Programación
ESTRUCTURAS DE CONTROL

Manuel Molino Milla Luis Molina Garzón

23 de noviembre de 2014

Índice

1. API de Java

1.1. Ejercicio 1

La API String de Java, contiene entre otros, los métodos abajo indicados en la tabla. Rellena la información correspondiente:

nombre método	valor retorno	parámetros	Breve descripción	Ejemplo
contains				
endsWith				
equalsIgnoreCase				
isEmpty				
startsWith				

2. Estructuras de control

2.1. Ejercicio 2

Utilizando solo el método *main*, es decir sin crear un constructor para objetos, ni atributos ni metodos asociados, realiza el siguiente codigo:

- Un programa que imprima los números del 1 al 10.
- Un programa que imprima la siguiente serie: 20 25 30 35 ... 70 75 80
- Un programa que imprima la siguiente serie: 100 98 96 94 ... 56 54 52 50
- Un programa que calcule la suma de los números enteros del 1 al 100.
- Un programa que calcule la la suma de los cuadrados de los 15 primeros números naturales.
- Un programa que calcule independientemente la suma de los pares e impares comprendidos entre 1 y 50.
- Un programa que dado un número entero positivo, reporte su tabla de multiplicar.

Realiza el ejercicio usando una vez el bucle for, otra vez el bucle while y también debes usar la estructura do-while

2.2. Ejercicio 3

Crea una clase Numeros, con atributo valor del digito, crea los correspondientes *getters y setters* y crea los siguientes metodos adicionales (no hace falta constructor):

- Un metodo que imprima todos los numeros que vayan desde cero a ese numero (inclusive).
- Un metodo que no indique si ese numero es bien divisible por dos, o por cinco, o por diez o por ningun numero de los anteriores.
- Un método que indique si ese número es mayor o menor que cien.
- Un método que solo acepte numeros inferiores a cien y nos de todos los divisores de ese número (que los imprima por pantalla)

Crea una clase denominada TestNumero dondes puedas crear objetos *Numero* en el correpondiente método *main*, que es el primero que se inicia cuando se corre un programa en Java.

2.3. Ejercicio 4

En español hay dos maneras para formar el plural de los sustantivos y adjetivos: -s y -es. Las reglas en las que se basan son:

- **Sustantivos y adjetivos terminados en vocal átona o en -e tónica**. Forman el plural con -s: *casas, estudiantes, taxis, planos, tribus, comités.*
- **Sustantivos y adjetivos terminados en -a o en -o tónicas**.Forman el plural con -s: *papás, sofás, bajás, burós, rococós, dominós.*
- **Sustantivos y adjetivos terminados en -i o en -u tónicas**. Admiten generalmente dos formas de plural, una con -es y otra con -s, aunque en la lengua culta suele preferirse la primera: *bisturíes o bisturís, carmesíes o carmesís, tisúes o tisús, tabúes o tabús.*
- **Sustantivos y adjetivos terminados en -y precedida de vocal**. Forman tradicionalmente su plural con -es: *rey, pl. reyes; ley, pl. leyes; buey, pl. bueyes; ay, pl. ayes; convoy, pl. convoyes; bocoy, pl. bocoyes*
- **Sustantivos y adjetivos terminados en -s o en -x.** Permanecen invariables: *crisis, pl. crisis; tórax, pl. tórax; fórceps, pl. fórceps.*
- **Sustantivos y adjetivos terminados en -l, -r, -n, -d, -z, -j.** Forman el plural con -es: *dócil, pl. dóciles; color, pl. colores; pan, pl. panes; césped, pl. céspedes; cáliz, pl. cálices; reloj, pl. relojes.*

Crea una clase llamada Plural, crea un unico atributo denominado palabra y un constructor que inicialize dicho atributo y posteriormente crea un metodo que devuelva el plural de dicha palabra.
Para comprobarlo crea otra unidad de compilacion o archivo que incluya el metodo void, que como hemos indicado anteriormente inicia el programa. Denomina a este fichero *SimulacionPlural.java*

3. API de Java

3.1. Ejercicio 5

Busca información sobre la clase Scanner, usada para leer datos desde el teclado o desde un fichero y explica que realizan los siguientes metodos:

- Scanner(System.in)
- hasNext()
- hasNextInt()
- hasNextDouble()
- next()

- nextInt()
- nextDouble()

3.2. Ejercicio 6

Realiza un programa en Java, que lea desde le entrada estandar tu nombre y que presente por pantalla el siguiente mensaje: Hola *nombreLeido*

3.3. Ejercicio 7

Realiza un programa en Java, que lea desde la entrada un número entero de tres cifras y posteriormente muestre sus cifras por separado. No crees objetos en este programa, haz todo en un metodo *main* de una clase denominadas Cifras.

3.4. Ejercicio 8

Programa que lea tres números enteros H, M, S que contienen hora, minutos y segundos respectivamente, y comprueba si la hora que indican es una hora válida. Emplea el paradigma POO para este ejercicio.

3.5. Ejercicio 9

Programa que lea doce números deciamales por teclado y devuelva su valor medio. No crees objetos en este programa, haz todo en un metodo *main* de una clase denominadas DiezNumeros. Posteriormente muestra por pantalla todos los valores tabulados(con mas de un espacio blanco entre ellos) en cuatro filas y con dos decimales.

3.6. Ejercicio 10

Realiza un programa que lea desde la entrada estándar (terminal) distintas entradas que sean o bien palabras (String), número enteros (int) o números decimales (double) y posteriormente nos diga cuantos valores de cada tipo se han introducido. Para finalizar la entrada de datos usa la letra *q*, de manera que una vez tecleada no siga leyendo datos.
Sigue las siguientes intrucciones para realizarlo:

- Una clase denominada Entrada, que tenga tres atributos denominados palabra, entero y decimal det tipo *int* que sirva para contar el número de valores correspondientes. Un metodo que cuente el numero de palabras, enteros o decimales de manera que cada vez que le llegue un valor de entrada incremente el valor del atributo correspondiente.
- Una clase denominada TestEntrada que tenga el método *main*.
- Muestra los datos formateados usando *printf*

3.7. Ejercico 11

Queremos implementar el típico ejercicio en que dado un número de días, conocer el día de la semana que corresponde. Ejemplo hoy es lunes y dentro de 14 días también es lunes.
Para esto empezamos con día 0 para el domingo. Para realizar el ejercicio usa la estructura de control *switch*. Plantea el ejercicio como quieras.

3.8. Ejercicio 12

Los sistemas de ecuaciones lineales, en el caso de dos ecuaciones con dos incógnitas se pueden representar de forma genérica de la siguiente forma:

$$\left.\begin{array}{l} a{\cdot}x{+}b{\cdot}y{=}e \\ c{\cdot}x{+}d{\cdot}y{=}f \end{array}\right\}$$

Un sistema de ecuaciones tiene solución si:

$$a \cdot d - b \cdot c \neq 0$$

La solución de un sistema de dos ecuaciones con dos incognitas, en el caso que lo tenga, viene dado por la siguiente regla, derivada de la regla de Cramer:

$$x = \frac{e{\cdot}d{-}b{\cdot}f}{a{\cdot}d{-}b{\cdot}c}$$

$$y = \frac{a{\cdot}f{-}e{\cdot}c}{a{\cdot}d{-}b{\cdot}c}$$

Se quiere realizar una clase que represente a dicho sistemas de dos ecuaciones con dos incógnitas, para esto ten en cuenta lo siguiente:

- Usa como atributos los que consideres oportunos, de acuerdo a la representación genérica de un sistema de dos ecuaciones con dos incógnitas, y los tipos que sean *double*
- Un constructor para los objetos de tipo Ecuación.
- Un método boolean que devuelva cierto o falso si el sistema es resoluble o no.
- Dos métodos que devuelvan el valor de x e y. Los tipos a devolver deben ser acordes con los datos de inicialización.

Crea una clase TestEcuación, con el método *main* que cree los dos siguientes objetos:

$$\left.\begin{array}{l} x{+}y{=}1 \\ 2{\cdot}x{+}2{\cdot}y{=}2 \end{array}\right\}$$

$$\left.\begin{array}{l} 2{\cdot}x{+}y{=}7 \\ -x{+}2{\cdot}y{=}{-}1 \end{array}\right\}$$

Debe indicar por pantalla, si el sistema es resoluble o no. Y en el caso que sea resoluble debe mostrar la resolucion de dicho sistema de ecuaciones.