Министерство науки и высшего образования Российской Федерации ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО ITMO University**

# ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

**По дисциплине** Проектирование программного обеспечения

**Обучающийся** Ряднов Вениамин Сергеевич

**Факультет** факультет прикладной информатики (ФПИн)

**Группа** К3250

**Направление подготовки** 09.03.03 Прикладная информатика

**Образовательная программа** Прикладная разработка информационных систем

|  |  |  |  |
| --- | --- | --- | --- |
| Обучающийся | (дата) | \_\_\_\_\_\_\_\_ (подпись) | Ряднов В.С. (Ф.И.О.) |
| Руководитель | (дата) | \_\_\_\_\_\_\_\_ (подпись) | Меркушев А.Е. (Ф.И.О.) |

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1 Ход работы 4

1.1 GRASP 4

1.1.1 Информационный эксперт (Information Expert) 4

1.1.2 Создатель (Creator) 4

1.1.3 Контроллер (Controller) 4

1.1.4 Слабая связанность (Low Coupling) 4

1.1.5 Высокая связность (High Cohesion) 4

1.1.6 Полиморфизм (Polymorphism) 4

1.1.7 Абстракция зависимости (Pure Fabrication) 4

1.1.8 Иерархия (Indirection) 4

1.1.9 Защита вариаций (Protected Variations) 5

1.2 Диаграмма взаимодействия 5

1.2.1 Определение пользователей 5

1.2.2 Случай использования - заказ такси 5

2 GOF 6

2.1 Классификация шаблонов проектирования GoF 6

2.1.1 Порождающие шаблоны (Creational Patterns) 6

2.1.2 Структурные шаблоны (Structural Patterns) 6

2.1.3 Поведенческие шаблоны (Behavioral Patterns) 6

ЗАКЛЮЧЕНИЕ 7

# 

# ВВЕДЕНИЕ

Этот отчёт посвящён разработке и анализу архитектуры автоматизированной системы “Обмен заказами такси”. В ходе работы будут рассмотрены основные принципы GRASP и GoF, которые применяются для построения программного обеспечения, а также приведены диаграммы взаимодействий и классов. Разработанная архитектура ориентирована на улучшение гибкости, масштабируемости и надёжности системы.

# 

# 1. ХОД РАБОТЫ

## 1.1 GRASP

### 1.1.1 Информационный эксперт (Information Expert)

Этот шаблон отвечает за назначение обязанностей объектам, которые обладают необходимой информацией для выполнения задачи. Применяется к модулю управления заказами для хранения и обработки данных о клиентах, заказах и водителях.

### 1.1.2 Создатель (Creator)

Создаёт объекты заявок и управляет их передачей в службу распределения. Модуль обеспечивает создание и сохранение информации о новых заказах в базе данных.

### 1.1.3 Контроллер (Controller)

Управляет запросами пользователей, контролирует обработку и статусы заявок. Он связывает интерфейс пользователя с логикой системы и обрабатывает входящие команды.

### 1.1.4 Слабая связанность (Low Coupling)

Минимизирует зависимости между модулями системы, что облегчает внесение изменений. В системе используется разделение модулей, чтобы каждый выполнял строго свою задачу.

### 1.1.5 Высокая связность (High Cohesion)

Обеспечивает, что каждый модуль решает одну конкретную задачу. Например, модуль отчётов работает только с аналитикой и не взаимодействует с логикой обработки заказов.

### 1.1.6 Полиморфизм (Polymorphism)

Поддерживает возможность добавления новых типов заказов без изменения существующего кода, что достигается за счёт наследования и интерфейсов.

### 1.1.7 Абстракция зависимости (Pure Fabrication)

Создание вспомогательных классов для обработки данных заказов, отделённых от основной бизнес-логики, что упрощает модификацию системы.

### 1.1.8 Иерархия (Indirection)

Добавляет уровни взаимодействия между модулями для упрощения связи и защиты от изменений.

### 1.1.9 Защита вариаций (Protected Variations)

Предотвращает влияние изменений на работу системы путём использования интерфейсов и абстрактных классов.

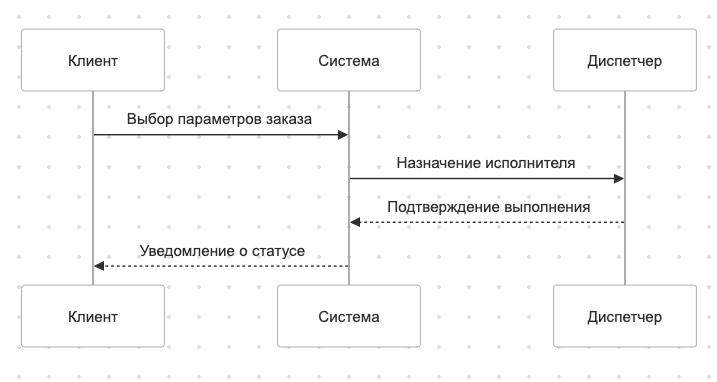
## 1.2 Диаграмма взаимодействия

### 1.2.1 Определение пользователей

Пользователи включают диспетчеров, клиентов и водителей, которые взаимодействуют с системой для создания и обработки заказов.

### 1.2.2 Случай использования - заказ такси

Клиент создаёт заказ, который распределяется системой между доступными службами такси. Водитель подтверждает заказ, а после выполнения данные передаются в отчёт.



# 2. GOF

## 2.1 Классификация шаблонов проектирования GoF

### 2.1.1 Порождающие шаблоны (Creational Patterns)

**Singleton**: Управляет аналитикой данных о заказах.

### 2.1.2 Структурные шаблоны (Structural Patterns)

**Adapter**: Работает с разными API служб такси.

### 2.1.3 Поведенческие шаблоны (Behavioral Patterns)

**Observer**: Уведомляет о статусах заказов.

# ЗАКЛЮЧЕНИЕ

Система “Обмен заказами такси” построена с применением GRASP и GoF. Рассмотрены диаграммы и сценарии взаимодействий, обеспечивающие масштабируемость и надёжность.