Министерство Образования Республики Молдова

Технический Университет Молдовы

Департамент программной инженерии и автоматики

**Отчет**

Лабораторная работа №3

Тема: «Bridge»

Выполнил: TI-197-Герца Николай

Проверил: Булдумаг Олег

Кишинев-2021

**Суть паттерна:**

- Мост — это структурный паттерн проектирования, который разделяет один или несколько классов на две отдельные иерархии — абстракцию и реализацию, позволяя изменять их независимо друг от друга. Заменяет обычное наследование композицией.

**Что будет происходить:**

- Создаю два корневых класса: Model - для описания моделей и TaxiType для описания типа такси. Создаю три подкласса: Econom, Comfort, Business. В своей реализации они работают только типами такси, ничего не зная о моделях машинах. Затем создаю две модели BMW и KIA. Сами модели не работают с типами такси, все что у них есть - это метод, который возвращает информацию о них. В нем они не на прямую работают с типом такси, а идет делегирование работы на класс TaxiType, в котором и описан метод get(). Можно создавать автомобиль такси, используя марку и тип такси.

**Диаграмма UML:**

-



**Реализация:**

-

class Model {

  constructor(carModel) {

    this.carModel = carModel;

  }

}

class TaxiType {

  constructor(type) {

    this.type = type;

  }

  get() {

    return this.type;

  }

}

class Econom extends TaxiType {

  constructor() {

    super("Rate 1$ per km");

  }

}

class Comfort extends TaxiType {

  constructor() {

    super("Rate 2$ per km");

  }

}

class Business extends TaxiType {

  constructor() {

    super("Rate 3$ per km");

  }

}

class BMW extends Model {

  constructor(carModel) {

    super(carModel);

  }

  info() {

    return `Auto: BMW, the cost of travel: ${this.carModel.get()}`;

  }

}

class KIA extends Model {

  constructor(carModel) {

    super(carModel);

  }

  info() {

    return `Auto: KIA, the cost of travel: ${this.carModel.get()}`;

  }

}

const carOne = new BMW(new Business());

const carTwo = new KIA(new Econom());

console.log(carOne.info());

console.log(carTwo.info());



**Применение:**

- Когда вы хотите разделить монолитный класс, который содержит несколько различных реализаций какой-то функциональности (например, если класс может работать с разными системами баз данных);

- Чем больше класс, тем тяжелее разобраться в его коде, и тем больше это затягивает разработку. Кроме того, изменения, вносимые в одну из реализаций, приводят к редактированию всего класса, что может привести к внесению случайных ошибок в код;

- Мост позволяет заменять реализацию даже во время выполнения программы, так как конкретная реализация не «вшита» в класс абстракции.

**Преимущества и недостатки:**

+

- Позволяет строить платформо-независимые программы;

- Скрывает лишние или опасные детали реализации от клиентского кода;

- Реализует принцип открытости/закрытости (разработанная изначально реализация класса в дальнейшем не модифицируется (разве что исправляются ошибки), а любые изменения производятся через создание нового класса, который обычно наследуется от изначального)

-

- Усложняет код программы из-за введения дополнительных классов;