EXECUTOR

Es una interfaz pública.

Un objeto que ejecuta Runnable tasks que se le han entregado.
Esta interfaz provee de maneras de desacoplar el submit de una task de los mecanismos de como cada task se ejecuta, incluso de algunos detalles del thread de uso, como scheduling, etc.
Separa la definición de un Thread de su mecánica de ejecución.
Un Executor se utiliza en lugar de crear threads de manera explícita. Por ejemplo, en lugar de invocar:

new Thread (new (RunnableTask())).start() para cada conjunto de tareas, se usaría:

```
Executor executor = anExecutor;
executor.execute (new RunnableTask1());
executor.execute (new RunnableTask2());
...
```

Sirve para la planificación (scheduling) de Runnables.

Un ejemplo sencillo de implementación sería:
Tenemos dos clases que implementan la interfaz Runnable:

```
public class RunnableTask1 implements Runnable{
    @Override
    public void run(){
        System.out.println("Soy el RunnableTask1");
    }
}
y
public class RunnableTask2 implements Runnable{
    @Override
    public void run(){
        System.out.println("Soy el RunnableTask2");
    }
}
```

Y ahora tenemos la clase que contendrá el programa principal:

```
public class Executor1 {
    public static void main (String[] args){
        Executor executor = new Executor (){
            @Override → se sobreescriben los métodos de la interfaz
            public void execute(Runnable r){
                r.run();
            }
        };
```

↑ le digo que ha de ejecutar el método run() de los Runnables que le paso.

```
        executor.execute (new RunnableTask1());
        executor.execute (new RunnableTask2());
```

En lugar de `Executor executor= new Executor (){`

```
            @Override
            public void execute (Runnable r){
                ...
            }
        };
```

Se podría haber creado una clase que implementara la interfaz Executor:

```
public class DirectorExecutor implements Executor {
    @Override
    public void execute (Runnable r){
        r.run();
    }
}
```

y luego declarar dentro de la clase Executor1:

```
DirectorExecutor executor = new DirectorExecutor ();
```

o bien

```
Executor executor=new DirectorExecutor ();
```

teniendo en cuenta que en esta última forma no se podrían usar los métodos que DirectorExecutor añadiera.