Todos los ejercicios que se plantean a continuación van a realizar tareas
que no requieren ninguna coordinación entre ellas, es decir,
cada hilo que va a formar parte de un proceso (programa) va a ser independiente de los otros.

Ejercicios hilos extendiendo Thread

1. Descarga el proyecto ContadorAscendenteDescendente y complétalo.
La clase AppMain crea los dos hilos e inicia su ejecución.
La clase ContadorAscendente representa a un hilo que cuenta hacia arriba hasta el valor indicado por el atributo y escribe el valor después de cada cuenta.
La clase ContadorDescendente representa a un hilo que cuenta hacia abajo desde el valor indicado por el atributo y escribe el valor después de cada cuenta.
Da la oportunidad de ejecutarse a cada hilo llamando a yield() después de escribir el valor del contador.
Ejecuta varias veces el programa para observar los resultados que se producen

posible ejecución

```
Hilo: Thread-29 Contador:1
Hilo: Thread-29 Contador:2
Fin del hilo main()
Hilo: Thread-30 Contador:8
Hilo: Thread-30 Contador:7
Hilo: Thread-30 Contador:6
Hilo: Thread-30 Contador:5
Hilo: Thread-30 Contador:4
Hilo: Thread-30 Contador:3
Hilo: Thread-30 Contador:2
Hilo: Thread-30 Contador:1
Fin del hilo Thread-30
Hilo: Thread-29 Contador:3
Hilo: Thread-29 Contador:4
Hilo: Thread-29 Contador:5
Hilo: Thread-29 Contador:6
Hilo: Thread-29 Contador:7
Hilo: Thread-29 Contador:8
Hilo: Thread-29 Contador:9
Hilo: Thread-29 Contador:10
Fin del hilo Thread-29
```

2. Crea un nuevo proyecto SumaFactorial en Eclipse (en el *workspace* UT1).
Añade la clase Suma que representa a un hilo que calcula y escribe la suma de los n primeros números naturales siendo n un valor que recibe el constructor de la clase.
Añade la clase Factorial que representa a un hilo que calcula y escribe el factorial de n siendo n un valor que recibe el constructor de la clase .
Incluye la clase Test con el método main() que acepta como argumento n (si no es así se emite un mensaje de error) y crea y lanza a ejecución los dos hilos.

```
La suma de 7 es: 55
Fin del hilo main()
El factorial de 7 es: 5040
```

posible ejecución

3. Crea un proyecto BlueJ de nombre Muestra Hora. El proyecto incluye las siguientes clases:

a) HiloHora – es el hilo que muestra 5 mensajes indicando en cada uno de ellos el nº de mensaje, el nombre del hilo y la hora actual. Obtén el nombre del hilo a través de currentThread().
Después de cada mensaje el hilo duerme 1 segundo
Para obtener la hora en Java consulta
http://www.mkyong.com/java/java-how-to-get-current-date-time-date-and-calender/

b) AppHora – contiene el main(). Lanza a ejecución 10 hilos de tipo HiloHora.

Ejercicios hilos implementando Runnable

4. Realiza una copia del proyecto que has hecho en el ejercicio 1. Haz los cambios necesarios para que ahora las clases ContadorAscendente y ContadorDescendente representen tareas que implementan Runnable.

5. Descarga el proyecto Lector Escritor números con Runnable AL y complétalo.
Desde el hilo *main* se lanzarán dos hilos de ejecución:

- un hilo que lee desde un fichero de texto números, muestra cada nº leído, duerme el hilo 300 msg y al final de la ejecución del hilo muestra el total de nºs leídos y su suma. El fichero ya está creado y es *numeros.txt*
- un hilo que escribe en un fichero de texto números aleatorios (cada nº comprendido entre 1 y 100) y escribe cada nº generado en una línea de texto del fichero *resultado.txt.* El hilo duerme 300 msg después de escribir cada nº.

posible ejecución

```
Número  leído 123
Fin del hilo main
Número  escrito 90
Número  leído 4
Número  escrito 7
Número  leído 55
Número  escrito 13
Número  leído 12
Número  escrito 35
Número  leído 78
Número  escrito 31
Número  leído 3
Número  escrito 90
Total Números leídos 6 Suma: 275
Fin del hilo lector
Número  escrito 25
Número  escrito 22
Número  escrito 74
Número  escrito 96
Número  escrito 10
Número  escrito 65
```

6. Descarga el proyecto Contador palabras en fichero AL y complétalo.
Dentro del hilo main tendrás que lanzar varios hilos de ejecución, tantos como ficheros se pasen como argumentos al main().
Cada hilo cuenta utilizando Scanner las palabras que hay en un fichero de texto y muestra al final del hilo el total de palabras encontradas

```
Fin del hilo main
fichero3.txt: 7
fichero2.txt: 4
fichero1.txt: 5
```

7. Abre el proyecto Ejemplo con clase anónima AL y complétalo.
Trabajaremos en este ejemplo con los métodos de la clase Thread (para mostrar las propiedades de los hilos) y crearás un hilo utilizando una clase anónima.

posible ejecución

```
Segundo hilo todavía no vivo.
Segundo hilo ahora vivo.
Propiedades del hilo main.
MAIN Nombre: main
MAIN Prioridad: 5
MAIN ID: 49
MAIN Estado: RUNNABLE
MAIN Está vivo?: true
MAIN Demonio?: false
 Primer hilo - main - finalizado.
Started running. - Thread-9
1 2 3 4 5 6 7 8 9 10
Propiedades del segundo  hilo.
SEGUNDO Nombre: Thread-9
SEGUNDO Prioridad: 5
SEGUNDO ID: 50
SEGUNDO Estado: RUNNABLE
SEGUNDO Está vivo?: true
SEGUNDO  Demonio?: false
Fin de run alcanzado en segundo hilo.
```