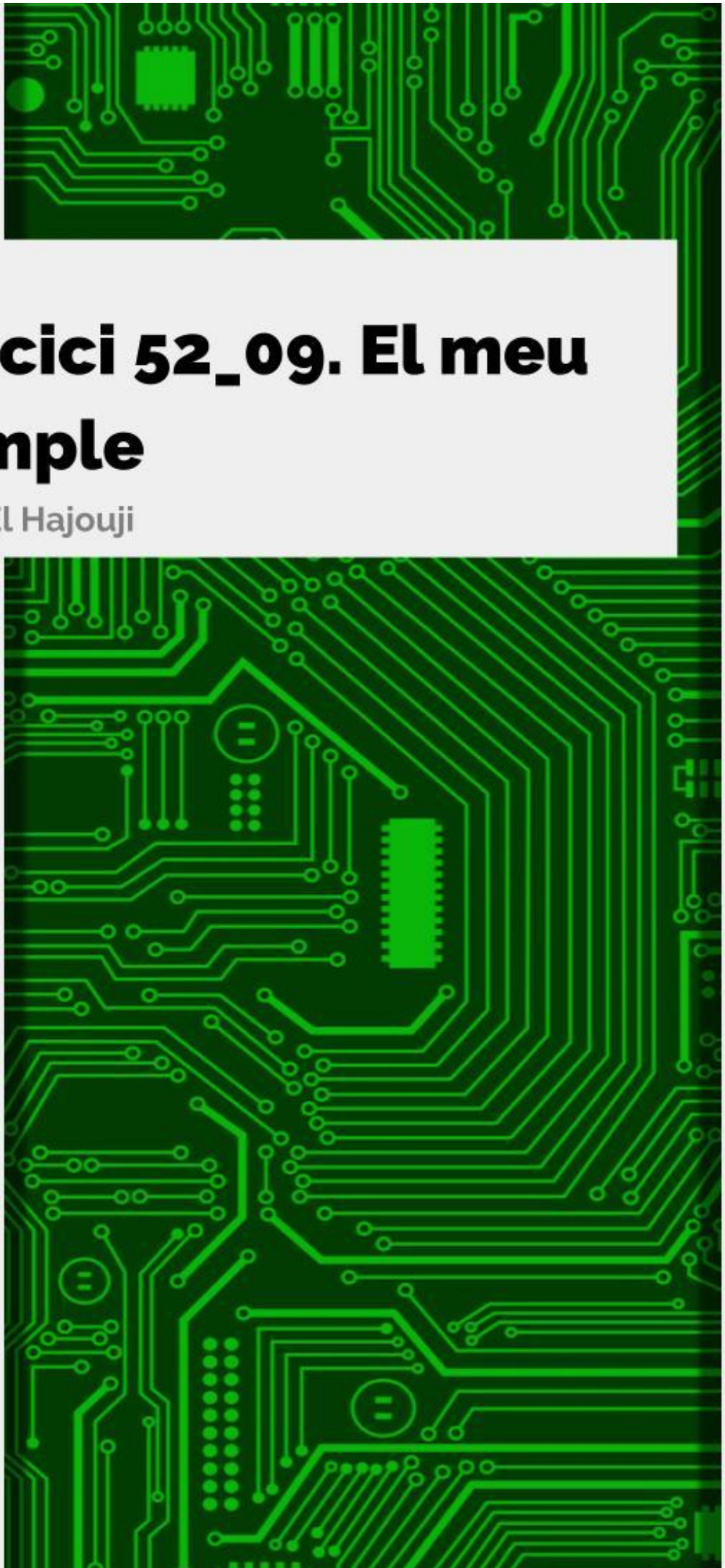

DAM 1

Exercici 52_09. El meu exemple

Soufian El Hajouji

ÍNDEX

Pregunta 1. El meu exemple Refactoring

Segons el que has après a la introducció, descriu un exemple de canvi de codi teu que es pugui considerar refacció i un altre que no. Pots fer servir codi de qualsevol dels teus exercicis del cicle (especifica la procedència). No oblidis descriure el perquè de la teva classificació basant-te en els conceptes d'aquesta secció.

Un exemple de canvi que podríem fer per refacturar el codi seria extreure la lògica de les operacions matemàtiques a mètodes separats. Així, el codi seria més modular i fàcil de llegir i mantenir. Per exemple:

DAQUEST CODI:

```
public class CalculadoraBasica {
    public static void main(String[] args) {
        // obté operands d'entrada
        System.out.println("Primer operand:");
        int primerOperand = Integer.parseInt(Entrada.readLine());
        System.out.println("Segon operand:");
        int segonOperand = Integer.parseInt(Entrada.readLine());

        // operacions
        int suma = primerOperand + segonOperand;
        int resta = primerOperand - segonOperand;
        int multiplicacio = primerOperand * segonOperand;
        int divisio = primerOperand / segonOperand;

        // mostra resultats
        System.out.println(primerOperand + " + " + segonOperand + " = " + suma);
        System.out.println(primerOperand + " - " + segonOperand + " = " + resta);
        System.out.println(primerOperand + " * " + segonOperand + " = " + multiplicacio);
        System.out.println(primerOperand + " / " + segonOperand + " = " + divisio);
    }
}
```

REFACTORING:

```
public class CalculadoraBasica {
    public static void main(String[] args) {
        // obté operands d'entrada
        System.out.println("Primer operand:");
        int primerOperand = Integer.parseInt(Entrada.readLine());
        System.out.println("Segon operand:");
        int segonOperand = Integer.parseInt(Entrada.readLine());

        // operacions
        int suma = sumar(primerOperand, segonOperand);
        int resta = restar(primerOperand, segonOperand);
        int multiplicacio = multiplicar(primerOperand, segonOperand);
        int divisio = dividir(primerOperand, segonOperand);
```

```
        // mostra resultats
        System.out.println(primerOperand + " + " + segonOperand + " = " + suma);
        System.out.println(primerOperand + " - " + segonOperand + " = " + resta);
        System.out.println(primerOperand + " * " + segonOperand + " = " + multiplicacio);
        System.out.println(primerOperand + " / " + segonOperand + " = " + divisio);
    }
    private static int sumar(int a, int b) {
        return a + b;
    }
    private static int restar(int a, int b) {
        return a - b;
    }
    private static int multiplicar(int a, int b) {
        return a * b;
    }
    private static int dividir(int a, int b) {
        return a / b;
    }
}
```

En aquest exemple, hem separat la lògica de les operacions matemàtiques en mètodes separats, el que fa que el codi sigui més modular i fàcil de llegir. Així mateix, és més fàcil de mantenir perquè, si volem fer un canvi en la lògica d'una operació, només hem de modificar el mètode corresponent en lloc de buscar la secció del codi que la conté. Això és un exemple de refactoring.

D'altra banda, un exemple de canvi que no seria considerat com a refactoring seria canviar el nom d'una variable sense cap motiu. Això no milloraria la llegibilitat ni la mantenibilitat del codi, simplement faria que fos més difícil de comprendre per a altres programadors.