

Dado un número se debe determinar si es número primo o, en caso de serlo, el mayor divisor que no sea el mismo número

Realizar la actividad utilizando Callable y luego repetir el ejercicio pero utilizando Runnable.

Consejos:

Solución Callable.- Aprovechar que el método call puede devolver un retorno de cualquier tipo y lanzar una excepción. El retorno se utiliza para proporcionar el mayor divisor, en caso de que el número no sea primo, mientras que la excepción se utiliza como mecanismo para indicar que el número es primo.

Solución Runnable.- Al no poder aprovechar la posibilidad de devolver retorno, buscar una solución alternativa, consejo utilizar un atributo privado con su método de acceso y depositar en él el valor -1 si el número es primo o el mayor divisor. Necesitaras también utilizar variables auxiliares.

```
1 package com.mycompany.psp_callableRunnable;
2
3 /**
4  *
5  * @author Fran
6  */
7 public class PSP_CallableRunnable {
8
9     public static void main(String[] args) {
10
11         System.out.println("*****Solución con Callable*****");
12         PrimoCallable p1 = new PrimoCallable(_numero: 78);
13         try {
14             System.out.println(":" + p1.call());
15         } catch (Exception ex) {
16             System.out.println(":" + ex.getMessage());
17         }
18
19         System.out.println(":" + "");
20         System.out.println("*****Solución con Runnable*****");
21
22         PrimoRunnable p2 = new PrimoRunnable(_numero: 78);
23
24         p2.run();
25
26         if (p2.getMayorDivisor() == -1) {
27             System.out.println(":" + "Número primo");
28         } else {
29             System.out.println("Número no primo. Mayor divisor: " + p2.getMayorDivisor());
30         }
31
32     }
33 }
34
35
36
37
38 }
```

```

1 package com.mycompany.psp_callableRunnable;
2
3 import java.util.concurrent.Callable;
4
5 /**
6  *
7  * @author Fran
8  */
9 public class PrimoCallable implements Callable<String> {
10
11     private int numero;
12
13     public PrimoCallable(int _numero) {
14         this.numero = _numero;
15     }
16
17     @Override
18     public String call() throws Exception {
19         if (!esPrimo(numero)) {
20             return "El número no es primo, y el mayor divisor es " + mayorDivisor(numero);
21         } else {
22             throw new Exception("El número es primo");
23         }
24     }
25 }
26
27 private boolean esPrimo(int numero) {
28     boolean divisible = true;
29
30     for (int i = 2; i < numero; i++) {
31         if (numero % i == 0) {
32             divisible = false;
33             break;
34         }
35     }
36
37     return divisible;
38 }
39
40 private int mayorDivisor(int numero) {
41     int mayorDivisor = 0;
42
43     for (int i = 2; i < numero; i++) {
44         if (numero % i == 0) {
45             mayorDivisor = i;
46         }
47     }
48
49     return mayorDivisor;
50 }

```

```

1 package com.mycompany.psp_callablerunnable;
2
3 /**
4  *
5  * @author Fran
6  */
7 public class PrimoRunnable implements Runnable {
8
9     private int numero;
10    private int mayorDivisor;
11
12    public PrimoRunnable(int _numero) {
13
14        this.numero = _numero;
15    }
16
17    public int getNumero() {
18        return numero;
19    }
20
21    public void setNumero(int numero) {
22        this.numero = numero;
23    }
24
25    public int getMayorDivisor() {
26        return mayorDivisor;
27    }
28
29    public void setMayorDivisor(int mayorDivisor) {
30        this.mayorDivisor = mayorDivisor;
31    }
32
33    @Override
34    public void run() {
35        if (!esPrimo(numero: this.numero)) {
36
37            this.mayorDivisor = mayorDivisor(numero: this.numero);
38
39        } else {
40
41            this.mayorDivisor = -1;
42
43        }
44    }
45
46    private boolean esPrimo(int numero) {
47
48        boolean divisible = true;
49
50        for (int i = 2; i < numero; i++) {
51
52            if (numero % i == 0) {
53
54                divisible = false;
55                break;
56
57            }
58
59        }
60
61        return divisible;
62    }
63
64 }

```

```

66     private int mayorDivisor(int numero) {
67     |
68         int mayorDiv = 0;
69
70         for (int i = 2; i < numero; i++) {
71
72             if (numero % i == 0) {
73
74                 mayorDiv = i;
75             }
76
77         }
78
79         return mayorDiv;
80
81     }
82
83 }
84

```

```

-----< com.mycompany:PSP_CallableRunnable >-----
Building PSP_CallableRunnable 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ PSP_CallableRunnable ---
*****Solución con Callable*****
El número no es primo, y el mayor divisor es 39

*****Solución con Runnable*****
Número no primo. Mayor divisor: 39
-----

BUILD SUCCESS
-----

Total time:  0.354 s
Finished at: 2023-10-28T13:52:43+02:00
-----
|

```

```

public static void main(String[] args) {

    System.out.println( x: "*****Solución con Callable*****");
    PrimoCallable p1 = new PrimoCallable( _numero: 7);
    try {
        System.out.println( x: p1.call());
    } catch (Exception ex) {
        System.out.println( x: ex.getMessage());
    }

    System.out.println( x: "");
    System.out.println( x: "*****Solución con Runnable*****");

    PrimoRunnable p2 = new PrimoRunnable( _numero: 7);

    p2.run();

    if (p2.getMayorDivisor() == -1) {

        System.out.println( x: "Número primo");

    } else {

        System.out.println("Número no primo. Mayor divisor: " + p2.getMayorDivisor());

    }

}

```

```

-----< com.mycompany:PSP_CallableRunnable >-----
Building PSP_CallableRunnable 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ PSP_CallableRunnable ---
*****Solución con Callable*****
El número es primo

*****Solución con Runnable*****
Número primo
-----

BUILD SUCCESS
-----

Total time: 0.346 s
Finished at: 2023-10-28T13:53:24+02:00
-----

```

```
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;

public class TareaCallable implements Callable<Integer> {
    private int numero;

    public TareaCallable(int numero) {
        this.numero = numero;
    }

    @Override
    public Integer call() throws Exception {
        Thread.sleep(3000);
        for (int i=numero-1;i>1;i--) {
            if (numero%i==0) {
                return i;
            }
        }
        throw new Exception("El número es primo");
    }
}
```