Министерство Образования Республики Молдова

Технический Университет Молдовы

Департамент программной инженерии и автоматики

**Отчет**

Лабораторная работа №3

Тема: «Adapter»

Выполнил: TI-197-Герца Николай

Проверил: Булдумаг Олег

Кишинев-2021

**Суть паттерна:**

- Адаптер — это структурный паттерн проектирования, который позволяет объектам с несовместимыми интерфейсами работать вместе.

**Что будет происходить:**

- Создаем два объекта возможных пользователей такси. Можно сказать, что у премиум пользователя больше возможностей и сложнее этапы обработки заказов. Для того, чтобы сделать из обычного пользователя премиум аккаунт, можно использовать адаптер. Адаптер будет оборачивать стандартный класс и подгонять его под нестандартные классы. Затем создаю класс User, который просто логинится и выводится сообщение о том, какой у него аккаунт.

**Диаграмма UML:**



**Реализация:**

class SimpleUser {

  simpleLog() {

    console.log("Аккаунт данного пользователя - стандартный");

  }

}

class PremiumUser {

  premiumLog() {

    console.log("Аккаунт данного пользователя - преимум");

  }

}

class PremiumUserAdapter {

  constructor(user) {

    this.user = user;

  }

  simpleLog() {

    this.user.premiumLog();

  }

}

class User {

  login(user) {

    user.simpleLog();

  }

}

const simple\_user = new User();

const old\_acc = new SimpleUser();

simple\_user.login(old\_acc); //Аккаунт данного пользователя - стандартный

const varAdapter = new PremiumUserAdapter(new PremiumUser());

simple\_user.login(varAdapter); //Аккаунт данного пользователя - преимум

const wrongAdapter = new PremiumUser();

//simple\_user.login(wrongAdapter) //error

**Применение:**

- когда вы хотите использовать сторонний класс, но его интерфейс не соответствует остальному коду приложения;

- когда вам нужно использовать несколько существующих подклассов, но в них не хватает какой-то общей функциональности, причём расширить суперкласс вы не можете.

**Преимущества:**

- отделяет и скрывает от клиента подробности преобразования различных интерфейсов.

**Недостатки:**

- усложняет код программы из-за введения дополнительных классов.