Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

**ПРАКТИЧЕСКАЯ РАБОТА №6**

**Специальность: «09.02.07 Информационные системы и программирование*»***

Квалификация: программист

**Тема**: Построение диаграммы классов UML при проектировании программного обеспечения.

|  |  |  |
| --- | --- | --- |
| Проверил преподаватель: | Выполнил: | Выполнил: |
| Волков Р.И. \_\_\_\_\_\_\_\_ | Студент группы П50-4-21 | Студент группы П50-1-18 |
| «\_\_\_» \_\_\_\_\_\_\_\_ 2024 год | \_\_\_\_\_\_\_\_\_\_\_\_ Игошев Р.В. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_И.И. Иванов |
| «\_\_\_» \_\_\_\_\_\_\_\_\_2024 год | | «\_\_\_» \_\_\_\_\_\_\_\_\_2020 год | |

Москва 2024

Цель работы: закрепить теоретические знания по принципам создания диаграммы классов. Получить практические навыки по построению диаграммы классов.

1) Составить перечень правил языка UML.

1. Классы и объекты:
   * Класс изображается в виде прямоугольника, разделенного на секции: имя класса, атрибуты, методы.
   * Объекты — это экземпляры классов, обозначаются с подчеркиванием имени.
2. Атрибуты и методы:
   * Атрибуты описывают свойства класса.
   * Методы описывают поведение класса.
3. Отношения между классами:
   * Ассоциация: Связь между классами, обозначается линией со стрелкой.
   * Наследование: Обозначается стрелкой с закрашенным треугольником.
   * Реализация интерфейса: Обозначается пунктирной стрелкой с закрашенным треугольником.
   * Агрегация: Обозначается ромбом на конце линии.
   * Композиция: Обозначается закрашенным ромбом на конце линии.
   * Зависимости: Обозначается пунктирной стрелкой с закрашенным треугольником в конце линии.
4. Многократность:
   * Указывает количество объектов, участвующих в отношении.
   * Например, 1..\* означает один или более объектов.
5. Модификаторы доступа:
   * + (public)
   * - (private)
   * # (protected)
6. Статические элементы:
   * Статические атрибуты и методы подчеркиваются.
7. Абстрактные классы и методы:
   * Имена абстрактных классов и методов записываются курсивом.
8. **Пакеты:**
   * Пакеты используются для организации классов и других элементов в группы.
   * Пакет изображается в виде папки с именем пакета.
   * Пакеты могут содержать другие пакеты, классы, интерфейсы и т.д.
9. **Интерфейсы:**
   * Интерфейс изображается в виде прямоугольника с именем интерфейса и ключевым словом «interface».
   * Интерфейс может содержать только методы (без атрибутов).
   * Реализация интерфейса обозначается пунктирной стрелкой с закрашенным треугольником.
10. **Последовательные диаграммы:**
    * Последовательные диаграммы описывают взаимодействие объектов во времени.
    * Объекты изображаются в виде вертикальных линий (жизненных линий).
    * Сообщения между объектами изображаются в виде горизонтальных стрелок.
11. **Диаграммы состояний:**
    * Диаграммы состояний описывают поведение объекта в зависимости от его состояния.
    * Состояния изображаются в виде кружков с именем состояния.
    * Переходы между состояниями изображаются в виде стрелок с условием перехода.
12. **Диаграммы компонентов:**
    * Диаграммы компонентов описывают структуру системы в терминах компонентов и их взаимосвязей.
    * Компонент изображается в виде прямоугольника с двумя маленькими прямоугольниками по бокам.
    * Взаимосвязи между компонентами изображаются в виде линий со стрелками.
13. **Диаграммы развертывания:**
    * Диаграммы развертывания описывают физическое размещение компонентов системы на узлах.
    * Узлы изображаются в виде кубов.
    * Компоненты размещаются на узлах, и их связи изображаются в виде линий со стрелками.
14. **Аннотации:**
    * Аннотации используются для добавления комментариев или пояснений к элементам диаграммы.
    * Аннотация изображается в виде прямоугольника с линией, соединяющей его с элементом диаграммы.
15. **Параметры методов:**
    * Методы могут иметь параметры, которые указываются в скобках после имени метода.
    * Например, + methodName(param1: Type1, param2: Type2): ReturnType.
16. **Константы:**
    * Константы (final атрибуты) изображаются с подчеркиванием и могут быть обозначены ключевым словом {readOnly}.
17. **Интерфейсы и абстрактные классы:**
    * Интерфейсы и абстрактные классы могут иметь методы по умолчанию, которые реализуются в классах, реализующих интерфейс или наследующих абстрактный класс.

2) Результаты:

2.1. Практической работы «Диаграмма вариантов использования»:

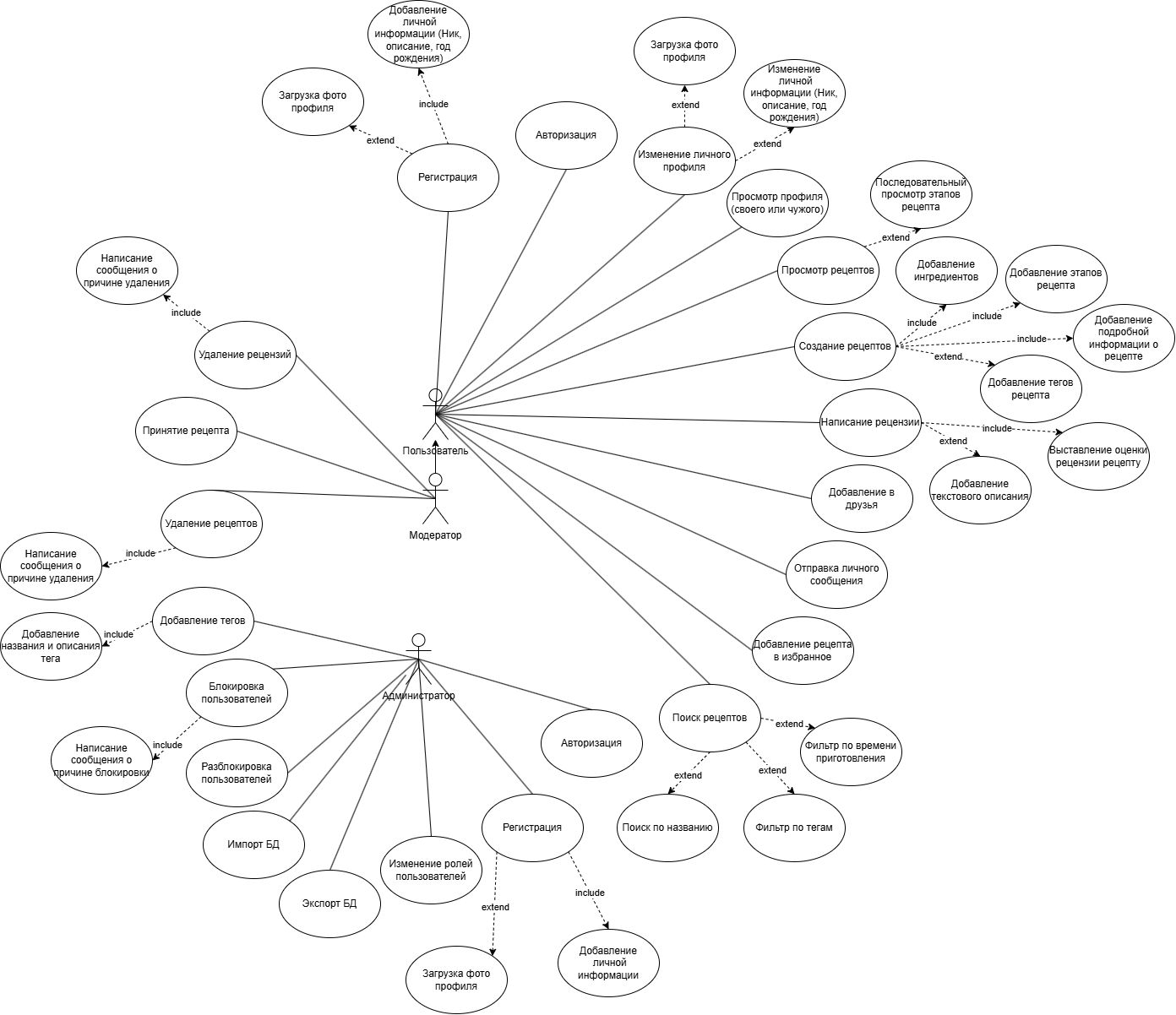


Рисунок 1 - Практическая работа «Диаграмма вариантов использования»

2.2. Практической работы «Практическое применение диаграммы в нотации IDEF1X»:

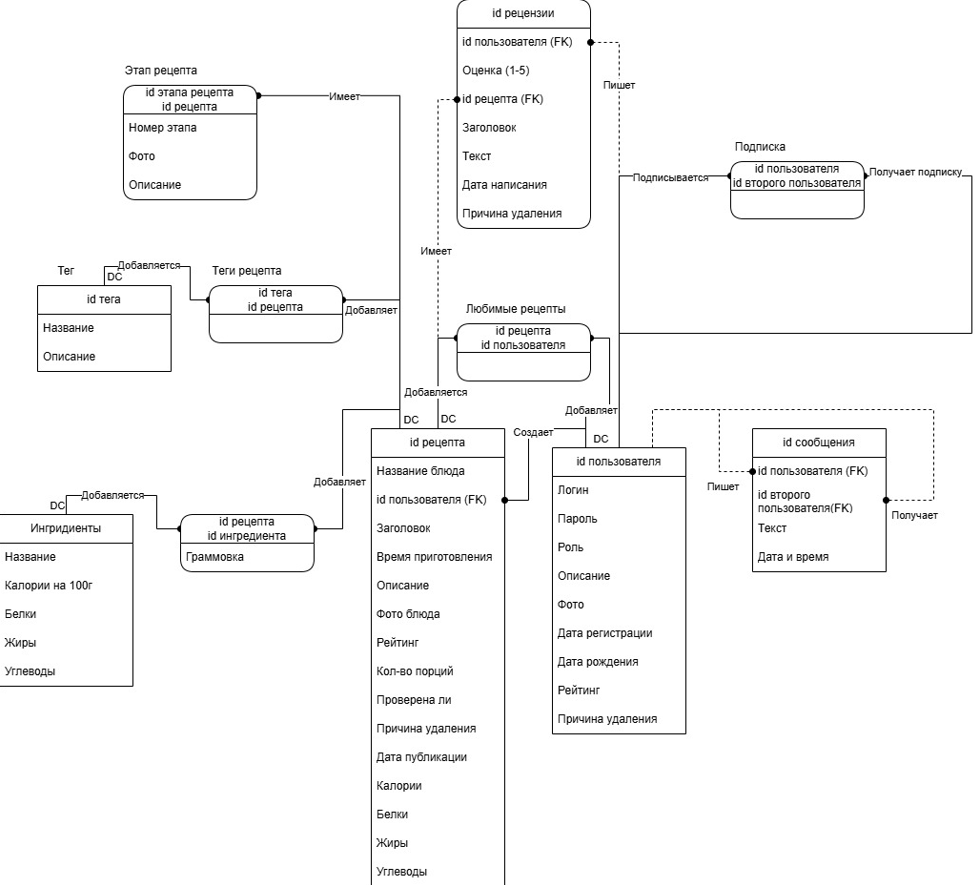


Рисунок 2 - Практическая работа «Практическое применение диаграммы в нотации IDEF1X»

2.4. Создайте макет окон для пользователей ИС (Рецепты блюд).

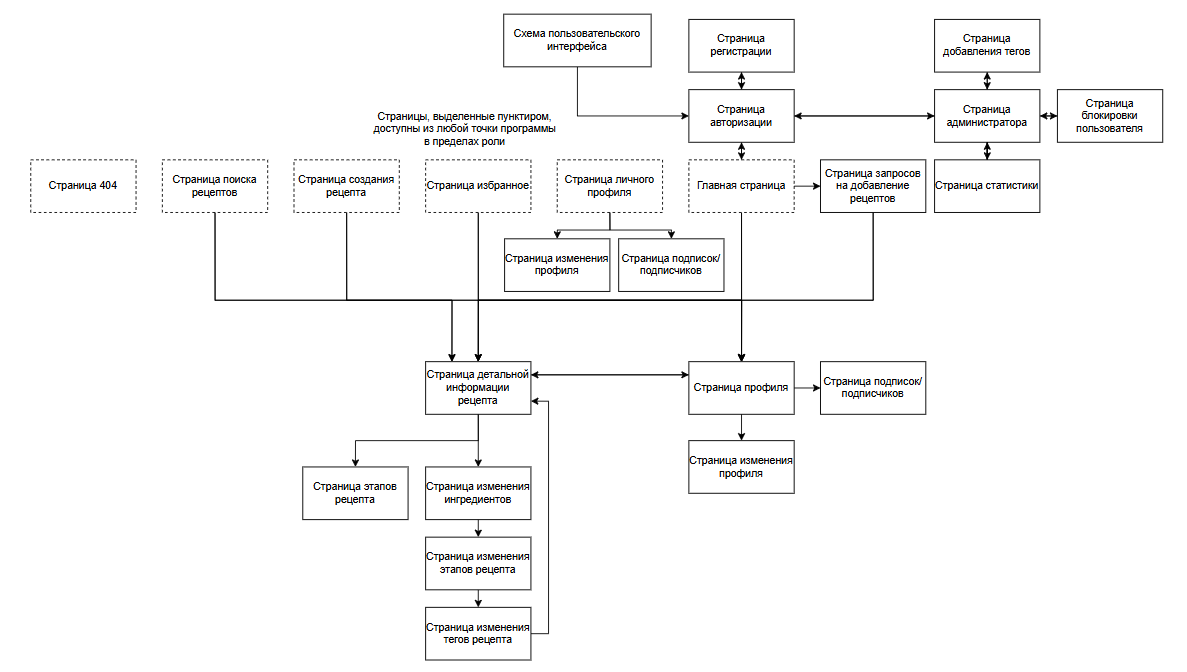


Рисунок 3 - Схема интерфейса

* 1. Построить диаграмму классов (1-ый-8-ой варианты). Следует пояснять выполненный вариант диаграммы классов.

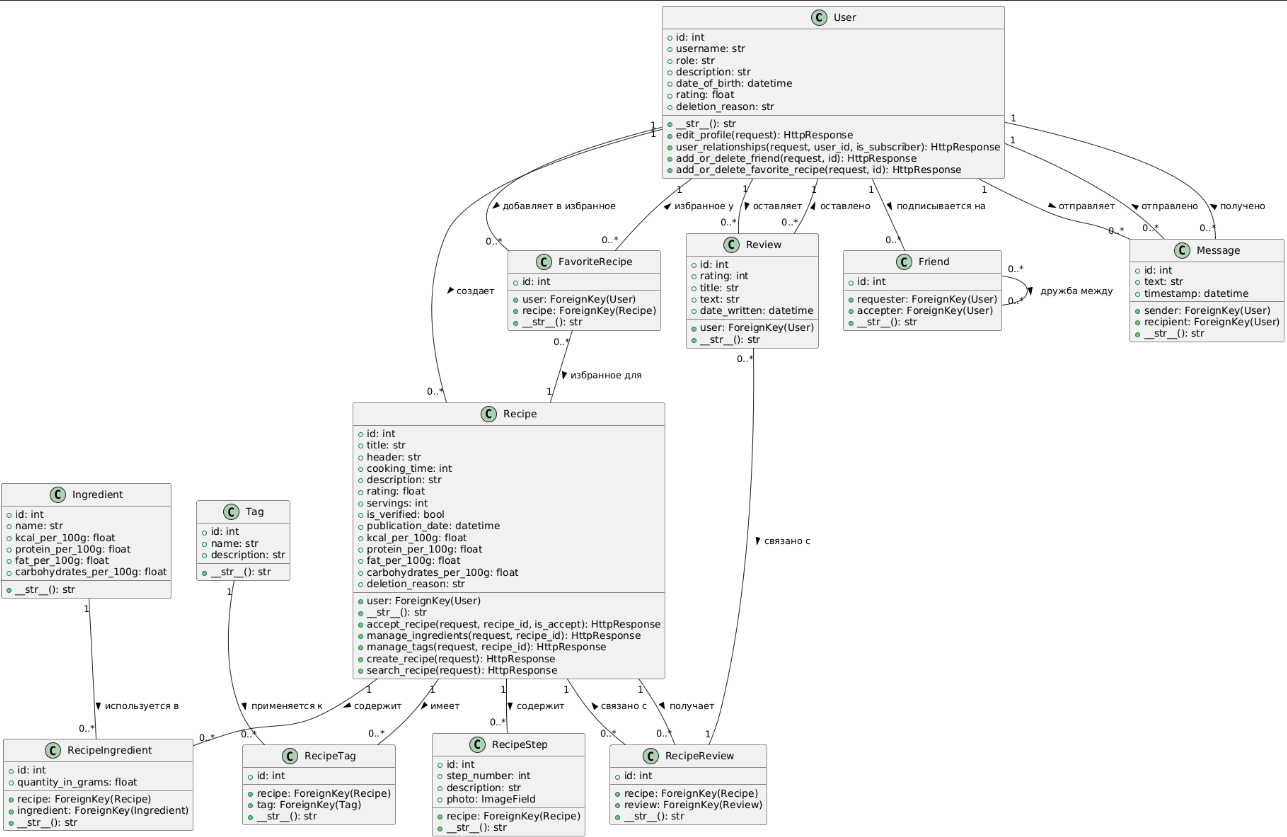


Рисунок 4 - Диаграмма классов

* 1. Произвести проверку правильности построения диаграммы, используя выполненный пункт 1.

|  |  |  |
| --- | --- | --- |
| **№** | **Название правила** | **Статус соответствия правилу** |
| 1 | Класс изображается в виде прямоугольника, разделенного на секции: имя класса, атрибуты, методы. | + |
| 2 | Объекты — это экземпляры классов, обозначаются с подчеркиванием имени. | + |
| 3 | Атрибуты описывают свойства класса. Методы описывают поведение класса. | + |
| 4 | Ассоциация: Связь между классами, обозначается линией со стрелкой. | + |
|  | Наследование: Обозначается стрелкой с закрашенным треугольником. | + |
|  | Реализация интерфейса: Обозначается пунктирной стрелкой с закрашенным треугольником. | + |
|  | Агрегация: Обозначается ромбом на конце линии. | + |
|  | Композиция: Обозначается закрашенным ромбом на конце линии. | + |
| 5 | Указывается количество объектов, участвующих в отношении. | + |
| 6 | Модификаторы доступа отображаются следующим образом: | + |
|  | - + (public) | + |
|  | - - (private) | + |
|  | - # (protected) | + |
| 7 | Статические атрибуты и методы подчеркиваются. | + |
|  | Имена абстрактных классов и методов записываются курсивом. | + |
| 8 | Пакеты используются для организации классов и других элементов в группы. Пакет изображается в виде папки с именем пакета. | + |
| 9 | Интерфейс изображается в виде прямоугольника с именем интерфейса и ключевым словом «interface». Интерфейс может содержать только методы (без атрибутов). | + |
| 10 | Последовательные диаграммы описывают взаимодействие объектов во времени. Объекты изображаются в виде вертикальных линий (жизненных линий). Сообщения между объектами изображаются в виде горизонтальных стрелок. | + |
| 11 | Диаграммы состояний описывают поведение объекта в зависимости от его состояния. Состояния изображаются в виде кружков с именем состояния. Переходы между состояниями изображаются в виде стрелок с условием перехода. | + |
| 12 | Диаграммы компонентов описывают структуру системы в терминах компонентов и их взаимосвязей. Компонент изображается в виде прямоугольника с двумя маленькими прямоугольниками по бокам. Взаимосвязи между компонентами изображаются в виде линий со стрелками. | + |
| 13 | Диаграммы развертывания описывают физическое размещение компонентов системы на узлах. Узлы изображаются в виде кубов. Компоненты размещаются на узлах, и их связи изображаются в виде линий со стрелками. | + |
| 14 | Аннотации используются для добавления комментариев или пояснений к элементам диаграммы. Аннотация изображается в виде прямоугольника с линией, соединяющей его с элементом диаграммы. | - |
| 15 | Методы могут иметь параметры, которые указываются в скобках после имени метода. Например, + methodName(param1: Type1, param2: Type2): ReturnType. | + |
| 16 | Константы (final атрибуты) изображаются с подчеркиванием и могут быть обозначены ключевым словом {readOnly}. | + |
| 17 | Интерфейсы и абстрактные классы могут иметь методы по умолчанию, которые реализуются в классах, реализующих интерфейс или наследующих абстрактный класс. | + |

Вывод: закрепили теоретические знания по принципам создания диаграммы классов. Получили практические навыки по построению диаграммы классов.