

Template Method Design Pattern

תבנית העיצוב של Template Method היא תבנית עיצוב מסוג התנהגות, היא מגדירה שלד לאלגוריתמים ומאפשרת לתתי מחלקות להגדיר מחדש שלבים מסויים באלגוריתם בלי לשנות את המבנה הכללי שלו.

דוגמה לשימוש ב- Template Method

תהליך יצירה של קפה יכול להיות דוגמה טובה לשימוש בתבנית העיצוב של Template Method.

השלבים הבסיסיים ליצירת קפה הם:

- להרתיח מים
- להוסיף קפה לכוס
- להוסיף תוספות אחרות

אבל הפרטים של כל שלב יכולים להשתנות בהתאם לסוג הקפה שאנחנו רוצים להכין.

עם תבנית העיצוב של Template Method נוכל ליצור מחלקה Coffee שמכילה את השלד ליצירת קפה, ומחלקות מופשטות שמממשות את השלבים המסויימים לסוג הקפה שאנחנו רוצים להכין, והתאים אישית שלבים ספציפיים כמו סוג פולי, כמות המים התוספות ועוד.

דוגמת קוד בהמשך

יתרונות

- שימוש חוזר של קוד: אנחנו משתמשים באותו קוד כמה פעמים במחלקות שונות, וזה מפחית את השימוש החוזר באותו קוד.
- גמישות: אם נרצה לשנות את השלבים באלגוריתם, נוכל לעשות זאת בקלות על ידי הגדרת מחדש שלבים במחלקת האב, ולא נצטרך לשנות את כל תתי המחלקות.

דוגמת קוד

בדוגמה הבאה ישנם שלוש מחלקות:

- CoffeeTemplate - מחלקת האב שמכילה את השלד ליצירת קפה.
- BlackCoffee - מחלקה שמממשת את השלד ומכילה קפה שחור.
- Latte - מחלקה שמממשת את השלד ומכילה לאטה.

יש לנו פונקציה מגדירה את האלגוריתם ליצירת קפה (שהיא בלתי ניתנת לשינוי), ופונקציות נוספות שניתן לשנות אותן במחלקות המופשטות.

```
abstract class CoffeeTemplate {  
  
    // Template method with defined steps  
    public final void makeCoffee() {  
        boilWater();  
        brewCoffee();  
        addCondiments();  
    }  
  
    private void boilWater() {  
        System.out.println("Boiling water...");  
    }  
  
    // Abstract method to be implemented by subclasses  
    protected abstract void brewCoffee();  
  
    // Hook method that can be overridden by subclasses  
    protected void addCondiments() {  
        System.out.println("Adding sugar...");  
    }  
}
```

```
class BlackCoffee extends CoffeeTemplate {
    @Override
    protected void brewCoffee() {
        System.out.println("Brewing black coffee...");
    }
}

class Latte extends CoffeeTemplate {
    @Override
    protected void brewCoffee() {
        System.out.println("Brewing espresso...");
    }

    @Override
    protected void addCondiments() {
        System.out.println("Adding milk and sugar...");
    }
}
```