

Автономная некоммерческая организация «Профессиональная образовательная организация «Московский Международный Колледж»

Специальность: 09.02.07 Информационные системы и программирование  
Дисциплина: Операционные системы

## **ОТЧЕТ**

### **Практическая работа № 4**

**На тему: Уточнение диаграмм последовательностей и диаграмм прецедентов с помощью цветового кодирования**

Выполнил студент группы: И-9-23 \_\_\_\_\_ Умаров Диёрбек

Оценка:

Проверил(-а): \_\_\_\_\_ Лихторенко О. С.

г. Москва

2024 г.

**Краткая теория:**

**Цветовое кодирование** — это метод визуализации информации, который использует различные цвета для обозначения различных категорий данных или процессов. В контексте UML-диаграмм, таких как диаграммы последовательностей и диаграммы прецедентов, цветовое кодирование может значительно улучшить восприятие и понимание информации.

#### **Легенда с пояснениями:**

**Диаграмма последовательностей** используется для отображения взаимодействий между объектами во времени. Цветовое кодирование в таких диаграммах может выполнять несколько функций:

- Идентификация участников: разные цвета могут быть использованы для обозначения различных объектов или участников взаимодействия. Например, все объекты пользователя могут быть окрашены в один цвет (например, зеленый), а объекты системы — в другой (например, синий). Это помогает быстро различать роли участников.
- Типы сообщений: цвета могут также использоваться для различия типов сообщений. Например, можно использовать красный цвет для ошибок или исключений, синий — для обычных запросов и ответов, а зеленый — для успешных завершений операций. Это позволяет визуально выделить важные моменты в процессе взаимодействия.
- Группировка сообщений: в случае сложных диаграмм с множеством взаимодействий можно применять цветовое кодирование для группировки сообщений по смыслу. Например, все сообщения, относящиеся к одной функциональной группе, могут быть окрашены в один цвет.

**Диаграмма прецедентов** (или use case diagram) отображает взаимодействие между актерами и системой, показывая функциональные требования к системе. Цветовое кодирование здесь также может быть весьма полезным:

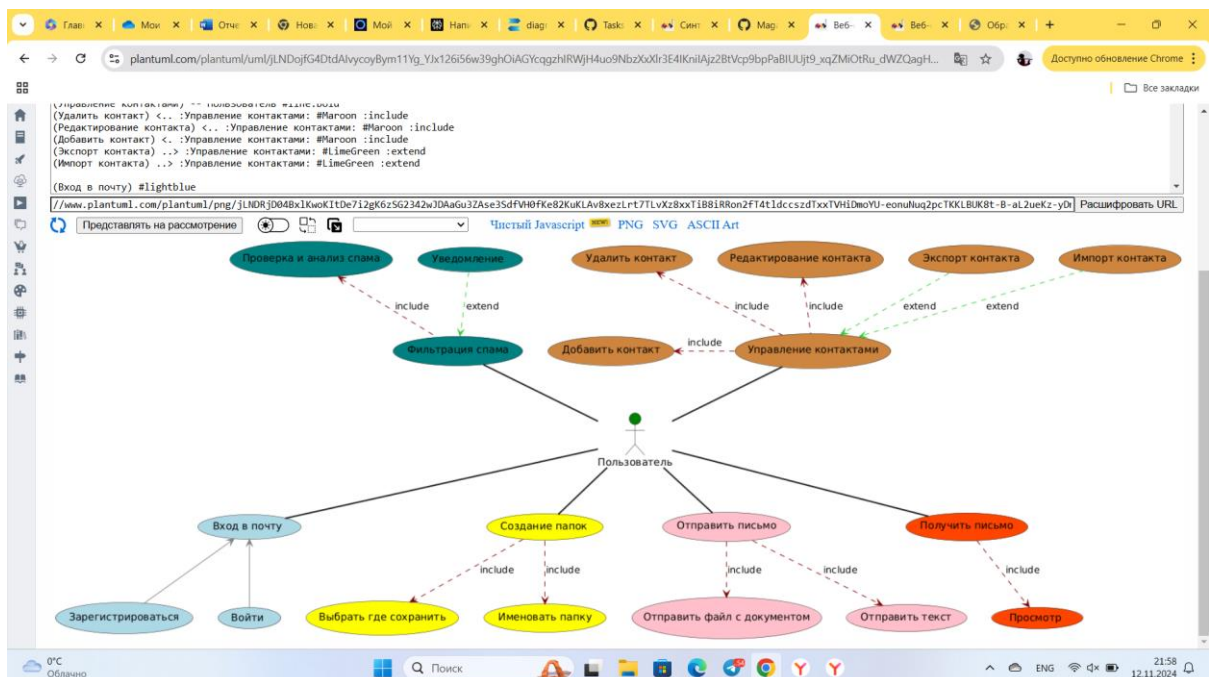
- Классификация акторов: разные цвета могут обозначать разные типы акторов. Например, внешние пользователи могут быть выделены одним цветом, а внутренние пользователи — другим. Это упрощает понимание структуры системы и ее пользователей.
- Типы прецедентов: цвета могут использоваться для различения типов прецедентов: основные (основные функции системы) могут быть одного цвета, альтернативные сценарии — другого, а

- Степень важности: можно использовать оттенки одного цвета для обозначения степени важности или приоритета прецедентов. Например, более темные оттенки могут указывать на более критичные функции системы.

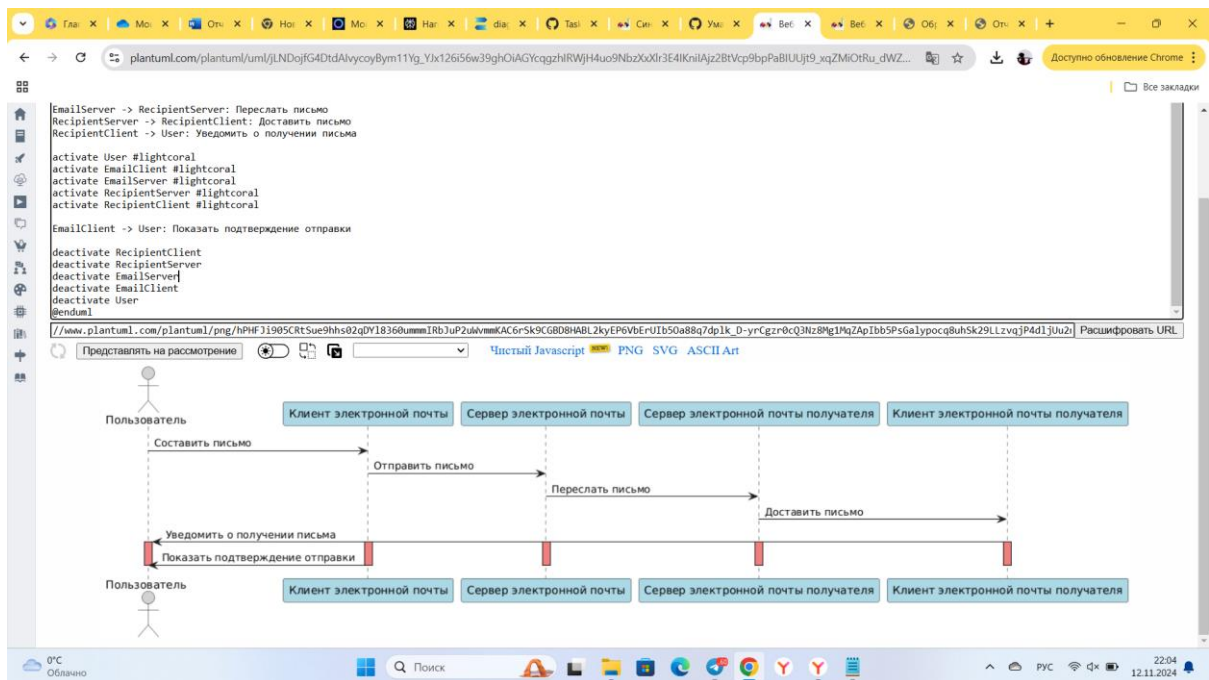
Научиться построению диаграммы прецедентов и диаграммы последовательности с помощью цветного кодирования.

- 1.Изучение теоретического материала и легенд с пояснением.
- 2.Обзор на синтаксис и особенностей.
- 3.Обзор на онлайн — инструменты.
- 4.Практическая часть.

Диаграмма прецедентов:



## Диаграмма последовательности:



## Листинги:

### Код прецендентов:

@startuml

actor Пользователь #green

Пользователь -- (Вход в почту) #line.bold

:Вход в почту: <-- (Войти) #Grey

:Вход в почту: <-- (Зарегистрироваться) #Grey

Пользователь -- (Создание папок) #line.bold

:Создание папок: ..> (Именовывать папку) #Maroon :include

:Создание папок: ..> (Выбрать где сохранить) #Maroon :include

Пользователь -- (Отправить письмо) #line.bold

:Отправить письмо: ..> (Отправить текст) #Maroon :include

:Отправить письмо: ..> (Отправить файл с документом) #Maroon :include

Пользователь -- (Получить письмо) #line.bold  
:Получить письмо: ..> (Просмотр) #Maroon :include  
(Фильтрация спама) -- Пользователь #line.bold  
(Проверка и анализ спама) <.. :Фильтрация спама: #Maroon :include  
(Уведомление) ..> :Фильтрация спама: #LimeGreen :extend  
(Управление контактами) -- Пользователь #line.bold  
(Удалить контакт) <.. :Управление контактами: #Maroon :include  
(Редактирование контакта) <.. :Управление контактами: #Maroon :include  
(Добавить контакт) <. :Управление контактами: #Maroon :include  
(Экспорт контакта) ..> :Управление контактами: #LimeGreen :extend  
(Импорт контакта) ..> :Управление контактами: #LimeGreen :extend  
(Вход в почту) #lightblue  
(Войти) #lightblue  
(Зарегистрироваться) #lightblue  
(Создание папок) #yellow  
(Именовывать папку) #yellow  
(Выбрать где сохранить) #yellow  
(Отправить письмо) #pink  
(Отправить текст) #pink  
(Отправить файл с документом) #pink  
(Получить письмо) #OrangeRed  
(Просмотр) #OrangeRed  
(Фильтрация спама) #Teal  
(Проверка и анализ спама) #Teal  
(Уведомление) #Teal

(Управление контактами) #Peru  
(Удалить контакт) #Peru  
(Редактирование контакта) #Peru  
(Добавить контакт) #Peru  
(Экспорт контакта) #Peru  
(Импорт контакта) #Peru  
@enduml

Код последовательности:

```
@startuml
actor "Пользователь" as User #lightgray
participant "Клиент электронной почты" as EmailClient #lightblue
participant "Сервер электронной почты" as EmailServer #lightblue
participant "Сервер электронной почты получателя" as RecipientServer #lightblue
participant "Клиент электронной почты получателя" as RecipientClient #lightblue

User -> EmailClient: Составить письмо
EmailClient -> EmailServer: Отправить письмо
EmailServer -> RecipientServer: Переслать письмо
RecipientServer -> RecipientClient: Доставить письмо
RecipientClient -> User: Уведомить о получении письма

activate User #lightcoral
activate EmailClient #lightcoral
activate EmailServer #lightcoral
activate RecipientServer #lightcoral
```

activate RecipientClient #lightcoral

EmailClient -> User: Показать подтверждение отправки

deactivate RecipientClient

deactivate RecipientServer

deactivate EmailServer

deactivate EmailClient

deactivate User

@enduml

### **Вывод:**

В ходе работы я приобрел умение написания диаграммы прецедентов и диаграммы последовательности с помощью цветового кодирования в специализированном языке UML.