Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп’ютерних наук

Контрольна робота
з навчальної дисципліни
«Проектування інформаційних систем»

Виконав:
Студент групи КС-34
Пархоменко О. А.

Харків – 2022

Шаблон проектування адаптера – це структурний шаблон проектування, що забезпечує спільну роботу двох незв'язаних інтерфейсів.

Реалізуємо програму RPG (рольова гра), використовуючи сторонній API, який має лише інтерфейс Fighter.

Fighter.class

```
public interface Figter {
    public void attack();
    public void defend();
    public void escape();
}
```

І базова реалізація нашого інтерфейсу Fighter.

Knight.class

```
public class Knight implements Fighter {

    @Override
    public void attack() {
       System.out.println("Knight attacks!");
    }

    @Override
    public void defend() {
       System.out.println("Knight defends...");
    }

    @Override
    public void escape() {
       System.out.println("Run Knight run...");
    }
}
```

Через деякий час інтерфейс/клас Wizard був розроблений та доданий до проекту, але ми все одно хочемо використовувати його знову як Fighter. Отже, що нам потрібно зробити, це адаптувати цей клас Wizard у Fighter за допомогою класу адаптера, оскільки Wizard має іншу функціональність, ніж Fighter.

Ось реалізація класу Wizard;

```
public class Wizard {

    public void castLightBeam() {
       System.out.println("Light beam - bshhh!!!");
    }
    public void fireBird() {
      System.out.println("Casted fire bird...");
    }
    public void openPortal() {
       System.out.println("Just choose the wall where you want to open portal");
    }
}
```

Отже, наш клас WizardAdapter реалізує інтерфейс Fighter і викликає відповідні функції класу Wizard.

Wizardadapter.java

```
public class WizardAdapter implements Fighter {

    private Wizard wizard;

    public WizardAdapter(Wizard wizard) {
       this.wizard = wizard;
    }
```

```
    @Override
    public void attack() {
        this.wizard.castLightBeam ();
    }

    @Override
    public void defend() {
        this.wizard.fireBird();
    }

    @Override
    public void escape() {
        this.wizard.openPortal();
    }
```

Наш проект готовий до використання адаптованого класу Wizard.

```
public class Main {
    public static void main(String[] args) {
        Fighter barbarian = new Knight();
        Wizard wizard = new Wizard();
        WizardAdapter wizardAdapter = new WizardAdapter(wizard);

        System.out.println("<----Barbarian's Action----->");
        barbarian.attack();
        barbarian. defend();
        barbarian.escape();

        System.out.println("\n<----Wizard's Action----->");
        wizardAdapter.attack();
        wizardAdapter.defend();
        wizardAdapter.escape();
```

```
    }
}
```

Ось як ми використали шаблон проектування адаптера на нашому прикладі.