

Proyecto 2 UT4 (para hacer en casa y entregar en GitHub)

Objetivos

Saber:

- construir adecuadamente bucles simples y anidados con sentencias *for* y *while* (repeticiones determinadas e indeterminadas)
- definir y utilizar objetos de la clase *Scanner* para leer valores desde teclado
- invocar al método *main()*
- invocar métodos estáticos

Repasar:

- crear objetos
- efectuar llamadas internas y externas a métodos
- formatear valores
- sentencia condicional *if*

Antes de empezar

- Este ejercicio es para realizar de forma **individual** en casa.
- El proyecto de partida está en <https://github.com/montsemsanz/ENTRE-02-UT4-PracticaBucles>. Deberás hacer un *fork* a tu cuenta y clonarlo en tu PC desde BlueJ tal y como se explicó en clase
- Una vez completado desde BlueJ haz un *push* del último *commit* a GitHub
- No olvides **entregar vía Moodle el texto de la actividad “Terminado proyecto 2 UT4 Bucles”** y pulsar *Enviar para calificar*
- Se valorará en la corrección que el programa esté probado (compila y ejecuta bien) y que esté claramente escrito y organizado (se respetan las reglas de estilo del lenguaje Java, nombres descriptivos, código no duplicado, ...)

- La fecha tope de entrega es el **15 de Noviembre** hasta las **23,30h.**

- Se anulará automáticamente la corrección del ejercicio y se **evaluará con un 0** si se detecta que ha sido copiado o dejado copiar a algún compañero/a
- Se penalizará si no se siguen las normas de entrega del ejercicio
 - ✗ no se ha hecho un *fork* / no se sube vía *commit*
 - ✗ hay algún *commit* posterior a esta fecha de entrega
 - ✗ no se ha enviado el texto de la actividad vía Moodle
- El profesorado podrá convocar al alumno/a para defender oralmente el proyecto

Especificaciones

En este proyecto completarás las clases **PracticaBucles** y **TestPracticaBucles**.

Haz el *fork* del proyecto **ENTRE-02-UT4-PracticaBucles** desde <https://github.com/montsemsanz> a tu cuenta GitHub y desde BlueJ clona el proyecto a tu PC.

Abre el proyecto BlueJ y completa las clases indicadas.

La clase **Utilidades** no hay que modificarla. Incluye dos métodos estáticos que deberás utilizar. Analiza su código.

Clase PracticaBucles

Esta clase incluye un par de constantes que habrá que utilizar y el generador de números aleatorios.

■ **public boolean esImpar(int numero)** – devuelve *true* si numero es impar, *false* en otro caso. Hazlo sin *if*.

■ **public int obtenerNumeroSinCeros(int numero)** - Dado un n° genera y devuelve otro nuevo con las mismas cifras y en el mismo orden pero sin los 0.

Ej -
si numero = 2040 devuelve 24
si numero = 1009 devuelve 19
si numero = 3000 devuelve 3

■ **public void generarNumeros(int n)** – este método genera valores aleatorios enteros en el intervalo [-1000, 50000] (inclusive) y calcula y escribe:

- ✓ la media de todos los números
- ✓ la suma de los impares
- ✓ el máximo de los pares

El proceso termina cuando sale un 0 o bien se han escrito *n* n°s.

Los números se escriben en filas de 5 columnas. Cada n° aleatorio se muestra en un espacio de 12 posiciones en la pantalla y al lado de cada n° se muestra otro n° con las mismas cifras y en el mismo orden pero sin ceros (en 5 posiciones).

Usa en los métodos anteriores bucles while.
!!Atención a la condición de salida del bucle while!!!

```
Blue: Ventana de Terminal - ENTRE-02-UT4-PracticaBucles-SOL
Opciones

Nº máximo de aleatorios a generar 42
o hasta que salga el 0

  78: 78      3448: 3448      -220: -22      338: 338      2043: 243
-398: -398    -664: -664      3792: 3792      2987: 2987      -248: -248
 875: 875      4580: 458      920: 92      3421: 3421      1551: 1551
-928: -928    2537: 2537      4493: 4493      1751: 1751      1549: 1549
 391: 391      3621: 3621      1540: 154      346: 346      2932: 2932
-468: -468    3143: 3143      3150: 315      2696: 2696      4008: 48
-271: -271    1072: 172      799: 799      2030: 23      1381: 1381
4705: 475     3177: 3177      2544: 2544      4288: 4288      2075: 275
1253: 1253     3969: 3969

      Media: 1911,57
      Suma impares: 45450
      Máximo pares: 4580

Can only enter input while your programming is running
```

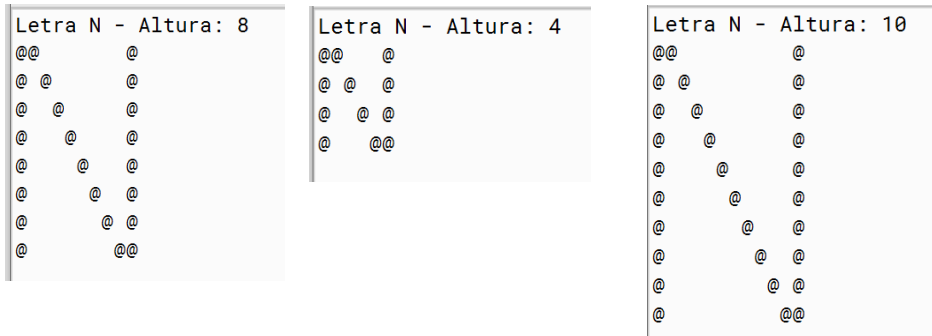
Columna a 25 posiciones alineada a derecha

Columna a 10 posiciones alineada a derecha (2 posiciones para decimales)

■ **public void escribirLetraN(int altura)** – muestra en pantalla la letra N de altura dada (asumimos altura correcta y con un valor > 2) usando el carácter indicado por la constante. Usa el método privado de ayuda **escribirCaracter()**

- x la altura de la letra es el n° de filas que tiene
- x en cada fila (incluidas la primera y la última) se repite el mismo patrón
 - o escribir el carácter @
 - o escribir espacios
 - o escribir el carácter @
 - o escribir espacios
 - o escribir el carácter @

■ **private void escribirCaracter(char caracter, int n)** – escribe *n* veces el caracter indicado en la misma línea



Usa bucles for para dibujar la letra en ambos métodos.

Clase TestPracticaBucles

Completa en esta clase el método **main()**:

- x define y crea el teclado
- x define e instancia un objeto **PracticaBucles**
- x pide al usuario la cantidad máxima de aleatorios a generar
- x valida que sea un n° positivo
- x muestra las estadísticas de números aleatorios
- x haz una pausa y borra la pantalla
 - o para hacer esto utiliza los método estáticos de la clase **Utilidades**. Como son estáticos no hay que crear previamente un objeto de la clase, se invocan **Utilidades.hacerpausa()** y **Utilidades.borrarPantalla()**
- x pide al usuario un valor para la altura de la letra N
- x valida que este valor es correcto, es decir, está en el intervalo $3 \leq \text{altura} \leq 10$
- x muestra la letra N

Prueba la aplicación completa:

- a) desde BlueJ – invocado al método **main()**
- b) desde línea de comandos

```

Dame nº máximo de aleatorios a generar (n > 0): -22
Dame nº máximo de aleatorios a generar (n > 0): 0
Dame nº máximo de aleatorios a generar (n > 0): 43

Nº máximo de aleatorios a generar 43
o hasta que salga el 0

3751: 3751      80: 8      -327: -327      541: 541      838: 838
1286: 1286      852: 852      3122: 3122      4568: 4568      1049: 149
4384: 4384      2800: 28      500: 5      1482: 1482      2768: 2768
3237: 3237      1417: 1417      3283: 3283      2733: 2733      1421: 1421
3891: 3891      1266: 1266      2116: 2116      1983: 1983      458: 458
204: 24      4271: 4271      2712: 2712      -842: -842      -786: -786
2018: 218      3455: 3455      758: 758      3622: 3622      2883: 2883
-314: -314      2176: 2176      1251: 1251      426: 426      -441: -441
3797: 3797      4593: 4593      825: 825

Media: 1862,95
Suma impares: 43613
Máximo pares: 4568

Intro para continuar

```

```
Teclea altura de la letra N (3 <= altura <= 10): 2
Altura incorrecta, Teclea altura de la letra N (3 <= altura <= 10): -8
Altura incorrecta, Teclea altura de la letra N (3 <= altura <= 10): 14
Altura incorrecta, Teclea altura de la letra N (3 <= altura <= 10): 9
Letra N - Altura: 9
@@      @
@ @      @
@  @      @
@   @      @
@    @      @
@     @      @
@      @      @
@       @      @
@        @      @
@         @      @
@          @      @
```

Rúbrica evaluación	
esImpar	2,5
generarNumeros	30
obtenerNumeroSinCeros	30
escribirLetraN	25
escribirCaracter	5
main	7,5
	100,00
Penalización (no compila, no probado, no se siguen reglas estilo,)	-0,5 (sobre 10)