**Диагра́мма де́ятельности** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *activity diagram*) — [UML](https://ru.wikipedia.org/wiki/UML)-диаграмма, на которой показаны действия, состояния которых описаны на [диаграммах состояний](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B9_(UML)). Под деятельностью ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *activity*) понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *action*, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

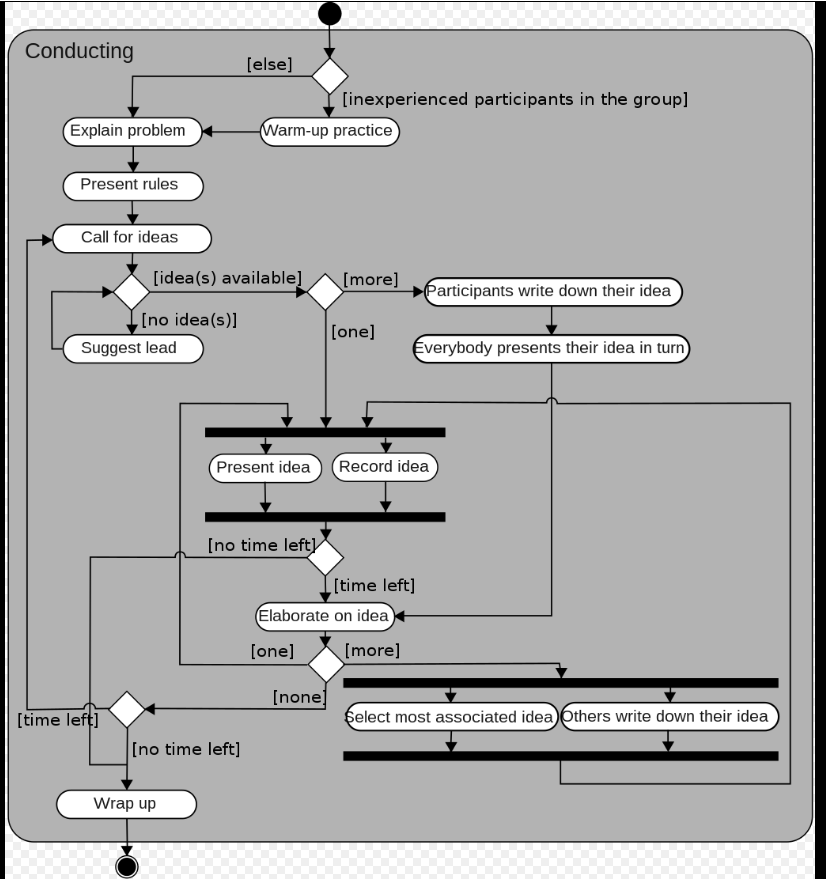
Диаграммы деятельности используются при моделировании [бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81), технологических процессов, последовательных и параллельных вычислений.

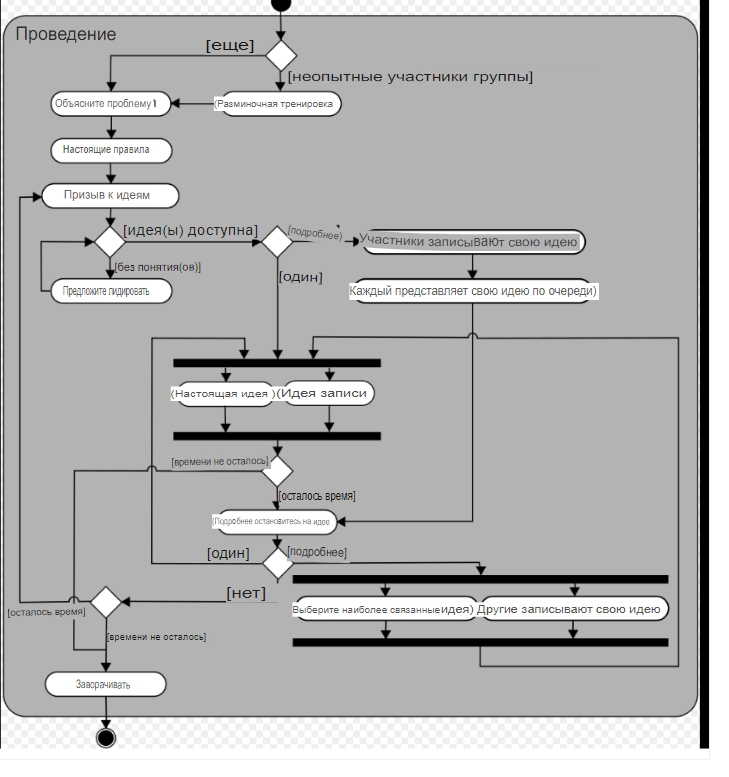
Диаграммы деятельности состоят из ограниченного количества фигур, соединённых стрелками. Основные фигуры (узлы)[[1]](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%B4%D0%B5%D1%8F%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8#cite_note-1):

1. Прямоугольники с закруглениями — действия (операция). **Узел управления (control node)** — это абстрактный узел действия, которое координирует потоки действий.
2. Ромбы — решения. **Узел решения** предназначен для определения правила ветвления и различных вариантов дальнейшего развития сценария. В точку ветвления входит ровно один переход, а выходит — два или более.
3. Широкие полосы — начало (разветвление) и окончание (схождение) ветвления действий. **Узел объединения** имеет два и более входящих узла и один исходящий.
4. Чёрный круг — начало процесса (начальный узел). **Начальный узел деятельности (или начальное состояние деятельности) (activity initial node)** является узлом управления, в котором начинается поток (или потоки) при вызове данной деятельности извне.
5. Чёрный круг с обводкой — окончание процесса (финальный узел). **Конечный узел деятельности (или конечное состояние деятельности) (activity final node)** является узлом управления, который останавливает все потоки данной диаграммы деятельности. На диаграмме может быть более одного конечного узла.

Стрелки идут от начала к концу процесса и показывают потоки управления или потоки объектов (данных).







5.1 Тест-требования

К разработанному приложению будут представлены некоторые требования надёжности. Для выполнения этих требований проводится процедура тестирования приложения.

Одним из этапов тестирования является подготовка тест-требований.

Так как в системе предусмотрены несколько конечных пользователей, то на операции представленные выше должны так же накладываться ограничения, которые должны быть проверены.

1. Проверка создания нового пользователя и авторизации в системе.
   1. Проверить что при корректном запросе на создания нового пользователя, создаётся новый пользователь и система оповещает об этом.
   2. Проверить что при некорректном создании нового пользователя система не создаёт пользователя и сообщает об ошибке.
   3. Проверить что при создании пользователя с уже имеющимся логином, пользователь создан не будет, и система сообщит об этом.
   4. Проверить что при корректном вводе данных будет осуществлён вход в приложение.
   5. Проверить что при некорректном вводе данных вход в приложение не будет осуществлён, и система выдаст сообщение об ошибке.
2. Проверка записи на приём и регистрации новых пациентов
   1. Проверить что при корректном запросе на создание нового пациента, создаётся новый пациент и система сообщает об этом.
   2. Проверить что при некорректном создании нового пациента система выдаёт сообщение об ошибке.
   3. Проверить что при вводе уже существующей даты и времени, при создании записи на приём к определённому врачу система выдаст сообщение о том, что данное время занято.
   4. Проверить, что при корректном запросе создания записи на приём, запись будет создана и система сообщит об этом.
   5. Проверить, что при некорректном создании записи на приём, система выдаст сообщение об ошибке
   6. Проверить, что при входе на панель с историей болезни, система отображает всю историю болезни.
   7. Проверить, что при поиске по ФИО или паспорту система отображает именно этого пациента.
3. Проверка осуществления приёма
   1. Проверить, что при корректном запросе на добавления записи о новой болезни пациента, установки диагноза и лечения, система оповещает о том, что запись создалась.
   2. Проверить, что при добавлении новой записи о болезни, автоматически заполняется истории болезни пациента.
   3. Проверить, что при выборе пациента отображается именно его история болезни. возможность сортировки истории болезни по ФИО или паспорту пациента.
4. Проверка смены графика работы
   1. Проверить, что при изменении графика работы, эти изменения буду учтены и системы выдаст сообщения об их изменении и передаст это сообщение регистратору.
5. Проверка работы шаблонов
   1. Проверить что при выборе шаблона данные о выбранном пациенте заполняются автоматически.

При выполнение данных требований стоит учесть, что в данной системе имеются несколько различных пользователей с определёнными права и ограничениями. Следовательно, проверку следует выполнять поэтапно и по пунктам для каждого пользователя. Тест требования для администратора составлены в пункте 1. Требования для регистратора составлены в пункте 2. Требования для врача составлены в остальных пунктах.

5.2 Тест-план

Следующим этапом для проведения тестирования становится разработка тест-плана.

Тестовый пример 1.

Номер тест требования 1.1.

Описание теста: в данном тесте проверяется корректность создания нового пользователя

Входные данные: Логин, состоящий из букв и чисел от 1 до 9, и любой пароль не используя пробелов

Ожидаемые выходные данные: оповещение системы о создании пользователя, запись о пользователе в БД.

Сценарий:

* Войти в систему как администратор.
* Перейти на панель с добавлением пользователя
* Ввести логин: Administrator
* Ввести пароль: 89Admin41\*

Тестовый пример 2.

Номер тест требования 1.2.

Описание теста: в данном тесте проверяется корректность создания нового пользователя

Входные данные: Логин или пароль имеющие пробелы, оставить поля не заполненными

Ожидаемые выходные данные: оповещение системы об ошибке, запись о пользователе в БД не появляется.

Сценарий:

* + Войти в систему как администратор.
  + Перейти на панель с добавлением пользователя
  + Оставить одно поле не заполненным или использовать логин/пароль имеющий пробел
  + Логин: Admin gr2
  + Пароль: 42 6798AD

Тестовый пример 3.

Номер тест требования 1.3.

Описание теста: в данном тесте проверяется корректность создания нового пользователя

Входные данные: дублируемый логин

Ожидаемые выходные данные: оповещение системы об ошибке пользователя, запись о пользователе в БД не появляется.

Сценарий:

* + Войти в систему как администратор.
  + Перейти на панель с добавлением пользователя
  + Добавить новый логин: User45
  + Сохранить данный логин
  + Заново добавить этот же логин

Тестовый пример 4.

Номер тест требования 1.4.

Описание теста: в данном тесте проверяется возможность входа в приложение

Входные данные: существующие данные о каком-либо пользователе

Ожидаемые выходные данные: вход в приложение.

Сценарий:

* + На панели регистрации ввести существующий логин и пароль

Тестовый пример 5.

Номер тест требования 1.5.

Описание теста: в данном тесте проверяется возможность входа в приложение

Входные данные: несуществующие заполненные данные

Ожидаемые выходные данные: оповещение системы об ошибке пользователя, отказ во входе в приложение.

Сценарий:

* + На панели регистрации ввести несуществующий логин и пароль

Тестовый пример 6.

Номер тест требования 2.1.

Описание теста: в данном тесте проверяется корректность создания нового пациента

Ожидаемые выходные данные: оповещение системы о создании пациента, запись о пациенте в БД.

Сценарий:

* + Войти в систему как регистратор.
  + Перейти на панель с добавлением пациента
  + Заполнить следующие данные:
  + Фамилия: Петров
  + Имя: Павел
  + Отчество: Анатольевич
  + Телефон: +79938567243
  + Дата рождения: 10.12.1985
  + Паспорт: 1945935349
  + Место работы: ООО Техавто
  + Профессия: Инженер
  + Город: Владимир
  + Район: Октябрьский
  + Область: оставить пустой
  + Улица: Отечественная
  + Дом: 46
  + Квартира: 32
  + Тестовый пример 7.
  + Номер тест требования 2.2.

Описание теста: в данном тесте проверяется корректность создания нового пациента

Входные данные: не полностью заполненные данные о новом пациенте

Ожидаемые выходные данные: оповещение системы об ошибке, запись о пациенте в БД не появляется.

Сценарий:

* Войти в систему как регистратор.
* Перейти на панель с добавлением пациента
* Оставить поля для ввода данных пустыми

Тестовый пример 8.

Номер тест требования 2.3.

Описание теста: в данном тесте проверяется корректность создания записи на приём

Входные данные: продублировать время и день к одному и тому же врачу

Ожидаемые выходные данные: оповещение системы об ошибке

Сценарий:

* Войти в систему как регистратор.
* Перейти на панель с добавлением записи на приём
* Ввести уже существующую запись, полностью дублируя её данные

Тестовый пример 9.

Номер тест требования 2.4.

Описание теста: в данном тесте проверяется корректность создания записи на приём

Входные данные: свободные в графике работы врача дата и время

Ожидаемые выходные данные: оповещение системы о том, что запись совершена

Сценарий:

* Войти в систему как регистратор.
* Перейти на панель с добавлением записи на приём
* Ввести дату: 10.11.2022
* Ввести время: 12:40:00

Тестовый пример 9.

Номер тест требования 2.5.

Описание теста: в данном тесте проверяется корректность создания записи на приём

Входные данные: дата и время

Ожидаемые выходные данные: оповещение системы о ошибке

Сценарий:

* Войти в систему как регистратор.
* Перейти на панель с добавлением записи на приём
* Ввести дату: 11.03.2022
* Поле для заполнения времени оставить пустым

Тестовый пример 10.

Номер тест требования 2.6.

Описание теста: Возможность просмотра графика или таблицы посещаемости

Ожидаемые выходные данные: Показ графика или таблицы с посещаемостью пациентов

Сценарий:

* Войти в систему как регистратор.
* Перейти на панель с записями о посещаемости
* Убедиться, что данные показываются корректно

Тестовый пример 11.

Номер тест требования 2.7.

Описание теста: Возможность поиска и сортировки данных

Входные данные: ФИО или паспорт пациента

Ожидаемые выходные данные: Показ графика или таблицы с посещаемостью конкретного пациента

Сценарий:

* Войти в систему как регистратор.
* Перейти на панель с записями о посещаемости
* Ввести в строке поиска существующие в БД ФИО или паспорт

Тестовый пример 12.

Номер тест требования 3.1.

Описание теста: Проверка осуществления приёма

Входные данные: Данные о болезни пациента и так далее.

Ожидаемые выходные данные: Добавления записи о новой болезни пациента

Сценарий:

* Войти в систему как врач.
* Перейти на панель с приёмом
* Выбрать необходимого пациента
* Внести новые данные о болезни:
* Диагноз: аритмия
* Лечение: Кардиолептин

Куприянов

3 Функции системы

3.1 Функция 1

Формирование массива целых чисел по запросу пользователя после ввода этого массива в определённое поле.

|  |  |
| --- | --- |
| **Последовательность действий** | |
| **Действия пользователя** | **Отклик системы** |
| 1. Пользователь вводит в отдельное поле массив  **Исключение №1.** | 2. Система создает массив при валидной введённых данных. |
| **Исключение №1**: Введено дробное число | |
| **Действия пользователя** | **Отклик системы** |
| 1. Пользователь вводит дробное число | 2. Система выдает сообщение о невалидности введенного числа и возвращается в исходное состояние. |

3.2 Функция 2

Вычисление максимального элемента массива.

|  |  |
| --- | --- |
| **Последовательность действий** | |
| **Действия пользователя** | **Отклик системы** |
| 1. Пользователь нажимает на кнопку «Вычислить максимальное значение из элементов массива». | 2. Система в отдельное поле выводит максимальное значение массива. |
| **Предусловия**: В системе создан массив целых чисел. | |

3.3 Функция 3

Вычисление максимального элемента массива в интервале значений номеров массива.

|  |  |
| --- | --- |
| **Последовательность действий** | |
| **Действия пользователя** | **Отклик системы** |
| 1. Пользователь вводит начальный индекс.  3. Пользователь вводит конечный индекс.  5. Пользователь нажимает на кнопку «Вычислить максимальное значение из элемента класса ограничено начальным и конечным индексом».  **Исключение №1.** | 2. Система создаёт начальный индекс, с которого начнётся выборка элементов.  4. Система создаёт конечный индекс, на котором закончится выборка элементов.  6. Система в отдельное поле выводит максимальное значение массива из заданного интервала. |
| **Предусловия**: В системе создан массив целых чисел. | |
| **Исключение №1**: Введено неположительное число или дробное число | |
| **Действия пользователя** | **Отклик системы** |
| 1. Пользователь вводит дробное число или неположительное число | 2. Система выдает сообщение о невалидности введенного числа и возвращается в исходное состояние. |