

Flyweight Design Pattern

תבנית העיצוב Flyweight היא תבנית עיצוב מבנית שנועדה לפתור את הבעיה של צריכת זיכרון גבוהה כאשר יש צורך במספר רב של אובייקטים קטנים. הבעיה מתעוררת כאשר כל אובייקט קטן מחזיק במידע משלו, גם אם המידע הזה זהה או דומה עבור אובייקטים רבים. כתוצאה מכך, צריכת הזיכרון יכולה לגדול באופן משמעותי, מה שעלול להוביל לבעיות ביצועים.

הפתרון של תבנית העיצוב Flyweight הוא להשתמש באובייקטים קיימים במקום ליצור אובייקטים חדשים. נשמור את האובייקטים הקיימים במבנה נתונים, וכאשר משתמש רוצה ליצור אובייקט חדש, נבדוק האם כבר קיים אובייקט זהה במבנה הנתונים. אם כן, נחזיר את האובייקט הקיים, ואם לא, ניצור אובייקט חדש ונוסיף אותו למבנה הנתונים.

יתרונות

- חסכון בזיכרון - אובייקטים זהים משתמשים באותו מידע.

דוגמה לשימוש ב-Flyweight

נניח שיש לנו מחלקה שמייצגת נקודה במרחב דו-מימדי. לא נרצה ליצור אובייקט חדש כל פעם שאנו רוצים ליצור נקודה חדשה, ולכן נשתמש בתבנית העיצוב Flyweight. נגדיר מחלקה בשם `Point` שמכילה את המידע המשותף לכל הנקודות, ומחלקה נוספת בשם `PointFactory` שמחזירה את הנקודה המתאימה לפי הקואורדינטות שנשלחות לה.

```
class Point {
    private int x;
    private int y;
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public void draw() {
        // draw point
    }
}

class PointFactory {
    private static Map<String, Point> points = new HashMap<>();

    public static Point getPoint(int x, int y) {
        String key = x + "," + y;
        if (!points.containsKey(key)) { // if point doesn't exist in the map
            points.put(key, new Point(x, y)); // create new point and add it to the map
        }
        return points.get(key);
    }
}

public class Main {
    public static void main(String[] args) {
        Point p1 = PointFactory.getPoint(1, 2);
        Point p2 = PointFactory.getPoint(1, 2);
        Point p3 = PointFactory.getPoint(2, 3);
        System.out.println(p1 == p2); // true
        System.out.println(p1 == p3); // false
    }
}
```

בדוגמה שלמעלה, כאשר אנו יוצרים נקודה חדשה, אנו בודקים האם כבר יש נקודה עם אותם קואורדינטות במפתח של ה-`Map`. אם כן, אנו מחזירים את הנקודה הקיימת, ואם לא, אנו יוצרים נקודה חדשה ומוסיפים אותה למפתח.