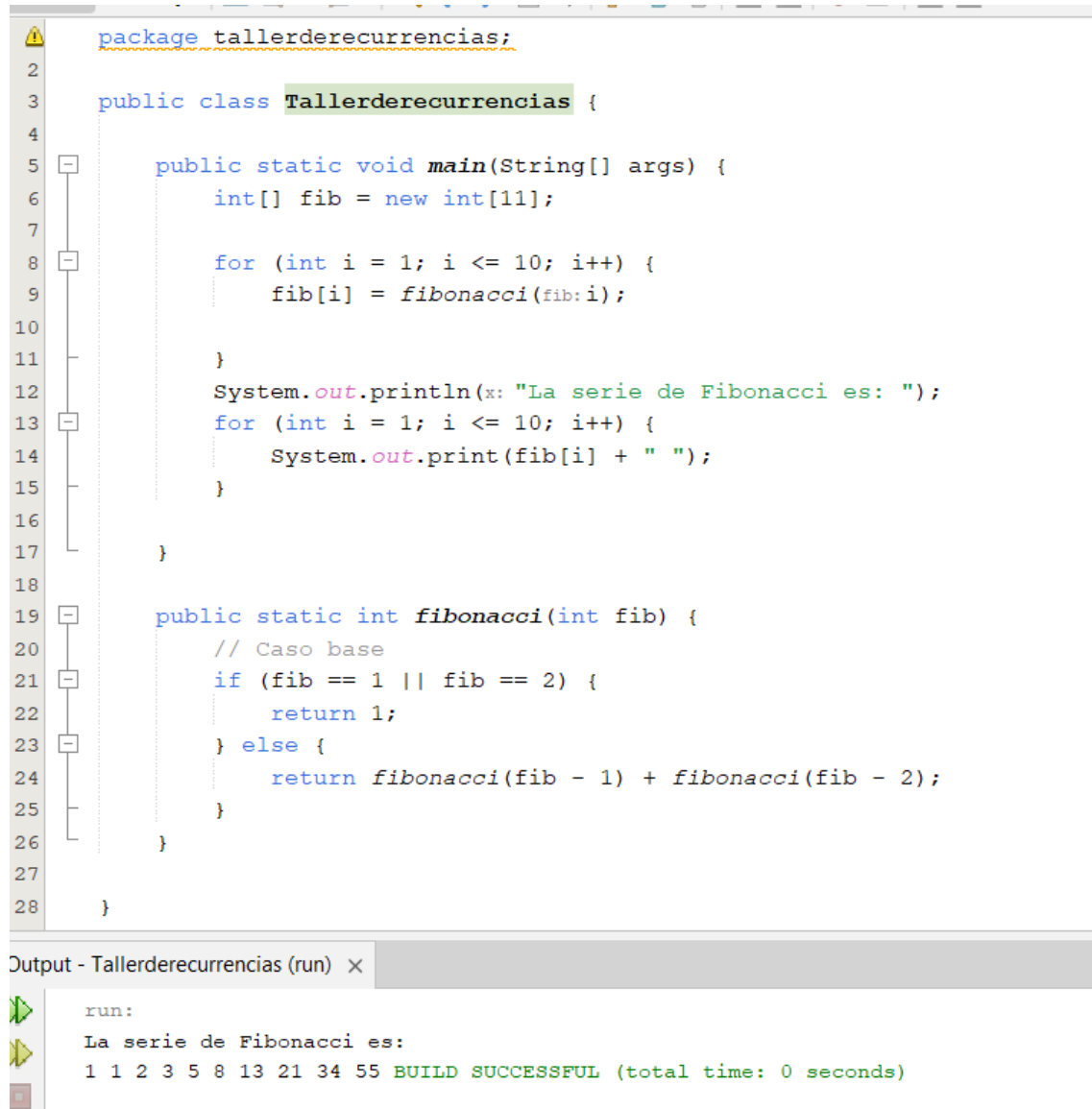


Taller de recurrencias: Análisis del algoritmo de Fibonacci

Valentina Samaniego



The screenshot shows an IDE with a Java file named `Tallerderecurrencias.java`. The code defines a package `tallerderecurrencias` and a public class `Tallerderecurrencias`. Inside the class, there is a `main` method that initializes an array `fib` of size 11, loops from `i = 1` to `i = 10` to calculate Fibonacci numbers using a recursive `fibonacci` method, and prints the results. The `fibonacci` method has a base case for `fib == 1` or `fib == 2` returning 1, and otherwise returns the sum of `fibonacci(fib - 1)` and `fibonacci(fib - 2)`. The output window shows the execution results: "La serie de Fibonacci es: 1 1 2 3 5 8 13 21 34 55" followed by "BUILD SUCCESSFUL (total time: 0 seconds)".

```
package tallerderecurrencias;

public class Tallerderecurrencias {

    public static void main(String[] args) {
        int[] fib = new int[11];

        for (int i = 1; i <= 10; i++) {
            fib[i] = fibonacci(fib, i);
        }

        System.out.println("La serie de Fibonacci es: ");
        for (int i = 1; i <= 10; i++) {
            System.out.print(fib[i] + " ");
        }
    }

    public static int fibonacci(int fib) {
        // Caso base
        if (fib == 1 || fib == 2) {
            return 1;
        } else {
            return fibonacci(fib - 1) + fibonacci(fib - 2);
        }
    }
}
```

Output - Tallerderecurrencias (run) x

```
run:
La serie de Fibonacci es:
1 1 2 3 5 8 13 21 34 55 BUILD SUCCESSFUL (total time: 0 seconds)
```

Código en texto

```
package tallerderecurrencias;
```

```
public class Tallerderecurrencias {
```

```
    public static void main(String[] args) {
```

```
        int[] fib = new int[11];
```

```
        for (int i = 1; i <= 10; i++) {
```

```
            fib[i] = fibonacci(i);
```

```

    }

    System.out.println("La serie de Fibonacci es: ");

    for (int i = 1; i <= 10; i++) {

        System.out.print(fib[i] + " ");

    }

}

public static int fibonacci(int fib) {

    // Caso base

    if (fib == 1 || fib == 2) {

        return 1;

    } else {

        return fibonacci(fib - 1) + fibonacci(fib - 2);

    }

}

}

```

Prueba de escritorio a mano

Valentina Samaniego

TALLER DE RECURRENCIAS: ALGORITMO FIBONACCI

PRUEBA DE ESCRITORIO

i	Fib[i]	Ejecuta ↓
1	1	fib(1) = 1
2	1	fib(2) = 1
3	2	fib(3) = fib(2) + fib(1) = 2
4	3	fib(4) = fib(3) + fib(2) = 3
5	5	fib(5) = fib(4) + fib(3) = 5
6	8	fib(6) = fib(5) + fib(4) = 8
7	13	fib(7) = fib(6) + fib(5) = 13
8	21	fib(8) = fib(7) + fib(6) = 21

$$T(n) = \begin{cases} 1 & \text{Si } n=1 \text{ o } n=2 \\ T(n-1) + T(n-2) + 1 & \text{Si } n \geq 2 \end{cases}$$

n=5

n	T(n)
1	1
2	1
3	T(2) + T(1) + 1 = 3
4	T(3) + T(2) + 1 = 5
5	T(4) + T(3) + 1 = 9

3 llamadas con recurrencia