



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

Лабораторна робота №5  
ШАБЛони «ADAPTER», «BUILDER», «COMMAND»,  
«CHAIN OF RESPONSIBILITY», «PROTOTYPE»  
Варіант 4

Виконав  
студент групи ІА – 13:  
Запотоцький І.А

Перевірив:  
М’який М.Ю

### **Завдання:**

1. Ознайомитися з короткими теоретичними відомостями.
2. Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.
3. Застосування одного з розглянутих шаблонів при реалізації програми.

### **Варіант:**

4)

Графічний редактор (проху, prototype, decorator, bridge, flyweight, SOA) Графічний редактор повинен вміти створювати / редагувати растрові (або векторні на розсуд студента) зображення в 2-3 основних популярних форматах (bmp, png, jpg), мати панель інструментів для створення графічних примітивів, вибору кольорів, нанесення тексту, додавання найпростіших візуальних ефектів (ч/б растр, інфрачервоний растр, 2-3 на вибір учня), роботи з шарами.

## **Хід роботи**

Шаблон проектування "Прототип" (Prototype) є одним із створюючих шаблонів, який дозволяє створювати нові об'єкти, копіюючи стан і структуру існуючого об'єкта, який є прототипом. Використання цього шаблону може значно спростити процес створення нових об'єктів, особливо якщо їхня ініціалізація важка або вони схожі за структурою.

Основні учасники шаблону "Прототип":

Прототип (Prototype): Інтерфейс або абстрактний клас, який визначає метод clone() для створення копії самого себе.

Конкретний прототип (Concrete Prototype): Конкретний клас, який реалізує інтерфейс або наслідує абстрактний клас прототипу. Використовує метод clone(), щоб створити копію себе.

Клієнт (Client): Використовує об'єкт прототипу для створення нових об'єктів шляхом клонування.

Прототип може використовуватися, коли об'єкти, які потрібно створити, мають складну структуру або коли їхня ініціалізація вимагає значних ресурсів. Замість того, щоб створювати новий об'єкт з нуля, можна використовувати існуючий об'єкт прототипу та клонувати його.

```

23 usages
@Data
@NoArgsConstructor
@Getter
@Setter
@Entity
@Table(name = "users")
public class User implements Cloneable{
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String login;
    private String password;

    @Override
    public User clone() {
        try {
            User user = (User) super.clone();
            user.setId(this.getId());
            user.setLogin(this.getLogin());
            user.setPassword(this.getPassword());
            return user;
        } catch (CloneNotSupportedException e) {
            throw new AssertionError();
        }
    }
}

```

В моєму випадку клас User має інтерфейс Cloneable з відповідним реалізованим методом.

Це дає можливість створювати точні копії об'єктів у пам'яті додатка.

Клонування об'єктів має свої переваги та використання, але також потребує обережності, оскільки неправильне клонування може призвести до непередбачуваних результатів. Ось деякі переваги клонування об'єктів:

**Ефективність:** Клонування може бути більш ефективним за створення нового об'єкта та ініціалізацію його значеннями. Це особливо важливо, якщо створення об'єкта вимагає значних ресурсів або займає багато часу.

**Збереження стану:** Клонування дозволяє зберегти стан об'єкта на певний момент часу. Це корисно, наприклад, при реалізації механізму сховища (Memento), коли потрібно зберегти та відновити стан об'єкта.

Легка реалізація копіювальних операцій: Використання клонування спрощує реалізацію копіювальних операцій, оскільки вбудований метод `clone()` надає засіб для створення копії об'єкта.

Гнучкість та розширюваність: Клонування дозволяє динамічно створювати копії об'єктів та розширювати їх, наприклад, за допомогою прототипного програмування. Копіюючи існуючі об'єкти, можна швидко створювати нові об'єкти з необхідними характеристиками.

Зменшення навантаження на ресурси: У деяких випадках клонування може допомогти зменшити навантаження на ресурси, оскільки обхідний об'єкт може бути клонований замість того, щоб створювати новий.