

VALIDADOR DE ISBN

PÉREZ SELLERS, ALEJANDRO
PROGRAMACION 1º Daw

1. Introducción a la práctica.

En esta practica he desarrollado un programa en java que me permite comprobar si un numero ISBN de 10 dígitos es valido y además, reparar aquellos que tengan un dígito faltante indicado con el símbolo interrogante.

El ISBN es un código único que identifica libros, y para saber si es correcto se realiza un cálculo sencillo: multiplico cada digito por un numero que va de 10 a 1, sumo los resultados, y si el total es multiplico de 11, el isbn es válido. Si el resultado indica que el ultimo digito debería ser 10, utilizo la letra x.

Con mi programa no solo compruebo si un ISBN es correcto, sino que también calculo digito que falta cuando un ¿, lo que permite corregir errores.

EJEMPLO. El '0201103311' es un *ISBN* válido porque:

$$10*0 + 9*2 + 8*0 + 7*1 + 6*1 + 5*0 + 4*3 + 3*3 + 2*1 + 1*1 = 55$$

y 55 es múltiplo de 11 ($55 = 11*5$).

Otro ejemplo de número *ISBN* válido es '156881111X':

$$10*1 + 9*5 + 8*6 + 7*8 + 6*8 + 5*1 + 4*1 + 3*1 + 2*1 + 1*10 = 231$$

y 231 es múltiplo de 11 ($231 = 11*21$).

Ejemplo de funcionamiento:

```
Introduzca ISBN:
0201103311
El ISBN es válido.

Process finished with exit code 0
```

```
Introduzca ISBN:
0201103411
El ISBN no es válido.

Process finished with exit code 0
```

Añade una opción al programa anterior que permita reparar números *ISBN* cuando detecte que le falta un dígito (marcado con el símbolo '?'). En ese caso, el programa debe imprimir el valor correcto del dígito que falta.

Ejemplo de funcionamiento

```
Introduzca ISBN:
15688?111X
El digito que falta es 1

Process finished with exit code 0
```

2. Explicación del Código.

Para que el usuario puede elegir de manera fácil la opción que desea, he creado un menú al principio del programa con tres opciones:

- 1- Validar isbn.
- 2- Reparar un isbn.
- 3- Salir del programa.

Cuando el usuario elige una opción el programa pide datos y ejecuta ese bloque.

```
public class practica_validacion_isbn_plus {  ▲ AlexPerse10 *
    public static void main(String[] args) {  ▲ AlexPerse10 *
        Scanner sc = new Scanner(System.in);
        int suma = 0; // aqui guardo el total de la acumulacion de las multiplicaciones
        int multiplicacion; // multiplico la posicion por la cifra;

        String numero;
        System.out.println("** VALIDADOR ISBN **"); //imprimo menu con las opciones
        System.out.println("1-VALIDAR ISBN");
        System.out.println("2-REPARAR ISBN ");
        System.out.println("X-SALIR");
        String opcion = sc.nextLine();
```

Validar un ISBN

Si el usuario selecciona validar, el programa le pide que introduzca un isbn.

Primero comprueba que tenga exactamente 10 caracteres, porque si no, ya sabemos que no puede ser valido.

Luego, recorro cada cifra y hago la suma del isbn, multiplico la primera cifra por 10, la segunda por 9, y así hasta la última. Si la cifra es x equivale a 10. Al final sumo todos los resultados y compruebo si la suma es divisible entre 11. Si lo es es válido, sino no lo es, es invalido. Se muestra la multiplicación.

```
switch (opcion) { // se eligen los bloques con el switch
    case "1":
        System.out.println("introduce isbn"); // se introduce el numero isbn por teclado
        numero = sc.nextLine();
        int numero_longitud = numero.length(); // cuenta cantidad de caracteres que tiene el numero
        if (numero_longitud != 10) { // si numero no tiene 10 caracteres es invalido
            System.out.println("el isbn es invalido");
            break; // sale del bloque
        }
        for (int i = 0; i < 10; i++) { // se realiza un for para recorrer los numeros del 0 al 9
            char c = numero.charAt(i); // obtengo el caracter en la posicion i
            int cifra; // cuando el valor de ese caracter

            if (c == 'x') { // si se introduce el valor x sera un 10
                cifra = 10;
            } else { // sino se introduce un valor x la cifra cogera el valor de c
                cifra = Integer.parseInt(String.valueOf(c));
            } // aqui convierto ese caracter a un string y despues tengo que transformar ese string a un numero para poder multiplicarlo

            int posicion = 10 - i; // Se calcula la posicion empezando de mayor a menor
            multiplicacion = cifra * posicion; // multiplico cifra por posicion
            suma = suma + multiplicacion; // se guarda el resultado en la variable suma que va incrementando
            System.out.println(cifra + "*" + posicion + "=" + multiplicacion); // va mostrando la multiplicacion
        }
        System.out.println("el resultado de la suma: " + suma); // muestra la suma total
        if (suma % 11 == 0) { // si la suma total es divisible entre 11 es valido, se muestra el mensaje
            System.out.println("el isbn es valido");
        } else {
            System.out.println("el isbn es invalido"); // si no es divisible entre 11 es invalido
        }
    }
}
```

Para conocer si el digito tiene 10 cifras utilizo el length, donde calcula la cantidad de cifras que tiene y introduzco un condicional donde la longitud del numero es diferente a 10 salta un mensaje de numero invalido. Si es correcto pasamos al bucle for donde recorre todos los números, se declara la variable char c = numero.charAt(i) donde obtengo cada digito, también declaro la variable cifra donde se guarda cada digito para multiplicarlo.

Paso al siguiente bloque condicional donde si aparece x es igual a 10 sino convierto el carácter c de char a string y de string a int.

Posteriormente se realiza la multiplicación y la suma y se verifica con el modulo si el numero isbn es correcto.

Reparar isbn.

En el caso 2 se introduce un numero el cual se quiere reparar, de igual modo se verifica que el numero no sea ni mayor ni menor que 10, se busca el interrogante con `int posicioninterrogante = numero_reparar.indexOf('?')`; si no se encuentra el valor de `int posicioninterrogante` es -1, si se encuentra el interrogante el valor de esa variable cambia al numero de la posición.

Posteriormente introduzco un for para probar todos los posibles números que podría tener el dígito faltante siendo interrogante, donde va desde cero a diez. También se crea una variable `int suma_prueba = 0`;

En el siguiente bucle for, recorre cada dígito del isbn, después guarda en `c` el carácter de la posición `i` y `cifra` será el valor numérico de ese carácter para después hacer la suma.

En el siguiente condicional se dice que si estamos en la posición `i == posicioninterrogante` usamos el número que estamos probando (`cifras_posibles`) si el carácter es `x` lo tomamos como 10, si es cualquier otro número lo convertimos de `char` a `int`.

Igual que en el bloque anterior se comprueba que sea divisible entre 11 y si lo es salta un mensaje donde te informa del número que falta.

```
case "2":
    System.out.println("Introduce el isbn que quieres reparar:"); // aqui introduzco el isbn que quiero reparar
    String numero_reparar = sc.nextLine();
    int longitud_numero = numero_reparar.length(); // se calcula la cantidad de caracteres que tiene el numero.
    if (longitud_numero != 10) { // Si el numero tiene un longitud diferente a 10 caracteres es invalido
        System.out.println("El numero ISBN es invalido");
        break;
    }

    // Uso indexOf para encontrar el '?'
    int posicioninterrogante = numero_reparar.indexOf('?');
    if (posicioninterrogante == -1) { // si no hay '?'
        System.out.println("No hay ningún signo '?' en el ISBN");
        break;
    }

    for (int cifras_posibles = 0; cifras_posibles <= 10; cifras_posibles++) { // tengo que probar posibles cifras que pueda tener la cifra faltante
        int suma_prueba = 0; // variable donde realiza la suma de cada prueba

        for (int i = 0; i < 10; i++) { // aqui vuelve a recorrer todos los numeros
            char c = numero_reparar.charAt(i);
            int cifra;
            if (i == posicioninterrogante) { // si estoy en la posicion de ? se usa el valor cifras_posibles
                cifra = cifras_posibles;
            } else if (c == 'x') { // si es x es igual a 10
                cifra = 10;
            }
        }
    }
}
```

```

    } else if (c == 'x') { // si es x es igual a 10
        cifra = 10;
    } else {
        cifra = Integer.parseInt(String.valueOf(c)); // el caracter se debe de pasar a string y luego a int
    }
    int posicion = 10 - i; // se calcula posicion empezando de mayor a menor
    suma_prueba = suma_prueba + (cifra * posicion); // se realiza la suma y multiplicacion
}

if (suma_prueba % 11 == 0) { // se verifica que sea divisible entre 11
    if (cifras_posibles == 10) {
        System.out.println("El digito faltante es: X"); // si la cifra_posible es 10 el caracter faltante es una x
    } else {
        System.out.println("El digito faltante es: " + cifras_posibles);
    }
    break;
}
}
break;

case "x":
    System.out.println("Saliendo del programa...");
    break;

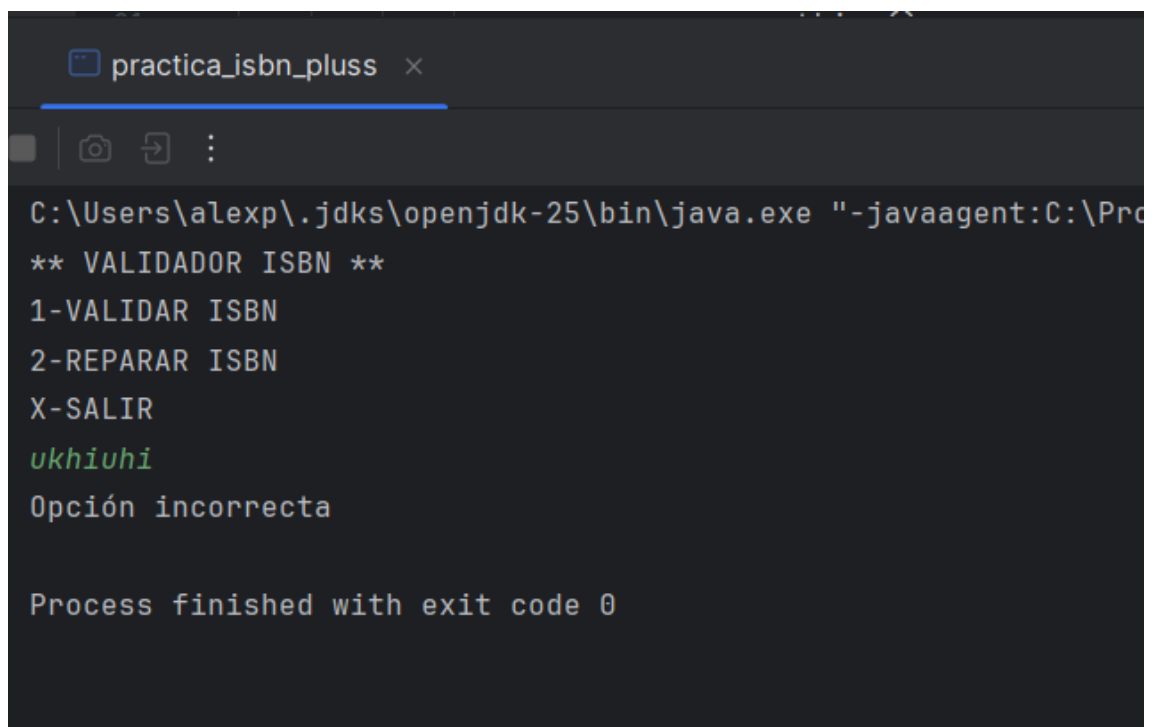
default:
    System.out.println("Opción incorrecta");
}

```

Por último, la tercera opción se sale del programa, aparece un mensaje donde informa que se esta saliendo, si por algún casual se introdujese mal la opción saltaría un mensaje de opción incorrecta.

3. Pruebas de ejecución.

- Se introduce caracteres diversos en la selección de menú



```

practica_isbn_pluss x
C:\Users\alexp\.jdk\openjdk-25\bin\java.exe -javaagent:C:\Pro...
** VALIDADOR ISBN **
1-VALIDAR ISBN
2-REPARAR ISBN
X-SALIR
ukhiuhi
Opción incorrecta

Process finished with exit code 0

```

- Se introduce números negativos en opción 1

```
C:\Users\alexp\.jdk\openjdk-25\bin\java.exe "-javaagent
** VALIDADOR ISBN **
1-VALIDAR ISBN
2-REPARAR ISBN
X-SALIR
1
introduce isbn
-4512365241
el isbn es invalido

Process finished with exit code 0
```

- Se introduce número de mas de 10 cifras

```
C:\Users\alexp\.jdk\openjdk-25\bin\java.exe "-jav
** VALIDADOR ISBN **
1-VALIDAR ISBN
2-REPARAR ISBN
X-SALIR
1
introduce isbn
16818614864864896489
el isbn es invalido

Process finished with exit code 0
|
```

- Se introduce en opción 1 numero negativo

```
C:\Users\alexp\.jdk\openjdk-25\bin\java.exe "-javaagent
** VALIDADOR ISBN **
1-VALIDAR ISBN
2-REPARAR ISBN
X-SALIR
1
introduce isbn
-4512365241
el isbn es invalido

Process finished with exit code 0
```


- Se introduce en opción 2 número con interrogantes

```
C:\Users\alexp\.jdk\openjdk-25\bin\java.exe "-javaage
** VALIDADOR ISBN **
1-VALIDAR ISBN
2-REPARAR ISBN
X-SALIR
2
Introduce el isbn que quieres reparar:
12????????
Error valores invalidos

Process finished with exit code 0
```

- Se introduce en opción 2 todo interrogantes

```
C:\Users\alexp\.jdk\openjdk-25\bin\java.exe "-javaag
** VALIDADOR ISBN **
1-VALIDAR ISBN
2-REPARAR ISBN
X-SALIR
2
Introduce el isbn que quieres reparar:
????????
Error valores invalidos

Process finished with exit code 0
```

- Se introduce en opción 2 letras

```
C:\Users\alexp\.jdk\openjdk-25\bin\java.exe "-javaage
** VALIDADOR ISBN **
1-VALIDAR ISBN
2-REPARAR ISBN
X-SALIR
2
Introduce el isbn que quieres reparar:
fvineriovjioerj
El numero ISBN es invalido

Process finished with exit code 0
```

- Se introduce numero en negativo

```
C:\Users\alexp\.jdk\openjdk-25\bin\java.exe "-javaagent:C:\P
** VALIDADOR ISBN **
1-VALIDAR ISBN
2-REPARAR ISBN
X-SALIR
2
Introduce el isbn que quieres reparar:
-7845698?23
El numero ISBN es invalido

Process finished with exit code 0
```

