1+" es exactamente igual a "+cad2);
```

```
    } else {
```

```
        if (cad1.compareTo(cad2)>0) {
```

```
            System.out.println(cad1+" es mayor alfabéticamente que "+cad2);
```

```
        } else {
```

```
            System.out.println(cad2+" es mayor alfabéticamente que "+cad1);
```

```
        }
```

```
    }
```

```
    char carac1=cad1.charAt(0);
```

```
    System.out.println("El primer caracter de "+cad1+" es "+carac1);
```

```
    int largo=cad1.length();
```

```
    System.out.println("El largo del String "+cad1+" es "+largo);
```

```
    String cad3=cad1.substring(0,3);
```

```
    System.out.println("Los primeros tres caracteres de "+cad1+" son "+cad3);
```

```
    int posi=cad1.indexOf(cad2);
```

```
    if (posi== -1) {
```

```
        System.out.println(cad2+" no está contenido en "+cad1);
```

```
    } else {  
  
        System.out.println(cad2+" está contenido en "+cad1+" a partir de la posición  
"+posi);  
  
    }  
  
    System.out.println(cad1+ " convertido a mayúsculas es "+cad1.toUpperCase());  
  
    System.out.println(cad1+ " convertido a minúsculas es "+cad1.toLowerCase());  
  
    }  
}
```

Para cargar los dos String utilizamos en este caso el método `nextLine` para permitir ingresar espacios en blanco:

```
System.out.print("Ingrese la primer cadena:");  
  
cad1=teclado.nextLine();  
  
System.out.print("Ingrese la segunda cadena:");  
  
cad2=teclado.nextLine();
```

Problemas propuestos

1. Realizar una clase, que permita cargar una dirección de mail en el constructor, luego en otro método mostrar un mensaje si contiene el caracter '@'.
2. Cargar un String por teclado e implementar los siguientes métodos:
 - a) Imprimir la primera mitad de los caracteres de la cadena.
 - b) Imprimir el último caracter.
 - c) Imprimirlo en forma inversa.
 - d) Imprimir cada caracter del String separado con un guión.
 - e) Imprimir la cantidad de vocales almacenadas.
 - f) Implementar un método que verifique si la cadena se lee igual de izquierda a derecha tanto como de derecha a izquierda (ej. neuquen se lee igual en las dos direcciones)
3. Desarrollar un programa que solicite la carga de una clave. La clase debe tener dos métodos uno para la carga y otro que muestre si la clave es la correcta (la clave a comparar es "123abc")

4. Confeccionar un programa que permita cargar los nombres de 5 personas y sus mail, luego implementar los siguientes métodos:
 - a) Mostrar por pantalla los datos.
 - b) Consulta del mail ingresando su nombre.
 - c) Mostrar los mail que no tienen el carácter @.
5. Codifique un programa que permita cargar una oración por teclado, luego mostrar cada palabra ingresada en una línea distinta.

Por ejemplo si cargo:

La mañana está fría.

Debe aparecer:

La

mañana

está fría.

Solución

```
import java.util.Scanner;

public class Cadena2 {

    private Scanner teclado;

    private String mail;

    public Cadena2() {

        teclado=new Scanner(System.in);

        System.out.print("Ingrese un mail:");

        mail=teclado.nextLine();
    }

    public void verificarArroba() {

        boolean existe=false;

        for(int f=0;f<mail.length();f++) {

            if (mail.charAt(f)=='@') {

                existe=true;

            }

        }

        if (existe==true) {

            System.out.println(mail+" contiene el caracter @");

        } else {

            System.out.println(mail+" no contiene el caracter @");

        }

    }

}
```

```
    }  
}  
  
public static void main(String[] ar) {  
    Cadena2 cad=new Cadena2();  
    cad.verificarArroba();  
}  
}
```

```
import java.util.Scanner;  
  
public class Cadena3 {  
    private Scanner teclado;  
    private String cad;  
    public Cadena3() {  
        teclado=new Scanner(System.in);  
        System.out.print("Ingrese una cadena:");  
        cad=teclado.nextLine();  
    }  
}
```



```
public void primerMitad() {  
    String parte;  
    parte=cad.substring(0,cad.length()/2);  
    System.out.println("Primer mitad de caracteres:"+parte);  
}
```

```
public void ultimoCaracter() {  
    char ultimo=cad.charAt(cad.length()-1);  
    System.out.println("Ultimo caracter:"+ultimo);  
}
```

```
public void formaInversa() {  
    System.out.println("Impresión en forma inversa:");  
    for(int f=cad.length()-1;f>=0;f--) {  
        System.out.print(cad.charAt(f));  
    }  
    System.out.println();  
}
```

```
public void conGuion() {  
    System.out.println("Separado por guiones:");  
    for(int f=0;f<cad.length();f++) {  
        System.out.print(cad.charAt(f)+"-");  
    }
```

```
}  
  
System.out.println();  
  
}
```

```
public void cantidadVocales() {  
  
    int cant=0;  
  
    for(int f=0;f<cad.length();f++) {  
  
        if (cad.charAt(f)=='a' || cad.charAt(f)=='e' ||  
            cad.charAt(f)=='i' || cad.charAt(f)=='o' ||  
            cad.charAt(f)=='u' || cad.charAt(f)=='A' ||  
            cad.charAt(f)=='E' || cad.charAt(f)=='I' ||  
            cad.charAt(f)=='O' || cad.charAt(f)=='U') {  
  
            cant++;  
  
        }  
  
    }  
  
    System.out.println("Cantidad de vocales:"+cant);  
  
}
```

```
public void esCapicua() {  
  
    int cant=0;  
  
    for(int f=0;f<cad.length()/2;f++) {  
  
        if (cad.charAt(f)==cad.charAt(cad.length()-1-f)) {  
  
            cant++;  
  
        }  
  
    }  
  
}
```

```
    }  
}  
  
if (cant==cad.length()/2) {  
    System.out.println("Es capicúa la cadena "+cad);  
}  
else {  
    System.out.println("No es capicúa la cadena "+cad);  
}  
}  
  
public static void main(String[] ar) {  
    Cadena3 cad=new Cadena3();  
    cad.primerMitad();  
    cad.ultimoCaracter();  
    cad.formaInversa();  
    cad.conGuion();  
    cad.cantidadVocales();  
    cad.esCapicua();  
}  
}
```

```
import java.util.Scanner;

public class Cadena4 {

    private Scanner teclado;

    private String clave;

    public Cadena4() {

        teclado=new Scanner(System.in);

        System.out.print("Ingrese clave:");

        clave=teclado.nextLine();

    }

    public void verificarClave() {

        if (clave.equals("123abc")==true) {

            System.out.println("Se ingresó la clave en forma correcta");

        } else {

            System.out.println("No se ingresó la clave en forma correcta");

        }

    }

}

public static void main(String[] ar) {

    Cadena4 cad=new Cadena4();

    cad.verificarClave();

}
```

```
}  
  
}
```

```
import java.util.Scanner;  
  
public class Cadena5 {  
  
    private Scanner teclado;  
  
    private String[] nombres;  
  
    private String[] mail;  
  
    public Cadena5() {  
  
        teclado=new Scanner(System.in);  
  
        nombres=new String[5];  
  
        mail=new String[5];  
  
        for(int f=0;f<nombres.length;f++) {  
  
            System.out.print("Ingrese nombre:");  
  
            nombres[f]=teclado.nextLine();  
  
            System.out.print("Ingrese mail");  
  
            mail[f]=teclado.nextLine();  
  
        }  
    }  
}
```

```
}
```

```
public void listar() {
```

```
    for(int f=0;f<nombres.length;f++) {
```

```
        System.out.println(nombres[f]+" - "+mail[f]);
```

```
    }
```

```
}
```

```
public void consultaMail() {
```

```
    String aux;
```

```
    System.out.print("Ingrese el nombre de la persona:");
```

```
    aux=teclado.nextLine();
```

```
    boolean existe=false;
```

```
    for(int f=0;f<nombres.length;f++) {
```

```
        if (aux.equals(nombres[f])) {
```

```
            System.out.println("Mail de la persona:"+mail[f]);
```

```
            existe=true;
```

```
        }
```

```
    }
```

```
    if (existe==false) {
```

```
        System.out.println("No existe una persona con ese nombre.");
```

```
    }
```

```
}
```

```
public void sinArroba() {  
  
    for(int f=0;f<mail.length;f++) {  
  
        boolean tiene=false;  
  
        for(int k=0;k<mail[f].length();k++) {  
  
            if (mail[f].charAt(k)=='@') {  
  
                tiene=true;  
  
            }  
  
        }  
  
        if (tiene==false) {  
  
            System.out.println(mail[f]+" no tiene @");  
  
        }  
  
    }  
  
}  
  
public static void main(String[] ar) {  
  
    Cadena5 cad=new Cadena5();  
  
    cad.listar();  
  
    cad.consultaMail();  
  
    cad.sinArroba();  
  
}  
  
}
```

```
import java.util.Scanner;

public class Cadena6 {

    private Scanner teclado;

    private String oracion;

    public Cadena6() {

        teclado=new Scanner(System.in);

        System.out.print("Ingrese oración:");

        oracion=teclado.nextLine();

    }

    public void imprimir() {

        for(int f=0;f<oracion.length();f++) {

            if (oracion.charAt(f)==' ') {

                System.out.println();

            } else {

                System.out.print(oracion.charAt(f));

            }

        }

    }

}
```



```
}
```

```
public static void main(String[] ar) {
```


```
    Cadena6 cad=new Cadena6();
```

```
    cad.imprimir();
```

```
}
```

```
}
```





Muchas gracias hasta la próxima clase.

Alsina 16 [B1642FNB] San Isidro | Pcia. De Buenos Aires |Argentina |

TEL.: [011] 4742-1532 o [011] 4742-1665 |

www.institutosanisidro.com.ar info@institutosanisidro.com.ar