Контрольна робота

З дисципліни «Проектування інформаційних систем»

Студентки групи КС31

Паю Анни

Варіант 4

**Prototype**

Суть патерну

Prototype – це породжуючий патерн проектування, який дозволяє копіювати об'єкти, не вдаючись до подробиць реалізації.

Проблема

Ви маєте об'єкт, який потрібно скопіювати. Як це зробити? Потрібно створити порожній об'єкт такого ж класу, а потім по черзі скопіювати значення всіх полів зі старого об'єкта в новий.

Прекрасно! Але є нюанс. Не кожен об'єкт вдасться скопіювати таким чином, адже частина його стану може бути приватною, а отже – недоступною для решти коду програми.

Але є й інша проблема. Копіюючий код буде залежним від класів об'єктів, що копіюються. Адже щоби перебрати всі поля об'єкта, потрібно прив'язатися до його класу. Через це ви не зможете копіювати об'єкти, знаючи лише їх інтерфейси, а не конкретні класи.

Рішення

Паттерн Prototype доручає створення копій самим об'єктам, що копіюються. Він вводить спільний інтерфейс для всіх об'єктів, які підтримують клонування. Це дозволяє копіювати об'єкти, не прив'язуючись до їх класів. Зазвичай такий інтерфейс має лише один метод clone.

Реалізація цього у різних класах дуже схожа. Метод створює новий об'єкт поточного класу і копіює значення всіх полів власного об'єкта. Так вдасться скопіювати навіть приватні поля, оскільки більшість мов програмування дозволяє доступ до приватних полів будь-якого об'єкта поточного класу.

Об'єкт, який копіюють, називається прототипом (звідси назва паттерну). Коли об'єкти програми містять сотні полів та тисячі можливих конфігурацій, прототипи можуть бути своєрідною альтернативою створенню підкласів.

В цьому випадку всі можливі прототипи заготовляються та налаштовуються на етапі ініціалізації програми. Потім, коли програмі потрібен новий об'єкт, вона створює копію з прототипу.

Структура

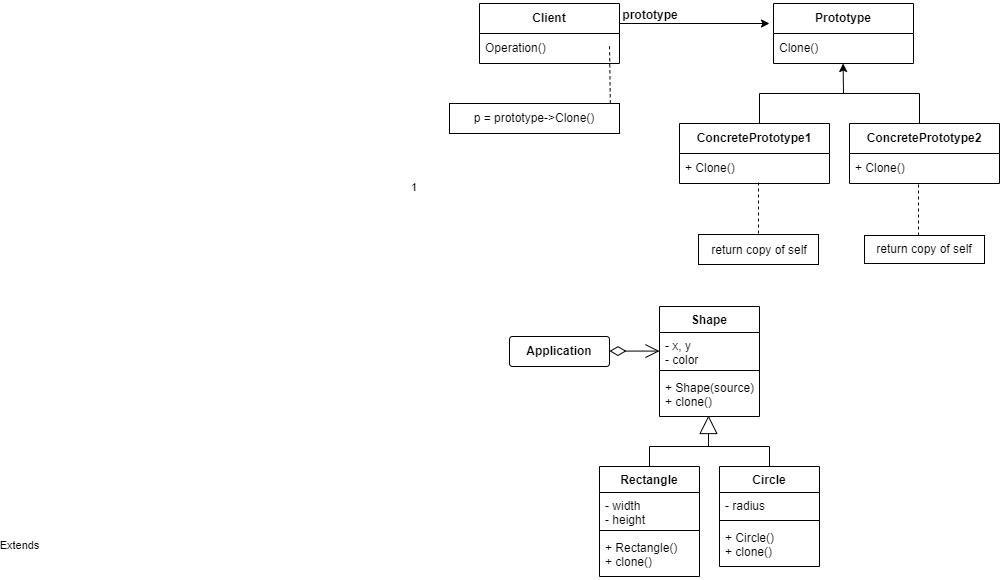


Рисунок 1. Узагальнена UML діаграма класів для паттерну Prototype

1. Інтерфейс прототипів описує операції клонування. Найчастіше – це єдиний метод clone.

2. Конкретний прототип реалізує операцію клонування себе. Крім банального копіювання значень всіх полів, тут можуть бути заховані різні складнощі, про які не потрібно знати клієнту. Наприклад, клонування пов'язаних об'єктів, розплутування рекурсивних залежностей та інше.

3. Клієнт створює копію об'єкта, звертаючись до нього через загальний інтерфейс прототипів.

Приклад конкретного кейсу

У цьому прикладі Prototype дозволяє робити точні копії об'єктів геометричних фігур, не прив'язуючись до класів.

Всі фігури реалізують інтерфейс клонування та надають метод для відтворення самої себе. Підкласи використовують метод клонування предка, а потім копіюють власні поля в об'єкт, що вийшов.

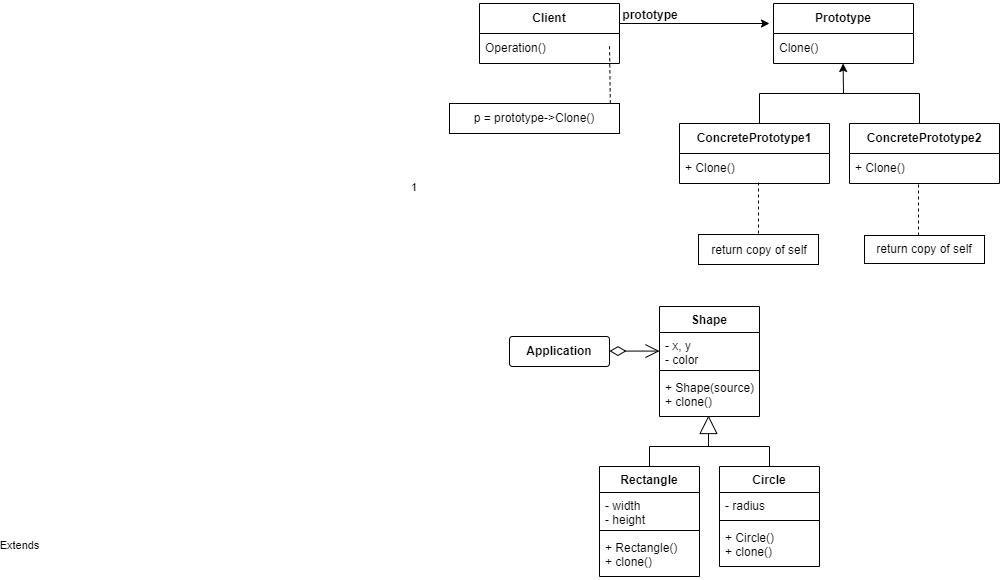


Рисунок 1. UML діаграма класів для даного кейсу

Лістинг розміщено окремим файлом на гіт.

**Результат виконання**

