

**Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”**

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1
«Разработка и оформление технического задания»**

Специальность 09.02.07 «Информационные системы и программирование»

ПМ.02 «Осуществление интеграции программных модулей»

МДК.2.1 «Технология разработки программного обеспечения»

Тема 2.1.1 «Основные понятия и стандартизация требований к программному обеспечению»

Преподаватель:

Коцюба И.Ю.

«15» октября 2025г.

Оценка:

Выполнил:

студент группы К3221

Доценников Н.А.

Санкт-Петербург
2025/2026

Цель работы.

Закрепить теоретические знания в вопросах создания технического задания и освоить применение паттернов проектирования (на примере паттерна «Фасад»).

Задачи.

- Ознакомиться с классификацией паттернов проектирования.
- Изучить особенности применения паттерна «Фасад» в UML-нотации.
- Разработать техническое задание на создание системы заметок с унифицированным интерфейсом доступа к функционалу.
- Сформировать требования к системе и представить выводы.

Индивидуальная тема.

«Проектирование системы заметок с использованием паттерна Фасад»

Общие сведения.

1.1. Полное наименование системы:

Информационная система «NoteManager» (система заметок).

1.2. Разработчик:

Студент группы К3221, Доценников Н.С.

1.3. Основания для проведения работ:

Учебный план дисциплины «Проектирование инфокоммуникационных систем».

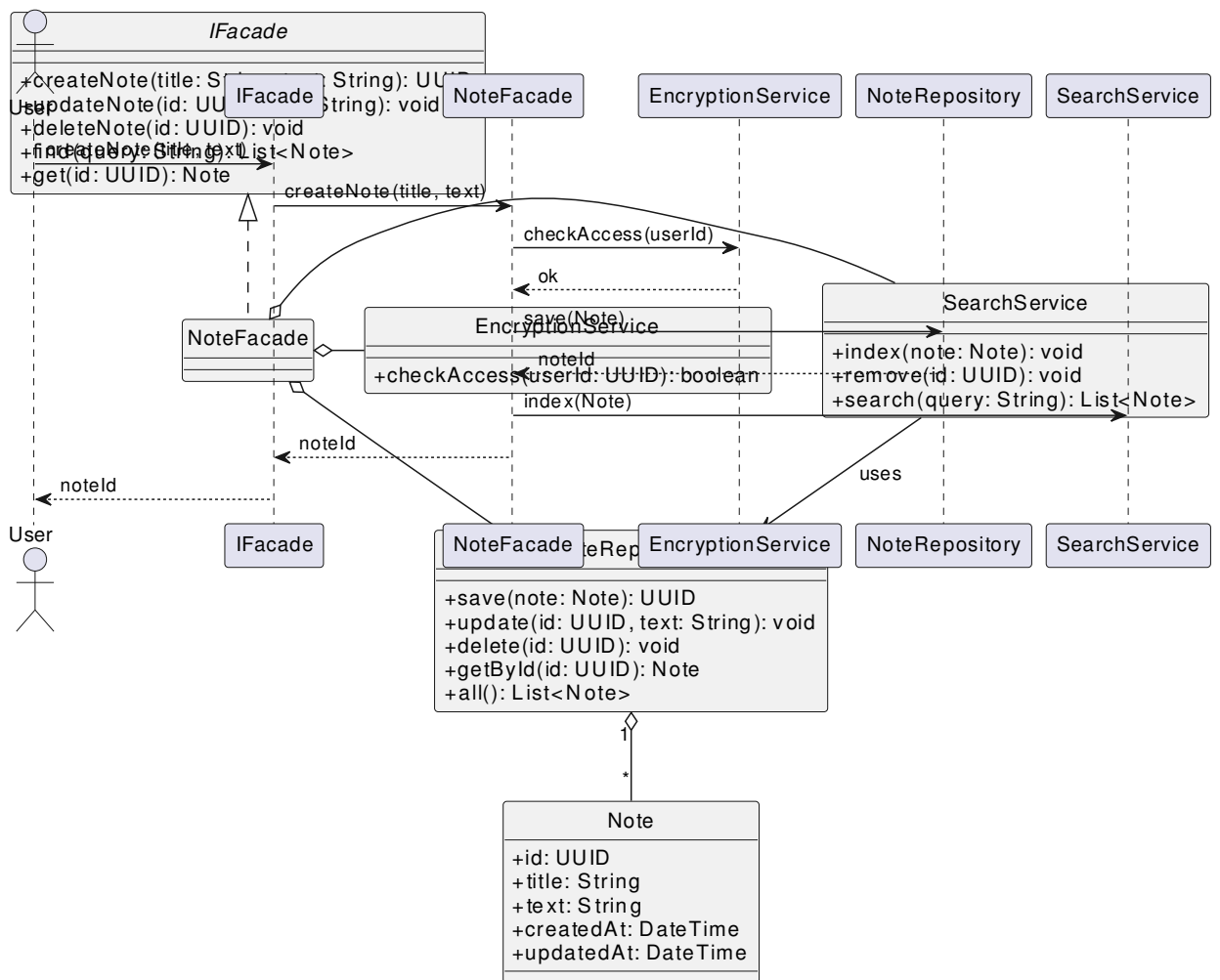
1.4. Плановые сроки:

Начало – октябрь 2025 г., окончание – ноябрь 2025 г.

1.5. Финансирование:

Учебная работа, без финансирования.

UML-диаграмма (паттерн «Фасад»).



Требования к системе.

1.1 Требования в целом:

Система должна предоставлять единый интерфейс для работы с заметками (создание, редактирование, удаление, поиск).

1.2. Требования к функциям:

- Создание текстовой заметки.
- Редактирование содержимого.
- Удаление заметки.
- Поиск заметки по ключевым словам.

1.3. Требования к защите информации:

Система должна поддерживать парольную защиту учётной записи пользователя.

1.4. Требования к эргономике:

Простой и удобный интерфейс, доступный через единое меню (реализация паттерна «Фасад»).

Вывод.

В ходе лабораторной работы было разработано техническое задание на систему заметок. Для упрощения доступа к функционалу предложено применение паттерна проектирования «Фасад». Это позволило выделить унифицированный интерфейс, обеспечивающий удобство использования и расширяемость системы.

Приложение.

Класс-диаграмма.

@startuml

hide circle

skinparam classAttributeIconSize 0

```
interface IFacade {  
    +createNote(title: String, text: String): UUID  
    +updateNote(id: UUID, text: String): void  
    +deleteNote(id: UUID): void  
    +find(query: String): List<Note>  
    +get(id: UUID): Note  
}
```

```
class NoteFacade implements IFacade
```

```
class NoteRepository {  
    +save(note: Note): UUID  
    +update(id: UUID, text: String): void
```

```

+delete(id: UUID): void
+getById(id: UUID): Note
+all(): List<Note>
}
class SearchService {
+index(note: Note): void
+remove(id: UUID): void
+search(query: String): List<Note>
}
class EncryptionService {
+checkAccess(userId: UUID): boolean
}
class Note {
+id: UUID
+title: String
+text: String
+createdAt: DateTime
+updatedAt: DateTime
}

```

NoteFacade o- NoteRepository

NoteFacade o- SearchService

NoteFacade o- EncryptionService

SearchService --> NoteRepository : uses

NoteRepository "1" o-- "*" Note

@enduml

Диаграмма последовательности.

@startuml

actor User

participant IFacade

participant NoteFacade

participant EncryptionService

participant NoteRepository

participant SearchService

User -> IFacade: createNote(title, text)

IFacade -> NoteFacade: createNote(title, text)

NoteFacade -> EncryptionService: checkAccess(userId)

EncryptionService --> NoteFacade: ok

NoteFacade -> NoteRepository: save(Note)

NoteRepository --> NoteFacade: noteId

NoteFacade -> SearchService: index(Note)

NoteFacade --> IFacade: noteId

IFacade --> User: noteId

@enduml