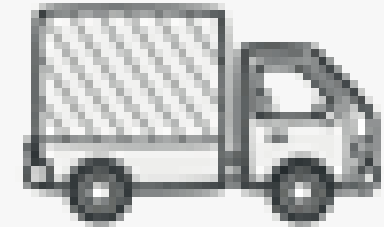


Що означає ця тема?

"Замінити конструктор фабричним методом" — це техніка рефакторингу, яка передбачає заміну виклику конструктора спеціальним статичним методом (фабрикою). Цей метод створює екземпляр класу, надаючи більше контролю над процесом створення об'єкта.

**LogisticsApp**



Для чого це потрібно?

Дозволяє створювати об'єкти з більш гнучкою логікою.

Приховує складність ініціалізації об'єктів.

Полегшує підтримку, тестування та зміну коду.

Дозволяє повертати підкласи або кешовані екземпляри.



```
class User {  
    private String name;  
  
    public User(String name) {  
        this.name = name;  
    }  
}
```

До рефакторингу метод `calculateTax()` був публічним (`public`), що дозволяло викликати його з будь-якого місця у програмі:

Після рефакторингу я зробив  
конструктор приватним і додав  
фабричний метод:

```
class User {  
    private String name;  
  
    private User(String name) {  
        this.name = name;  
    }  
  
    public static User create(String name) {  
        return new User(name);  
    }  
}
```

Що змінилося?

- Використання `User.create("John")` замість `new User("John")` покращує читабельність.
- Конструктор приховано, що дозволяє змінювати внутрішню логіку створення об'єкта без зміни коду клієнта.
- Додано можливість контролювати створення об'єктів (наприклад, можна повертати закешовані екземпляри).

THANK YOU