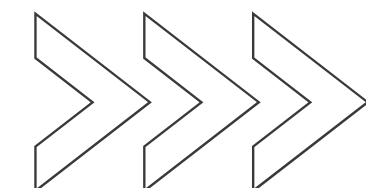
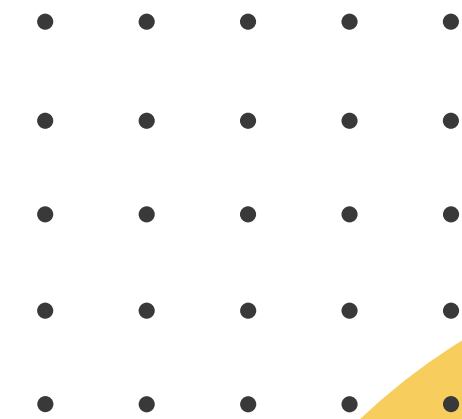


Astuce clean code : Organisez vos fonctions comme une liste d'étapes



Philémon Globléhi
Développeur Back-end Java

```
1 // ❌ Désorganisé : l'ordre perturbe la lecture
2 public class OrderService {
3
4     public void placeOrder(Order order) {
5         validateOrder(order);
6         registerOrderInWarehouse(order);
7         sendEmailTo(order.getCustomer());
8     }
9
10    private void sendEmailTo(Customer customer) { ... } // Étape 3 en premier
11    private void registerOrderInWarehouse(Order order) { ... } // Étape 2 après
12    private void validateOrder(Order order) { ... } // Étape 1 à la fin
13 }
14
15
```

```
1 // ✅ Organisé : logique claire
2 public class OrderService {
3
4     // Méthode principale en haut
5     public void placeOrder(Order order) {
6         validateOrder(order); // Étape 1 : Validation
7         registerOrderInWarehouse(order); // Étape 2 : Stockage
8         sendEmailTo(order.getCustomer()); // Étape 3 : Notification
9     }
10
11    // Sous-méthodes dans l'ordre d'appel
12    private void validateOrder(Order order) { ... }
13    private void registerOrderInWarehouse(Order order) { ... }
14    private void sendEmailTo(Customer customer) { ... }
15 }
16
```