Creación de Threads

Fibonacci:

Podemos hacerlo de dos formas: implementando Runnable o extendiendo Thread. De las dos formas se que el método run() igual.

Main.java | Fibonacci.java

```java
package es.iespuertodelacruz.odhh.extensionThreats;

//public class Fibonacci implements Runnable{
2 usages
public class Fibonacci extends Thread{

    5 usages
    public int parametro;

    1 usage
    public Fibonacci(int parametro) {
        this.parametro = parametro;
    }

    @Override
    public void run() {
        int num1 = 1;
        int num2 = 1;
        while (num1 <= parametro || num2 <= parametro) {
            if (num1 <= parametro) {
                System.out.println("num1: " + num1);
            }
            if (num2 <= parametro) {
                System.out.println("num2: " + num2);
            }
            num1 = num1 + num2;
            num2 = num1 + num2;
        }
    }
}

```

Y cogemos el parámetro desde el constructor.
Luego le hacemos un new desde el main y le hacemos start o run en el caso de extender, y el new Thread en caso de implementar.

Main.java | Fibonacci.java

```java
package es.iespuertodelacruz.odhh.extensionThreats;

public class Main {
    public static void main(String[] args) {
        Fibonacci f = new Fibonacci( parametro: 20);
        f.start();
        //f.run();
        //new Thread(f).start();
    }
}
```

Carrera de hilos

Main.java | Coche.java

```java
package es.iespuertodelacruz.odhh.extensionThreats;

//public class Coche extends Thread {
10 usages
public class Coche implements Runnable {

    2 usages
    long time;

    5 usages
    public void setTime(long time) {
        this.time = time;
    }

    @Override
    public void run() {
        for (int i = 0; i < 50; i++) {
            try {
                Thread.sleep(time);
            } catch (Exception e) {
                e.printStackTrace();
            }

        }
        System.out.println("Ha llegado el coche: " + /*this*/new Thread().getName());
    }
}
```

Si usas el extends no necesitas el “Thread.” en el sleep y en el getName() le puedes hacer “this”, haciéndolo con el implements usaremos lo que no está comentado. Y el setTime es para ponerle la cantidad de sleep que va a estar en cada pasada del bucle.

Y en el main si hacemos el extends no hace falta lo de new Thread sino haciendo un start funcionará, de todos modos funcionará así hagas lo que hagas.