

Práctica 7

Objetivos:

- Usar instrucciones condicionales múltiples, entrada/salida y funciones para manejo de Strings

Descripción:

- Escribir una clase Java llamada **Alumnos** que permita obtener cifras de los alumnos matriculados en una universidad, a partir de datos expresados inicialmente mediante texto
- Podrá haber alumnos de los siguientes ciclos:
 - grado
 - master
 - doctorado
- Para cada ciclo se guarda la suma de todos sus alumnos
 - en total, 3 atributos

Práctica 7: Diseño

Métodos de la clase `Alumnos`

- *constructor*: Pone todos los atributos a cero
- *inserta*: acumula los datos de una asignatura en los atributos
 - Recibe como parámetro un texto que contiene el ciclo, el número de alumnos y el nombre de la asignatura separados por un espacio en blanco
 - Ejemplos de parámetros de entrada a este método:
grado 63 Introducción al software
master 18 Desarrollo de software
doctorado 4 Ciencia y tecnología
 - Añade el número de alumnos a la suma de su ciclo
 - Observar que no se hace nada con el nombre de la asignatura
 - Para facilitar el método, no es preciso comprobar si en el texto de entrada hay errores

Práctica 7: Diseño

Otros métodos de la clase `Alumnos` :

- `numAlumnos`: retorna el número de alumnos del ciclo indicado
 - recibe como parámetro un texto que indica el ciclo que se desea ("grado", "master", o "doctorado")

Práctica 7. Programa principal

Escribir además un programa principal en una clase aparte, que lea los datos de varias asignaturas de un objeto de la clase `CajaTexto` (paquete `fundamentos`) y que sirva para probar la clase `Alumnos`

Se ejecutará el siguiente algoritmo descrito mediante pseudocódigo:

```
Alumnos al = nuevo objeto de la clase Alumnos
CajaTexto caja = nuevo objeto de la clase CajaTexto
caja.espera()
mientras caja.hayMas() hacer
    String linea = caja.leeString()
    caja.avanzaLinea()
    al.inserta(linea)
fin mientras
muestra en pantalla al.numAlumnos("grado")
muestra en pantalla al.numAlumnos("máster")
muestra en pantalla al.numAlumnos("doctorado")
```

Práctica 7: Realización y entrega

Realización:

- Escribir la clase `Alumnos` y probar sus métodos para los diferentes ciclos
- Escribir el programa principal y probarlo, comprobando que los valores obtenidos son los correctos

Entregar:

- Un diagrama de la clase `Alumnos`
- El código Java desarrollado
- Los resultados del programa principal

Práctica 7: Notas sobre los strings

Nota: Disponemos de los siguientes métodos de la clase `String`, teniendo en cuenta que los caracteres se numeran empezando por cero

Operación	Descripción
<code>String substring (int ini, int fin)</code>	Retorna el trozo del string que está comprendido entre los caracteres ini (incluido) a fin (excluido)
<code>int length()</code>	Retorna el número de caracteres
<code>int indexOf(char c)</code>	Buscar el carácter c en el String; si lo encuentra retorna la posición en que está; si no, retorna -1
<code>boolean equals(String s)</code>	Compara el string con s; retorna true si sus contenidos son iguales y false si no
<code>String trim()</code>	Retorna una copia del string original omitiendo los espacios en blanco iniciales y finales

Práctica 7: Notas sobre los strings

Nota: Obsérvese que para comparar dos strings no debe usarse el operador de igualdad:

```
if (s1==s2) ... // mal (aunque compila)
if (s1.equals(s2)) ... // bien
```

Nota: Para convertir un texto a número real o entero usar respectivamente estos métodos de las clases `Double` e `Integer`

Operación	Descripción
<code>static double parseDouble(String s)</code>	Convierte el texto <code>s</code> que se le pasa como parámetro a número real y lo retorna
<code>static int parseInt(String s)</code>	Convierte el texto <code>s</code> que se le pasa como parámetro a número entero y lo retorna

Ejemplos de conversión de texto a número

Por ejemplo, si `s1` es un String:

```
double p=Double.parseDouble(s1);  
int    n=Integer.parseInt(s1);
```


Práctica 7: Parte avanzada

Descripción: Modificar el método `inserta()` para tener en cuenta los siguientes errores en los que el parámetro tiene un formato diferente del esperado:

- a) tiene más de un espacio en blanco entre el ciclo y el número de alumnos
 - el resultado debe ser como si sólo hubiera un único espacio
- b) no tiene ningún espacio en blanco o solo hay uno
 - incrementar un nuevo atributo que cuenta el número de errores detectados
- c) tiene algún espacio en blanco antes del ciclo
 - ignorar esos espacios
- d) el ciclo no es uno de los tres esperados
 - incrementar el atributo que cuenta el número de errores detectados

Práctica 7: Parte avanzada

Además:

- añadir a `Alumnos` un método observador del número de errores
- modificar el `main` para que muestre al final el número de errores
- hacer pruebas que muestren el funcionamiento ante estos errores