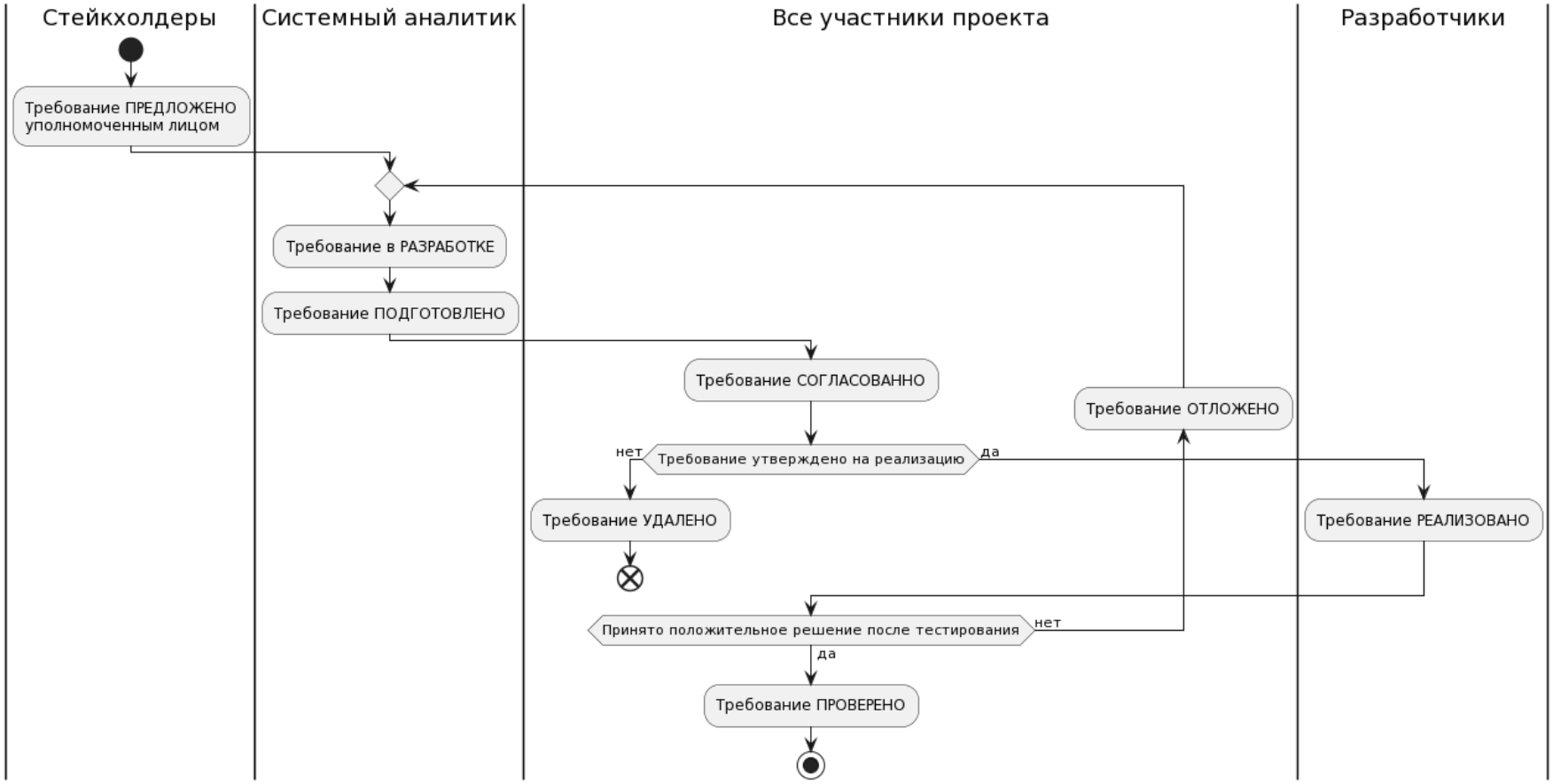
# Диаграммы деятельности (Activity Diagrams)

**1.Согласование и управление изменениями требований**
**Контекст:**
Составить описание бизнес-процесса в любой нотации. Путь согласования требования: от момента его инициации до завершения работы с требованием. Бизнес-процесс должен описать процесс перехода из одного состояния требования в другое. Подробное описание не нужно.
**Задание:**
Заполнить порядок согласования требования или изменения требования.

**Выполнение:** в [http://www.plantuml.com/](http://www.plantuml.com/)

**Алгоритм согласования и управления изменениями требований**




```
@startuml
|Стейкхолдеры|
start
:Требование ПРЕДЛОЖЕНО
уполномоченным лицом;
|Системный аналитик|
repeat
  :Требование в РАЗРАБОТКЕ;
  :Требование ПОДГОТОВЛЕНО;
|Все участники проекта|
   :Требование СОГЛАСОВАННО;
    if (Требование утверждено на реализацию) then (нет)
      :Требование УДАЛЕНО;
      END
    else (да)
      |Разработчики|
        :Требование РЕАЛИЗОВАНО;
    endif
|Все участники проекта|
backward :Требование ОТЛОЖЕНО;
repeat while (Принято положительное решение после тестирования)
is (нет) not (да)
  :Требование ПРОВЕРЕНО;
Stop    @enduml
```

**2.Алгоритм работы CRM системы**

**Контекст:**

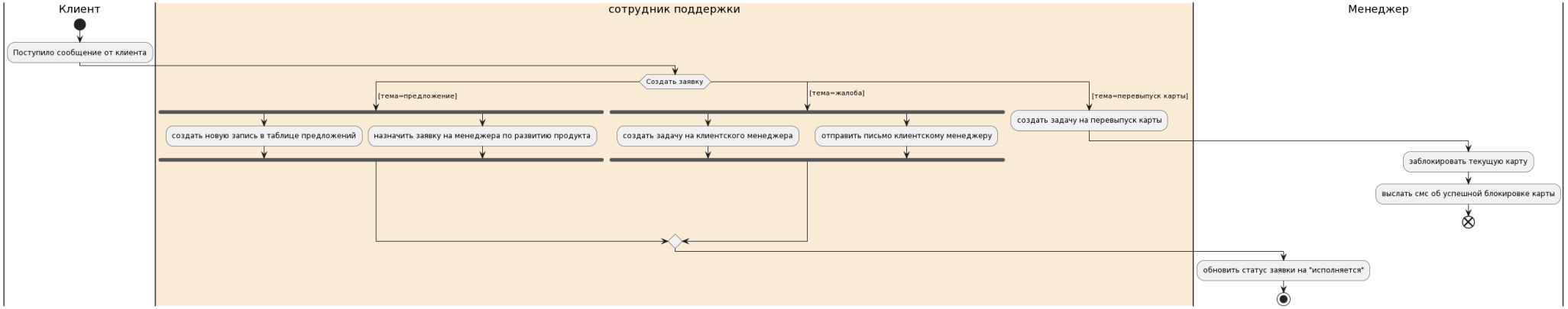
Необходимо описать алгоритм работы CRM системы, в которой сотрудник поддержки заводит заявку, поступившую к нему от клиента сервиса топливных карт. Заявки эти он как-то маркирует по темам, что должно заставлять CRM систему включить соответствующий способ реагирования.

**Задание:**

Составить диаграмму деятельности. Указать все необходимые элементы, подумать, где действия можно запараллелить, а где – они строго последовательны.

**Выполнение:** в [http://www.plantuml.com/](http://www.plantuml.com/)

**Алгоритм работы в CRM системе**

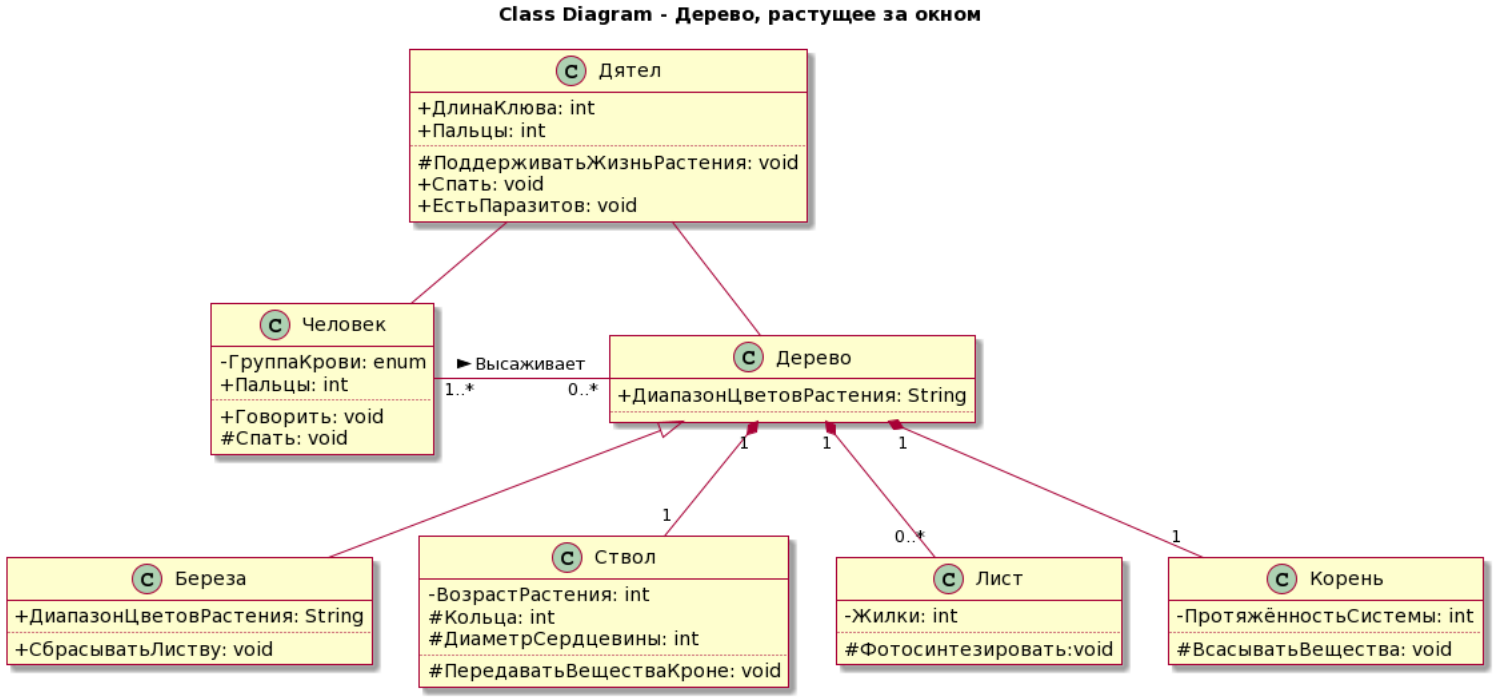



```
@startuml
|Клиент|
start
:Поступило сообщение от клиента;
|#AntiqueWhite|сотрудник поддержки|
switch (Создать заявку)
case ( [тема=предложение] )
 fork
  :создать новую запись в таблице предложений;
 fork again
  :назначить заявку на менеджера по развитию продукта;
 end fork
case ( [тема=жалоба] )
 fork
  :создать задачу на клиентского менеджера;
 fork again
  :отправить письмо клиентскому менеджеру;
end fork
case ( [тема=перевыпуск карты] )
 :создать задачу на перевыпуск карты;
|Менеджер|
 :заблокировать текущую карту;
 :выслать смс об успешной блокировке карты;
 end
endswitch
|Менеджер|
:обновить статус заявки на "исполняется";
stop
@enduml
```

# Диаграмма классов (Class Diagram)

**Контекст:**
Представьте, что систему, которую вы опишете с помощью диаграммы – простое дерево, растущее за окном. UML помогает разглядеть лес за стволами деревьев.
**Задание:**
составить диаграмму, правильно расположить и связать между собой все элементы. Используйте только корректные ассоциации и укажите множественность там, где нужно. Не забудьте проставить видимость и тип данных для атрибутов.

**Выполнение:** в https://www.planttext.com/




```
@startuml
skin rose
title  Class Diagram - Дерево, растущее за
окном
skinparam classAttributeIconSize 0
class Человек {
-ГруппаКрови: enum
+Пальцы: int
..
+Говорить: void
#Спать: void
}
class Дерево {
 +ДиапазонЦветовРастения: String
 ..
}
class Дятел {
+ДлинаКлюва: int
+Пальцы: int
..
#ПоддерживатьЖизньРастения: void
+Спать: void
+ЕстьПаразитов: void
}
class Береза {
+ДиапазонЦветовРастения: String
..
+СбрасыватьЛисту: void
}
Class Ствол {
-ВозрастРастения: int
#Кольца: int
#ДиаметрСердцевины: int
..
#ПередаватьВеществаКроне: void
}
Class Лист {
-Жилки: int
..
#Фотосинтезировать:void
}
class Корень {
-ПротяжённостьСистемы: int
..
#ВсасыватьВещества: void
}
Человек "1..*" - "0..*" Дерево :
Высаживает >
Дятел -- Человек
Дятел -- Дерево
Дерево <|-- Береза
Дерево "1" *-- "1" Ствол
Дерево "1" *-- "0..*" Лист
Дерево "1" *-- "1" Корень
@enduml
```

**Диаграмма вариантов использования (UseCase Diagram)**

**Контекст:**
Представьте, что вы прорабатываете приложение, которое помогает связаться съёмщикам и арендодателям для того, чтобы снять\сдать квартиру. Необходимо обозначить действующие лица, очертить границы приложения, перечислить варианты использования и правильно связать их между собой.
**Задание:** подготовьте диаграмму системных вариантов использования для приложения по съёму жилья.

## Выполнение в draw.io:

# Описание вариантов использования

**Задание:**
Необходимо подготовить текстовое описание варианта использования “Примерить товар в магазине”
**Выполнение:**

**UC-BC-001** Примерить товар в магазине
**Краткое описание:** Потенциальный покупатель бронирует товар для примерки на сайте магазина одежды, выбирая удобный магазин и размер товара
**Действующие лица:** покупатель, сотрудник магазина
**Триггер:** покупатель зашел в каталог товаров
**Предусловия:** Товар присутствует в каталоге товаров. Сотрудник авторизован в системе учета товаров магазина.

**Основной поток:**
1. Покупатель открывает информацию о товаре
2. Система предлагает покупателю выбрать размер товара
3. Покупатель выбирает размер товара
4. **Если** товар доступен в запрошенном размере в каком-либо из магазинов, то управление переходит на следующий шаг
5. Покупатель выбирает опцию «Где примерить?»
6. Система предлагает список магазинов, в которых доступен данный товар в указанном размере
7. Покупатель выбирает магазин для явки на примерку товара
8. Покупатель подтверждает бронирование товара в выбранном магазине для примерки
9. Сотрудник магазина получает запрос на подготовку товара к бронированию и примерке
10. Сотрудник готовит товар к примерке и выставляет статус конкретной единицы товара «Забронирован»
11. Система сохраняет статус указанного товара в состоянии «забронирован» на 1 сутки
12. **Если** покупатель является в течение суток для примерки, то управление переходит на следующий шаг
13. Покупатель примеряет товар
14. Вариант использования завершает свою работу

**Альтернативный поток:**
4а. Товар в указанном размере не доступен ни в одном магазине
1. Система отображает уведомление о недоступности товара
2. Управление переходит на шаг 2

**Поток исключения:**
12а. Покупатель не является на примерку товара в течение суток
1. Система меняет статус товара с «Забронировано» на статус по умолчанию
2. Вариант использования завершает свою работу

**Постусловие:** В случае успешного выполнения основного потока, покупатель примеряет товар

**Результат:** Если ВИ выполнен успешно, то товар был примерен покупателем