**Практическое задание 3**

**Анализ предметной области. Определение требований проекта. Разработка документа «Техническое задание» (разработка и оформление документа, согласование документа с заказчиком и руководителем, корректировка документа). Выбор модели разработки программного обеспечения**

**Цель занятия:** приобрести разработки приложений к техническому заданию: описание предметной области в терминах ООП, выделение классов, создания спецификации классов, диаграммы классов

**Оборудование:** ПК, доступ к Интернет, Umbrello

**Источники:**

1. [Полное руководство по диаграмме классов UML - Кибермедиана (cybermedian.com)](https://www.cybermedian.com/ru/a-comprehensive-guide-to-uml-class-diagram/)
2. Видеоурок. Построение диаграммы классов в Umbrello. [**https://youtu.be/TWVrZCTS5m0**](https://youtu.be/TWVrZCTS5m0)

**ЗАДАНИЕ К ПРАКТИЧЕСКОМУ ЗАНЯТИЮ**

Вы являетесь разработчиком программного обеспечения в фирме ИП «Программист», директором фирмы является ваш руководитель практики. В Вашу фирму обратился заказчик, для которого необходимо разработать бот для работы с информацией указанной в таблице предметной области (см. вариант задания). В беседе с пользователем было установлено, ряд требований заказчика согласно которым должна осуществляться логика работы бота

1. При запуске бота появляется окно для задания имени пользователя, потом основное окно.
2. Ответ ботом на несколько реплик заданного шаблона («Привет, Бот!» и т.п.)
3. Ответ на простые команды (вопросы). Например: «Который час?», вопросы о статистике по обмену сообщениями и т.п.
4. Ответ на команды с параметрами: Например: «умножь 12 на 157»
5. Бот должен хранить историю сообщений, включая время отправки и автора.
6. Записывать историю в файл при завершении программы. Загружать из файла при запуске программы
7. Программа должна иметь диалоговый интерфейс с пользователем
8. Программа должна иметь модульную архитектуру и соответствовать принципам построения ООП.

**Варианты:**

|  |  |
| --- | --- |
| № Варианта | Предметная область |
| 1 | Страховая Медицинская компания |
| 2 | Агентство недвижимости |
| 3 | Кадровое агентство |
| 4 | Компания по разработке программных продуктов |
| 5 | Туроператор |
| 6 | Регистратура поликлиники |
| 7 | Станция техобслуживания автомобилей |
| 8 | Гостиница |
| 9 | Агентство по трудоустройству |
| 10 | Отдел кадров |
| 11 | Агентство грузоперевозок |
| 12 | Фирма-производитель мебели |
| 13 | Кинотеатр |
| 14 | Магазин компьютерной техники |
| 15 | Библиотека |

Прежде чем приступить к разработке бота **НЕОБХОДИМО РАЗРАБОТАТЬ ТЕХНИЧЕСКОЕ ЗАДАНИЕ и утвердить его руководителем ИП «Программист» и согласовать с заказчиком Ивановым И.И.** Основанием разработки технического задания является договор №1 от 29.11.2023 г «О разработке бота по требованиям заказчика». Срок реализации заказа не более 5 календарных дней со дня утверждения технического задания на разработку приложения.

Для понимания особенности построения и работы приложения бота заказчик попросил сопроводить техническое задание приложениями:

Приложение А. Описание предметной области в терминах ООП

Приложение Б. Спецификация приложения

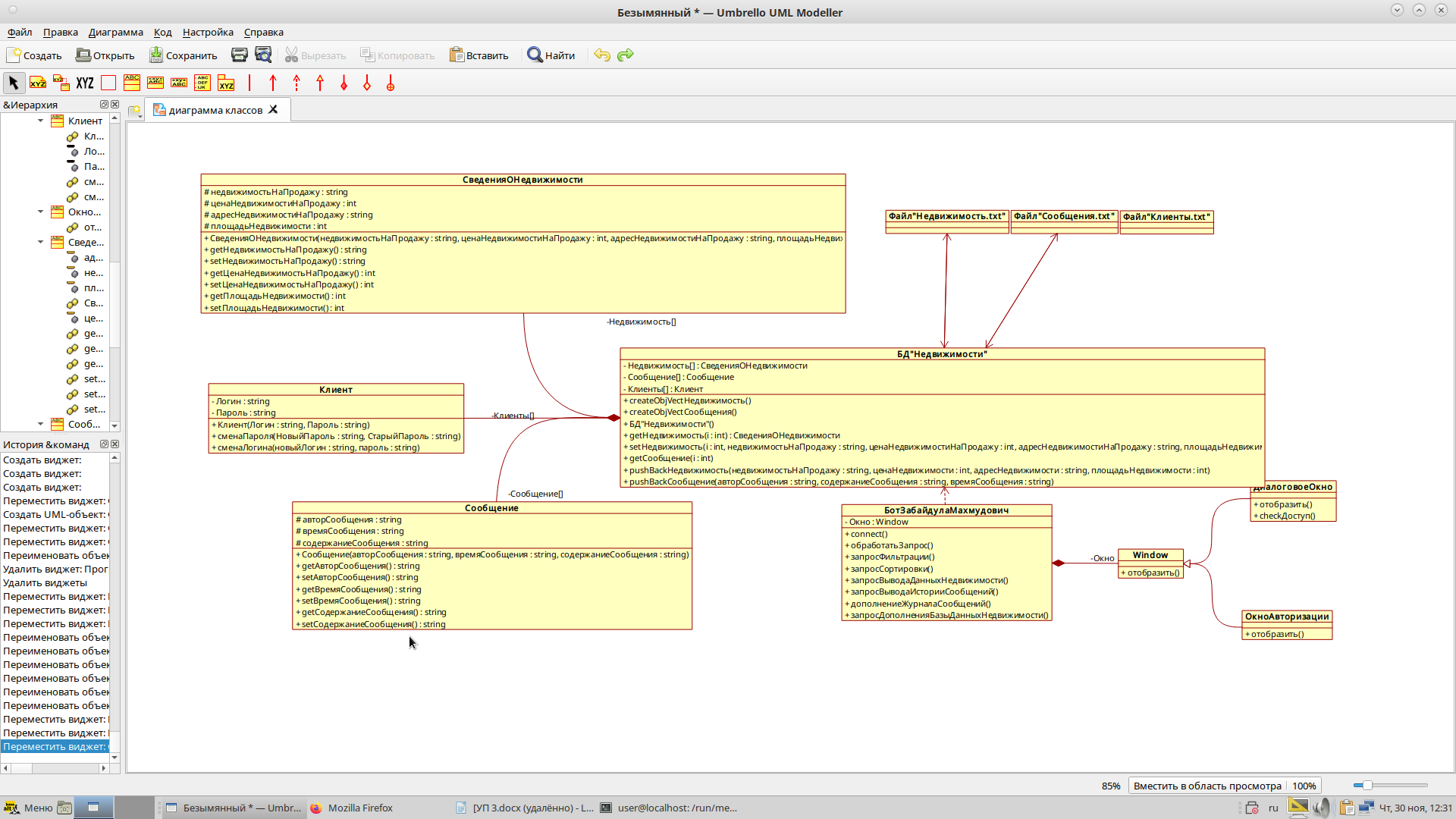
Приложение В. Диаграмма классов

**Для решения задачи выполните следующие задания:**

**Задание 1.** Работая с источником 1 ознакомьтесь с правилами представления классов на языке UML. Заполните таблицу данными

|  |  |
| --- | --- |
| Графическое обозначение класса: |  |
| Графическое обозначение отношений:   1. Наследование – это Механизм создания одного класса с помощью другого с использованием всех свойств и методов супер-класса, добавляя новые методы при необходимости |  |
| 2. Ассоциация – это непрерывная линия между двумя классами, направленная от исходного класса к целевому классу. Имя свойства (вместе с кратностью) располагается на целевом конце ассоциации. |  |
| 3. Агрегация – это отношение которое показывает,между двумя классами,что один из них включает в себя другой класс в качестве составной части. При этом класс-часть может и существовать обособленно от класса-целого |  |
| 4. Состав – это отношение которое является частным случаем отношения агрегации. Однако у него есть одно отличие – классы-части, которые он соединяет с классом-целым, не могут существовать обособленно. |  |
| 5. Зависимость – это отношение возникает между двумя классами, если изменения в определении одного могут вызвать изменения в другом (но не наоборот). |  |
| Графическое обозначение модификаторов доступа   1. Публичный – это открытый доступ при котором атрибуты видны всем пользователям |  |
| 1. Частный – это закрытый доступ при котором атрибуты невидны всем пользователям |  |
| 1. Защищенный – это доступ открытый доступ при котором атрибуты видны только одному пользователю и они имеют привязку к логину и паролю |  |

**Задание 2.** Работая с техническим заданием составьте в Umbrello диаграмму классов.



**Задание 3.** Работая с диаграммой классов составьте спецификацию программного приложения. Для этого используйте ниже представленные шаблоны

Спецификация программного приложения: свойства классов

|  |  |  |  |
| --- | --- | --- | --- |
| КЛАСС | Свойства | | |
| Имя | Тип данных | Модификатор доступа |
| Клиент | программа | БотЗабайдулаМахмудович | private |
| токен | string | private |
| СведенияОНедвижемости | недвижимостьНаПродажу | string | protected |
| ценаНедвижимостиНа продажу | int | protected |
| адрессНедвижимостиНапродажу | string | protected |
| площадьНедвижимости | int | protected |
| Сообщения | авторСообщения | string | protected |
| времяСообщения | string | protected |
| содержаниеСообщения | string | protected |
| БД«Недвижимости» | Недвижимость | СведенияОНедвижимости | private |
| Сообщение | Сообщения | private |
| Файл «Недвижимость.txt» |  |  |  |
| Файл «Сообщения.txt» |  |  |  |
| Файл «Клиенты.txt» |  |  |  |
| БотЗабайдулаМахмудов | Окно | Window | private |

Спецификация программного приложения: методы классов

|  |  |  |
| --- | --- | --- |
| КЛАСС | Методы | |
| Имя | Тип данных, описание, доступ |
| Сообщения | Сообщение | Тип данных: авторСообщения:string,времяСообщения:string, содержаниеСообщения:string  Описание:public |
| getАвторСообщения() | Тип данных:string  Описание:public |
| setАвторСообщения() | Тип данных:string  Описание:public |
| getВремяСообщения() | Тип данных:string  Описание:public |
| setВремяСообщения() | Тип данных:string  Описание:public |
| getСодержаниеСообщения() | Тип данных:string  Описание:public |
| setСодержаниеСообщения() | Тип данных:string  Описание:public |
| СведенияОНедвижмости | СведенияОНедвижмости | Тип данных:string  Описание: |
| getНедвижмостьНаПродажу() | Тип данных:недвижмостьНаПродажу:string,ценаНедвижмостиНаПродажу:int,адрессНедвижимости:string,площадьНедвижмости:int  Описание:public |
| setНедвижмостьНаПродажу() | Тип данных:string  Описание: |
| getЦенаНедвижмостиНаПродажу() | Тип данных:int  Описание:public |
| setЦенаНедвижмостиНаПродажу() | Тип данных:int  Описание:public |
| getНедвижмости() | Тип данных:int  Описание:public |
| setНедвижмости | Тип данных:int  Описание:public |
| БД «Недвижимости» | createObjVectНедвижимость() | Тип данных:  Описание:public |
| createObjVectСообщения() | Тип данных:  Описание:public |
| БД «Недвижимости»() | Тип данных:  Описание:public |
| getНедвижимость | Тип данных:  (i:int):СведенияОНедвижмости  Описание:public |
| setНедвижимость | Тип данных:  (i:int,недвижмостьНаПродажу:string,ценаНедвижмостиНаПродажу:int,адрессНедвижимости:string,площадьНедвижмости:int)  Описание:public |
| getСообщение | Тип данных:int  Описание:public |
| pushBackНедвижимость | Тип данных  (недвижмостьНаПродажу:string,ценаНедвижмостиНаПродажу:int,адрессНедвижимости:string,площадьНедвижмости:int)  Описание:public |
| pushBackСообщение | Тип данных:  (авторСообщения:string,времяСообщения:string, содержаниеСообщения:string)  Описание:public |
| БотЗабайдулаМахмудов | connect() | Тип данных:  Описание:public |
| обработатьЗапрос | Тип данных:  Описание:public |
| запросФильтрации | Тип данных:  Описание:public |
| запросСортировки | Тип данных:  Описание:public |
| запросВыводаДанныхНедвижимости | Тип данных:  Описание:public |
| запросВыводаИсторииСообщений | Тип данных:  Описание:public |
| дополнениеЖурналаСообщений | Тип данных:  Описание:public |
|  | запросДополненияБазыДанныхНедвижимости() | Тип данных:  Описание:public |
| Window | отбразить() | Тип данных:  Описание:public |
| ДиалоговоеОкно | отбразить() | Описание:public |
|  | Тип данных:  Описание:public |
| ОкноАвторизации | отбразить() | Тип данных:  Описание:public |

**Задание 4.** Работая с диаграммой классов составьте описание предметной области в терминах ООП. Для этого используйте ниже представленный план:

1. **Описание методов**:

А) Класс клиент умеет регистрироваться и менять пароль и логин и взаимодействовать с ботом

Б)Класс СведенияОНедвижимости может показывать и создавать записи по темам связанными с недвижимостью (НедвижимостьНаПродажу, АдрессНедвижимостиНаПродажу,ЦенаНедвижемостиНаПродажу,ПлощадьНедвижемости)

В)Класс Сообщения может показывать и создавать сообщения и сведения о них (Автор,время)

Г)Класс БД «Недвижимость» умеет создавать массивы объектов сведенияОНедвижимости, Клиент, Сообщение

Д)Класс БотЗабайдулаМахмудов умеет обрабатывать запросы, филтрировать и сортировать их, выводить данные о недвижимости и историю сообщений,дополнять журнала сообщений,выводить дополнение базы данных недвижимости

Е)Класс Window умеет отображать диалоговое окно и окно авторизации

1. **Описание свойств**:

А) Класс БД «Недвижимость» хранит в себя сведение о недвижимости,логин и пароль клиенты и всю информацию о сообщениях.

1. **Зависимость**:

Класс БД «Недвижимость» могут затронуть изменения в классе БотЗабайдулаМахмудов.

Класс Файл«Недвижимость.txt» могут затронуть изменения в классе БД «Недвижимость».

Класс Файл«Сообщения.txt» могут затронуть изменения в классе БД «Недвижимость».

Класс Файл«Клиент.txt» могут затронуть изменения в классе БД «Недвижимость».

1. **Агрегация**:

Объект Window знает об объекте ДиалоговоеОкно и состоит из него и управляет его жизненным циклом. Класс Window зависит от ДиалоговоеОкно.

Объект Window знает об объекте ОкноРегистрации и состоит из него и управляет его жизненным циклом. Класс Window зависит от ОкноРегистрации.

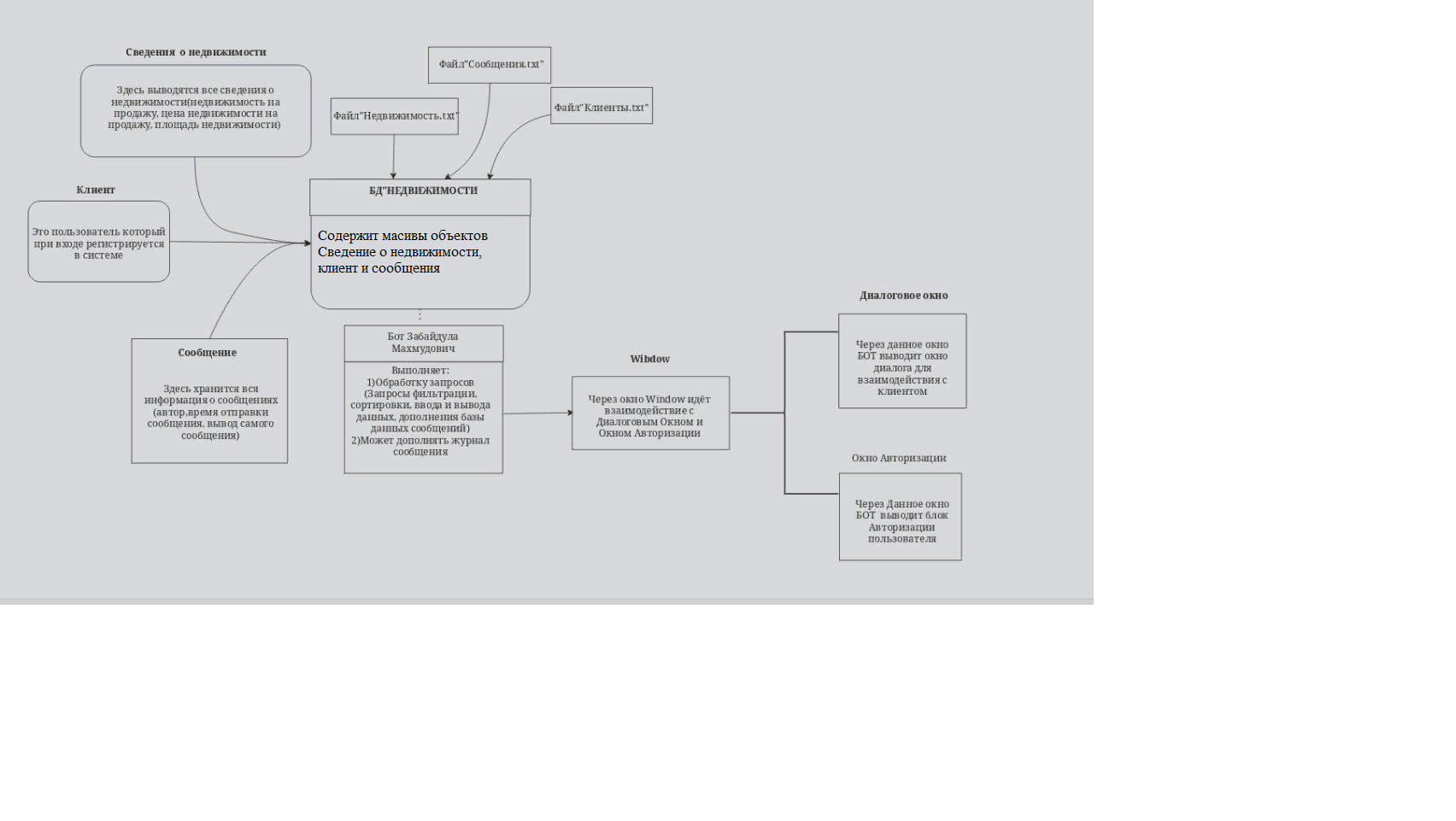
1. **Композиция**:

Объект БД «Недвижимость» знает об объекте СведенияОНедвижимости и состоит из него и управляет его жизненным циклом. Класс БД «Недвижимость» зависит от СведенияОНедвижимости.

Объект БД «Недвижимость» знает об объекте Сообщения и состоит из него и управляет его жизненным циклом. Класс БД «Недвижимость» зависит от Сообщения.

Объект БД «Недвижимость» знает об объекте Клиент и состоит из него и управляет его жизненным циклом. Класс БД «Недвижимость» зависит от Клиент.

Объект БотЗабайдулаМахмудов знает об объекте Window и состоит из него и управляет его жизненным циклом. Класс БотЗабайдулаМахмудов зависит от Window.

**Задание 5.** Работая с редактором DRAW.IO , диаграммой классов и техническим заданием создайте схему работы бота. Пример схемы.

