



17 DE ABRIL DE 2020

UTILIDADES API JAVA

RODRIGO HERRANZ



Ejercicio 1:

Partiendo de la cadena de caracteres “Me gusta programar en Java.”, aplicar los métodos expuestos más arriba para:

- A. Obtener la longitud de la cadena de caracteres

```
public static void main(String[] args) {  
  
    String cadena = "Me gusta programar en Java.";  
  
    System.out.println(cadena.length());  
  
}
```

- B. Obtener carácter que ocupa posición 5ª

```
public static void main(String[] args) {  
  
    String cadena = "Me gusta programar en Java.";  
  
    System.out.println(cadena.charAt(4));  
  
}
```

- C. Obtener la sub-cadena desde la posición 21 hasta el final

```
public static void main(String[] args) {  
  
    String cadena = "Me gusta programar en Java.";  
  
    System.out.println(cadena.substring(21));  
  
}
```

- D. Obtener la sub-cadena indicada por los índices 8 y 16

```
public static void main(String[] args) {  
  
    String cadena = "Me gusta programar en Java.";  
  
    System.out.println(cadena.substring(8,16));  
  
}
```

- E. Muestra por pantalla el resultado de concatenar la cadena “casi siempre”

```
public static void main(String[] args) {  
  
    String cadena = "Me gusta programar en Java.";  
  
    System.out.println(cadena.concat(" Casi siempre"));  
  
}
```

- F. En la nueva cadena de caracteres obtenida en E. recuperar la posición de la cadena “casi”.

```

public static void main(String[] args) {

    String cadena = "Me gusta programar en Java.";

    String cadenaDos = cadena.concat(" Casi siempre");

    System.out.println(cadenaDos.indexOf("Casi"));

}

```

- G. *Muestra por pantalla la cadena de caracteres original en mayúscula*

```

public static void main(String[] args) {

    String cadena = "Me gusta programar en Java.";

    String cadenaDos = cadena.concat(" Casi siempre");

    System.out.println(cadena.toUpperCase());

}

```

- H. *Compara la cadena de caracteres original con la cadena de caracteres en mayúsculas. Primero teniendo en cuenta la diferencia entre mayúsculas y minúsculas y por último ignorando esta diferencia*

```

public static void main(String[] args) {

    String cadena = "Me gusta programar en Java.";

    String cadenaMayus = cadena.toUpperCase();

    cadena.compareTo(cadenaMayus);

    cadena.compareToIgnoreCase(cadenaMayus);

}

```

Ejercicio 2:

Desarrollar una función que haga el reverso de una cadena de caracteres que se suministre por parámetro. Investiga que función de la API String (no explicada aquí) podría ser más útil para tal cometido.

```

public class EjercicioDos {

    public static String darVuelta(String a) {
        String cadena = a;

        String cadenaVuelta = "";

        int longitud = cadena.length() - 1;

        for(int i=longitud;i>=0;i--) {

            cadenaVuelta = cadenaVuelta+cadena.charAt(i);
        }
        return cadenaVuelta;
    }

    public static void main(String[] args) {

        System.out.println(darVuelta("Esto es una cadena de prueba"));
    }
}

```

Ejercicio 3:

Trabajando con el API de `StringBuilder` y partiendo de la cadena de caracteres “Hello Java world.”, realizar las siguientes operaciones:

1. *Eliminar la cadena de caracteres “Java” y mostrar la cadena de caracteres del `StringBuilder` resultante.*

```

public class Ejercicio3 {

    public static void main(String[] args) {

        StringBuilder cadena = new StringBuilder ("Hello Java world.");

        System.out.println(cadena.delete(5, 10));
    }
}

```

2. *Sobre el `StringBuilder` resultante del apartado anterior insertar la cadena “my ” en la posición 6. Mostrar la cadena de caracteres resultante.*

```

public class Ejercicio3 {

    public static void main(String[] args) {

        StringBuilder cadena = new StringBuilder ("Hello Java world.");

        StringBuilder cadena2 = new StringBuilder();

        cadena2 = cadena.delete(5, 10);

        System.out.println(cadena2.insert(6, new StringBuilder("my ")));
    }
}

```

3. *Añadir al final de la cadena de caracteres del StringBuilder resultante del apartado 2 la cadena de caracteres: “ I am xxxx.”. Donde xxxx es mi nombre de pila. Mostrar la cadena de caracteres resultante.*

```
public class Ejercicio3 {  
  
    public static void main(String[] args) {  
  
        StringBuilder cadena = new StringBuilder ("Hello Java world.");  
  
        StringBuilder cadena2 = new StringBuilder();  
  
        cadena2 = cadena.delete(5, 10);  
  
        StringBuilder cadena3 = new StringBuilder(cadena2.insert(6, new StringBuilder("my ")));  
  
        System.out.println(cadena3.insert(cadena3.length(), " I am Rodrigo"));  
    }  
  
}  
  
Hello my world. I am Rodrigo
```

4. *En el StringBuilder del apartado anterior reemplazar vuestro nombre de pila por vuestro primer apellido.*

```
public class Ejercicio3 {  
  
    public static void main(String[] args) {  
  
        StringBuilder cadena = new StringBuilder ("Hello Java world.");  
  
        StringBuilder cadena2 = new StringBuilder();  
  
        cadena2 = cadena.delete(5, 10);  
  
        StringBuilder cadena3 = new StringBuilder(cadena2.insert(6, new StringBuilder("my ")));  
  
        StringBuilder cadena4 = new StringBuilder(cadena3.insert(cadena3.length(), " I am Rodrigo"));  
  
        System.out.println(cadena3.replace(21, cadena3.length(), "Herranz"));  
    }  
  
}
```

5. *Obtener la subcadena entre las posiciones 0 y 5, creando un nuevo StringBuilder con ella. Hacer el reverso de este nuevo substring y añadela de nuevo al StringBuilder del apartado 4 a partir de la posición 0. Mostrar la cadena de caracteres resultante.*

```
public class Ejercicio3 {  
  
    public static void main(String[] args) {  
  
        StringBuilder cadena = new StringBuilder ("Hello Java world. I am Herranz");  
  
        StringBuilder cadena2 = new StringBuilder (cadena.substring(0, 5));  
  
        cadena2 = cadena2.reverse();  
  
        cadena = cadena.insert(0, cadena2);  
  
        System.out.println(cadena);  
    }  
  
}
```

```
olleHHello Java world. I am Herranz
```

Ejercicio 4.1:

Visita la API SimpleDateFormat y haz un pequeño resumen de los patrones de los formatos de las fechas.

¿Qué se muestra por pantalla con un objeto Date sin formato?

Se muestra la fecha del día actual

Ejercicio 4.2:

¿Qué pasa si el contenido de la cadena de caracteres no es una fecha o no coincide con el formato especificado? ¿Cómo solucionarlo? Implementa la solución para el código del ejemplo 2.

Salta una excepción. La forma de solucionarlo es utilizando un try catch

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Ejercicio4 {

    public static void main(String[] args) throws ParseException {

        SimpleDateFormat pdf = new SimpleDateFormat("dd/M/yyyy hh:mm:ss");

        try {

            String dateInString = "16-18-2020 20:5:30";

            Date date = pdf.parse(dateInString);

            System.out.println(date);
        } catch (Exception e) {

            System.out.println("La fecha no esta escrita correctamente");
        }
    }
}
```

Ejercicio 4.3:

Modifica el código del ejemplo 3 para mostrar sólo la fecha. Ahora muestra sólo la hora.

```

import java.text.SimpleDateFormat;
import java.util.Date;

public class Ejercicio4_3 {

    public static void main(String[] args) {

        SimpleDateFormat dateFormat = new SimpleDateFormat ("dd/M/yyyy");

        Date date = new Date();

        System.out.println(dateFormat.format(date));

    }

}

import java.text.SimpleDateFormat;
import java.util.Date;

public class Ejercicio4_3 {

    public static void main(String[] args) {

        SimpleDateFormat dateFormat = new SimpleDateFormat ("hh:mm:ss");

        Date date = new Date();

        System.out.println(dateFormat.format(date));

    }

}

```

Ejercicio 4.4:

Comprueba la salida del código anterior en caso de que el `GregorianCalendar` calendar se crease sin parámetros, es decir, si se usa el constructor por defecto de `GregorianCalendar` para la variable `calendar`.

La salida por defecto de `GregorianCalendar` es la fecha que tiene el ordenador

Ejercicio 5:

Dada la cadena de caracteres `sFechas`, en el que se guardan fechas separadas por comas con el formato días/mes/año, generar un array `vFechas` en el que cada posición del array será un objeto de tipo `Date` con formato yyyy-MM-dd, con todas las fechas de `sFechas`. Dicho array deberá ser ordenado de manera ascendente.

```

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class EjercicioCinco {

    public static void main(String[] args) throws ParseException{

        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

        SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy/MM/dd");

        String sFechas = new String("21/04/2015, 12/06/1955, 26/01/1985,"
            + " 18/06/1936, 12/10/1492, 25/12/2000, 02/05/2018,13/10/1975,"
            + " 01/03/1991, 20/10/1972, 17/04/2020");

        int aux1 = 0;

        String [] fechasSeparadas = sFechas.split(",");

        String [] fechasFragmentadas= new String[(fechasSeparadas.length)*3];

        for(int i=0;i<fechasSeparadas.length;i++) {

            fechasSeparadas[i] = fechasSeparadas[i].trim();

            for(int j=aux1;j<aux1+1;j++) {

                fechasFragmentadas[j]=fechasSeparadas[i].substring(0, 2);

                fechasFragmentadas[j+1]=fechasSeparadas[i].substring(3, 5);

                fechasFragmentadas[j+2]=fechasSeparadas[i].substring(6);

            }
            aux1=aux1+3;
        }

        aux1 = 2;

        String[] fechasColocadas = new String[fechasSeparadas.length];

        Date [] vFechas = new Date[fechasSeparadas.length];
    }
}

```



```

        for(int i=0;i<fechasSeparadas.length;i++) {

            for(int j=aux1;j<aux1+1;j++) {

                fechasColocadas[i]= fechasFragmentadas[j] + "/" +
                fechasFragmentadas[j-1] + "/" + fechasFragmentadas[j-2];
            }

            aux1=aux1+3;
        }

        for(int i=0;i<vFechas.length;i++){

            vFechas[i] = sdf.parse(fechasColocadas[i]);
        }

        Date ayuda = sdf.parse("958/1/1");

        for(int i=0;i<vFechas.length;i++) {
            for(int j=0;j<vFechas.length;j++) {

                if(vFechas[i].before(vFechas[j])) {

                    ayuda = vFechas[i];

                    vFechas[i] = vFechas[j];

                    vFechas[j] = ayuda;
                }
            }
        }

        for(int i=0;i<vFechas.length;i++) {

            System.out.println(vFechas[i]);
        }

    }
}

```

```

Fri Oct 12 00:00:00 CET 1492
Thu Jun 18 00:00:00 CEST 1936
Sun Jun 12 00:00:00 CET 1955
Fri Oct 20 00:00:00 CET 1972
Mon Oct 13 00:00:00 CET 1975
Sat Jan 26 00:00:00 CET 1985
Fri Mar 01 00:00:00 CET 1991
Mon Dec 25 00:00:00 CET 2000
Tue Apr 21 00:00:00 CEST 2015
Wed May 02 00:00:00 CEST 2018
Fri Apr 17 00:00:00 CEST 2020

```