МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

КУРСОВАЯ РАБОТА (ПРОЕКТ)   
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ст. преподаватель |  |  |  | Е. О. Шумова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ |
| РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ОРГАНИЗАЦИИ ВЗАИМОДЕЙСТВИЯ ОБЪЕКТОВ ПРИ ЗАДАННЫХ КРИТЕРИЯХ |
| по дисциплине: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Д. В. Самарин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

Оглавление

[1. Введение 3](#_Toc148565734)

[2. Постановка задачи 3](#_Toc148565735)

[2.1. Анализ предметной области 3](#_Toc148565736)

[2.2. Формулировка технического задания 4](#_Toc148565737)

[3. Разработка классов 6](#_Toc148565738)

[3.1. Классы сущностей: 6](#_Toc148565739)

[3.2. Управляющие классы: 8](#_Toc148565740)

[3.3. Интерфейсные классы: 9](#_Toc148565741)

[3.4. Используемые паттерны проектирования (их обоснования): 9](#_Toc148565742)

[3.5. Диаграмма классов: 10](#_Toc148565743)

[4. Разработка приложения 13](#_Toc148565744)

[5. Тестирование 13](#_Toc148565745)

[6. Приложение с полным кодом программ 13](#_Toc148565746)

[7. Заключение 13](#_Toc148565747)

[8. Список использованных источников 13](#_Toc148565748)

# Введение

В ходе курсового проекта необходимо разработать приложение,

позволяющее протестировать взаимодействие объектов классов,

спроектированных и реализованных мной для решения конкретной

задачи при заданных критериях.

При разработке программного обеспечения следует использовать

шаблоны проектирования. Также предпочтение должно быть отдано

графическому приложению.

Целями курсового проекта являются:

- закрепление знаний, полученных в ходе изучения дисциплины

«Объектно-ориентированное программирование»;

- приобретение навыков практического программирования с

использованием объектно-ориентированной парадигмы;

- подготовка к выполнению выпускной квалификационной работы.

Задачей курсового проекта является разработка иерархии типов в

заданной предметной области, включающей в себя:

- разработку иерархии классов

- разработку приложения

- подготовку презентации.

Актуальность проекта на высоком уровне. В наше время существует множество дилерских центров. Подобные приложения делают жизнь подобных организаций проще.

Ниже будет расписана основная часть пояснительной записки, которая включается в себя: постановку задачи, разработку классов, разработку приложения, тестирование.

# Постановка задачи

Данная курсовая работа по программированию направлена на разработку системы классов, которая будет описывать различные транспортные средства, управляемые дилерским центром. Основной целью проектирования является создание программного продукта, который позволит учитывать закупки различных брендов транспортных средств от производителей, а также фиксировать данные о продажах и покупателях.

## 2.1. Анализ предметной области

Предметной областью является управление транспортными средствами в дилерском центре. Основными сущностями предметной области являются транспортные средства (автомобили, мотоциклы и другие), производители, дилерский центр, покупатели, операции закупки и продажи.

Словарь предметной области:

- Транспортное средство

- Производитель

- Дилерский центр

- Покупатель

- Закупка

- Продажа

Функциональные требования:

- Ведение данных о закупках транспортных средств от различных производителей.

- Фиксация данных о продажах транспортных средств покупателям.

- Сохранение информации о производителях, моделях, характеристиках транспортных средств.

- Учет данных о покупателях, их контактных данных.

- Расчет стоимости транспортных средств с учетом скидок и наценок.

- Поиск и фильтрация данных о транспортных средствах и продажах.

## 2.2. Формулировка технического задания

Наименование проекта: разработка системы классов для учета транспортных средств в дилерском центре.

Задача проекта:

Разработать систему классов, предназначенную для учета и управления данными о различных транспортных средствах, которые закупаются дилерским центром от различных производителей. Система должна обеспечивать учет процессов закупки, продажи транспортных средств, а также хранение данных о покупателях.

Описание проекта:

Необходимо создать программную систему, которая позволит дилерскому центру вести учет закупок различных брендов транспортных средств у производителей и осуществлять продажу этих транспортных средств конечным покупателям. Система должна предоставлять следующие функциональные возможности:

Учет закупок:

Регистрация данных о закупках транспортных средств от производителей.

Хранение информации о поставщиках и поставках, включая даты, количество и характеристики транспортных средств.

Учет продаж:

Фиксация данных о продажах транспортных средств конечным покупателям.

Отслеживание информации о покупателях, включая их контактные данные и историю покупок.

Управление данными о транспортных средствах:

Хранение характеристик каждого транспортного средства, включая модель, год выпуска, цену и другие характеристики.

Возможность редактирования и обновления данных о транспортных средствах.

Отчетность:

Создание отчетов о текущем состоянии склада транспортных средств.

Формирование отчетов о продажах и доходах дилерского центра.

Технические требования:

Разработка системы должна быть выполнена на QTCreator.

Система должна быть разделена на классы, обеспечивающие модульность и расширяемость.

Интерфейс пользователя должен быть интуитивно понятным и обеспечивать доступ к функциональным возможностям системы.

Итогом проделанных работ станет: разработка и внедрение системы классов для учета транспортных средств, обеспечивающей эффективное управление данными о закупках, продажах и клиентах дилерского центра.

# 3. Разработка классов

Для разработки иерархии классов мы начнем с выделения основных сущностей предметной области и определения классов, описывающих эти сущности. Затем мы определим управляющие классы и интерфейсные классы для организации взаимодействия между ними и с внешней средой. Ниже приведена детальная разработка иерархии классов.

## 3.1. Классы сущностей:

1. Транспортное средство (Vehicle):

- Поля:

- ID

- Марка

- Модель

- Год выпуска

- Стоимость

- Описание

- Статус (доступно, продано, зарезервировано и т.д.)

- Методы:

- Конструктор для создания нового транспортного средства

- Методы для получения и установки значений полей

- Метод для изменения статуса

2. Производитель (Manufacturer):

- Поля:

- ID

- Название

- Страна происхождения

- Методы:

- Конструктор для создания нового производителя

- Методы для получения и установки значений полей

3. Дилерский центр (Dealership):

- Поля:

- ID

- Название

- Адрес

- Список доступных транспортных средств

- Методы:

- Конструктор для создания нового дилерского центра

- Методы для добавления и удаления транспортных средств

- Методы для получения информации о транспортных средствах в наличии

4. Покупатель (Customer):

- Поля:

- ID

- Имя

- Фамилия

- Контактная информация

- Методы:

- Конструктор для создания нового покупателя

- Методы для получения и установки значений полей

5. Операция продажи (Sale):

- Поля:

- ID

- Дата продажи

- Продавец (сотрудник дилерского центра)

- Покупатель

- Транспортное средство

- Сумма продажи

- Методы:

- Конструктор для создания новой операции продажи

- Методы для получения и установки значений полей

## 3.2. Управляющие классы:

1. Склад (Inventory):

- Поля:

- Список всех доступных транспортных средств

- Методы:

- Методы для добавления и удаления транспортных средств

- Методы для поиска транспортных средств по различным критериям (марка, модель, статус и т.д.)

2. Продажи (SalesManager):

- Поля:

- Список всех операций продажи

- Методы:

- Методы для регистрации новой операции продажи

- Методы для отчетности по продажам

## 3.3. Интерфейсные классы:

1. Графический интерфейс (GUI):

- Методы:

- Методы для взаимодействия с пользователем, включая ввод и вывод данных.

Обоснования применения классов в нашей будущей программе:

- Использование классов сущностей для описания базовых объектов предметной области позволяет удобно хранить и обрабатывать информацию о транспортных средствах, производителях, дилерских центрах, покупателях и операциях продажи.

- Управляющие классы, такие как Склад и Продажи, позволяют организовать управление данными и операциями, связанными с транспортными средствами и продажами.

- Интерфейсный класс GUI обеспечивает взаимодействие пользователя с программой через графический интерфейс, что делает использование приложения более удобным для конечных пользователей.

## 3.4. Используемые паттерны проектирования (их обоснования):

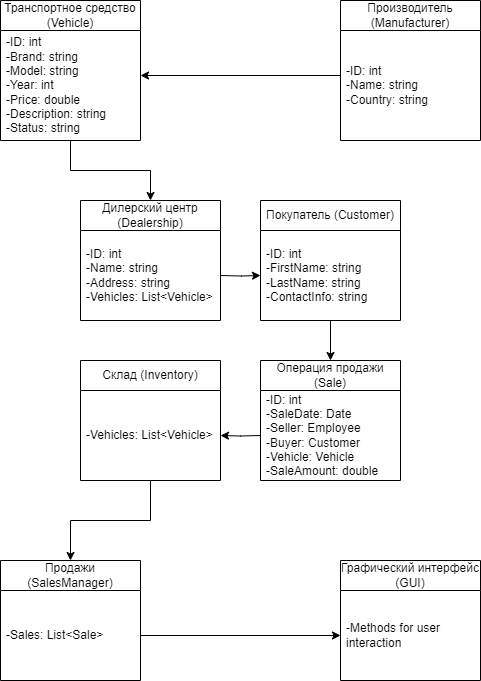
1. Фабричный метод (Factory Method): Паттерн Factory Method будет применен для создания экземпляров транспортных средств. Каждый производитель может иметь свой фабричный метод, который создает экземпляры транспортных средств конкретного бренда.

2. Одиночка (Singleton): Паттерн Singleton будет использован для создания и управления объектом базы данных или же для управления доступом к хранилищу данных о транспортных средствах и продажах. Это позволит обеспечить единственный экземпляр класса для доступа к данным из разных частей приложения.

3. Наблюдатель (Observer): Паттерн Observer будет применен для реализации механизма уведомления дилерского центра о новых закупках транспортных средств от производителей или о новых операциях продажи. Дилерский центр может быть наблюдателем, а производители и менеджеры продаж могут быть субъектами.

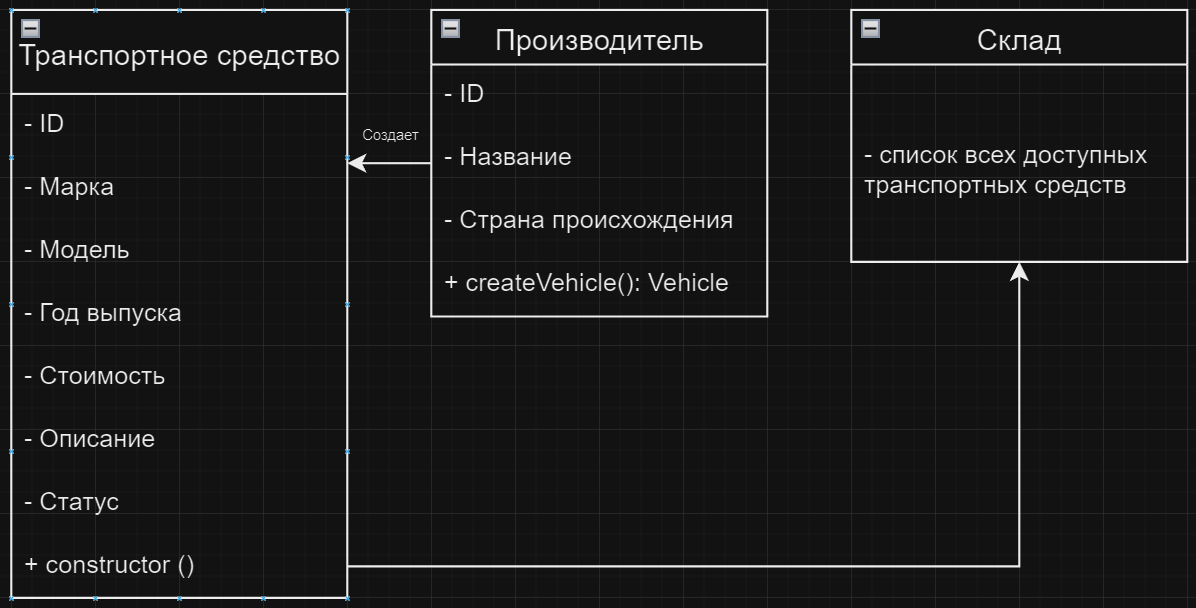
4. Команда (Command): Паттерн Command может будет использован для реализации истории операций продажи, которую можно отменять и восстанавливать. Каждая операция продажи может быть представлена командой.

## 3.5. Диаграмма классов:



Помимо этого, рассмотрим диаграммы для наших будущих паттернов:

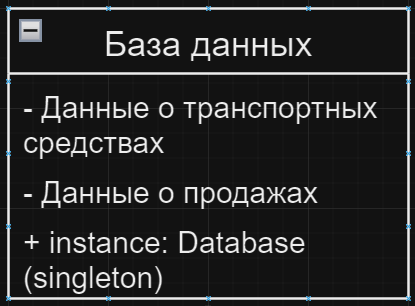
Factory Method:



Стрелка от "Производитель" к "Транспортное средство" указывает на то, что "Производитель" создает экземпляры "Транспортного средства" с помощью Factory Method.

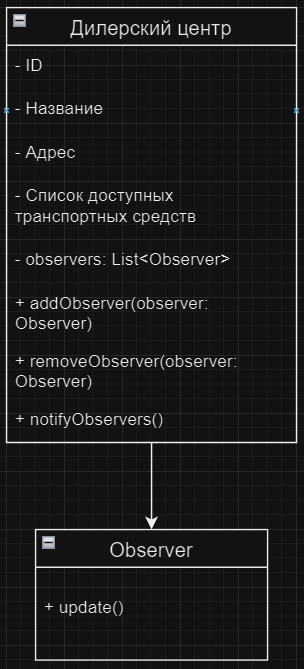
Стрелка от "Транспортное средство" к "Склад" указывает, что экземпляры "Транспортного средства" добавляются или удаляются из "Склада" через методы класса "Склад (Inventory)".

Singleton:



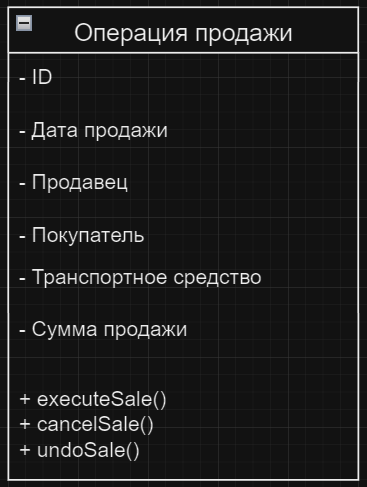
Здесь паттерн Singleton применяется для класса Database, чтобы обеспечить единственный экземпляр базы данных для доступа к данным о транспортных средствах и продажах.

Observer:



Здесь паттерн Observer применяется для класса Dealership, который может иметь наблюдателей (например, производителей и менеджеров продаж), и уведомлять их об изменениях в списке доступных транспортных средств.

Command:



Здесь паттерн Command применен для класса Sale, где каждая операция продажи может быть представлена как команда, которую можно выполнять, отменять и восстанавливать.

Таким образом созданная иерархия классов и применяемые паттерны позволят эффективно управлять данными и операциями, связанными с транспортными средствами и продажами в дилерском центре, а также обеспечит удобный интерфейс для пользователей.

# 4. Разработка приложения

# 5. Тестирование

# 6. Приложение с полным кодом программ

# 7. Заключение

# 8. Список использованных источников