**<https://github.com/AlexFatkin/HealthCalculator5.git>**

**st\_main.py**

[**https://share.streamlit.io/**](https://share.streamlit.io/)

[**https://healthcalculator2.streamlit.app/**](https://healthcalculator2.streamlit.app/)

Добрый день.

Минимальный рабочий продукт можно тестировать на [**https://healthcalculator2.streamlit.app/**](https://healthcalculator2.streamlit.app/)

Предлагаю обсудить калибровочные диаграммы калькулятора здоровья в воскресенье 28 апреля в 19:00 часов?

0

PyCharm предложит методы и атрибуты во время ввода. Поэтому, если вы, например, наберете str()., PyCharm отобразит список с предлагаемыми элементами для подражания, который сузится, как только вы продолжите вводить. Вы можете принудительно выполнить *завершение кода*, нажав Ctrl + Space.

Если вы нажмете Ctrl + P внутри функции или метода, вы также сможете отобразить его параметры.

To see the function docstring, you should press **Ctrl+Shift+I**

**Снятие ограничения (Close Restriction)**

Приложение «Система биржевой торговли» (**Trading**),

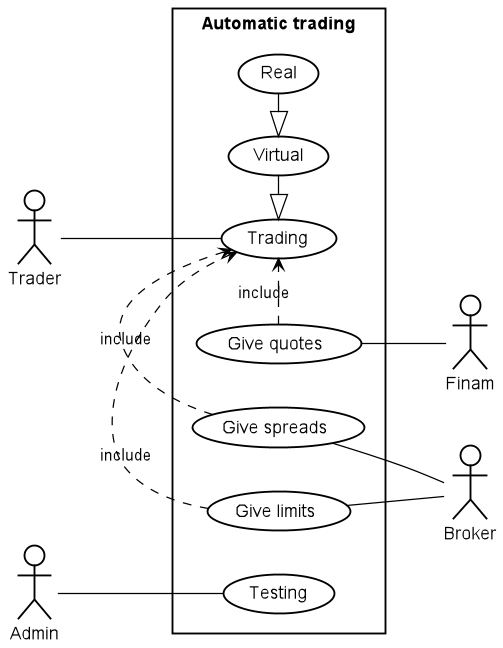
(D:\Mission\5.Programming\Python\Trade11\model\Game11.2.ipynb), достигло ограничений разработки на скриптах в виде монолитного блока, и стала сложным для понимания и сопровождения.

Для снятия данного ограничения проводим рефакторинг из скриптов в объектную модель. Вначале описываем на языке UML *(Unified Modeling Language)* основные схемы приложения**:**

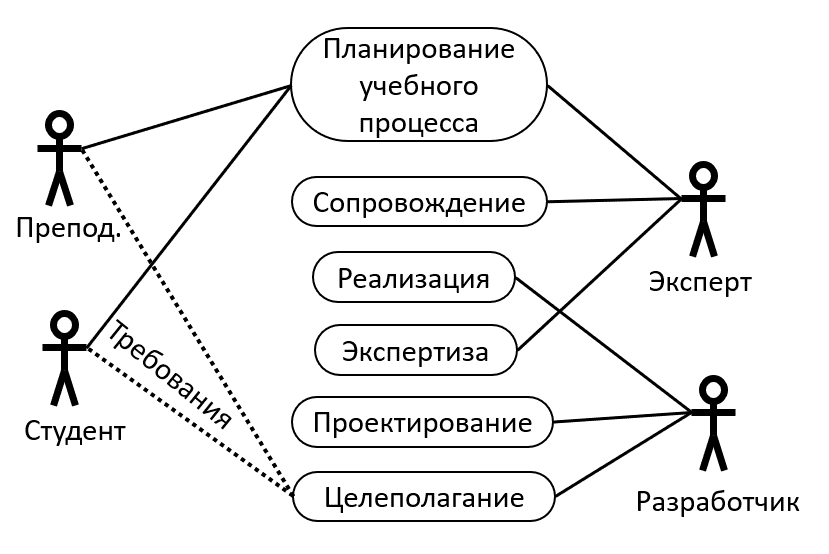
1. Вариант использования **(Use Case**), Сценарий активности (**Scenario)**   
2. Схемы предметной области (**Components, Classes, Sequence, State**)   
3. Тестирование (**T**est) и Рефакторинг(**Refactoring**)  
4. Кодирование (**Code)** и Создание интерфейса (**Interface**).

Разрабатываемое приложение и UML схемы находятся по адресу: D:\Mission\5.Programming\Python\Trade11\model\Game11\_3.py

**Варианты использования (Use case)**

****

Варианты использования **приложения Расписание**



* **Целеполагание -** определение корневого ограничения и постановка цели
* **Проектирование -** создание модели решения
* **Экспертиза -** оценка решения экспертами
* **Реализация -** управление осуществлением приложения
* **Сопровождение -** адаптация окружения
* **Окружение -** область взаимодействия

## Сценарий проекта Расписание

**Процесс:** Устранение ошибок и минимизация предупреждений в расписании.

**Участники процесса**: Разработчик приложения (Разработчик), Эксперт по расписанию (Эксперт), Студент, Преподаватель.

**Предусловия** (условия, без которых невозможно начать процесс)  
Известны списки **Ресурсов**:   
**Субъектов:** преподавателей и их нагрузок, студентов по группам;  
**Объектов:** аудиторий по корпусам, расписание и время трансфера между корпусами, дисциплин.

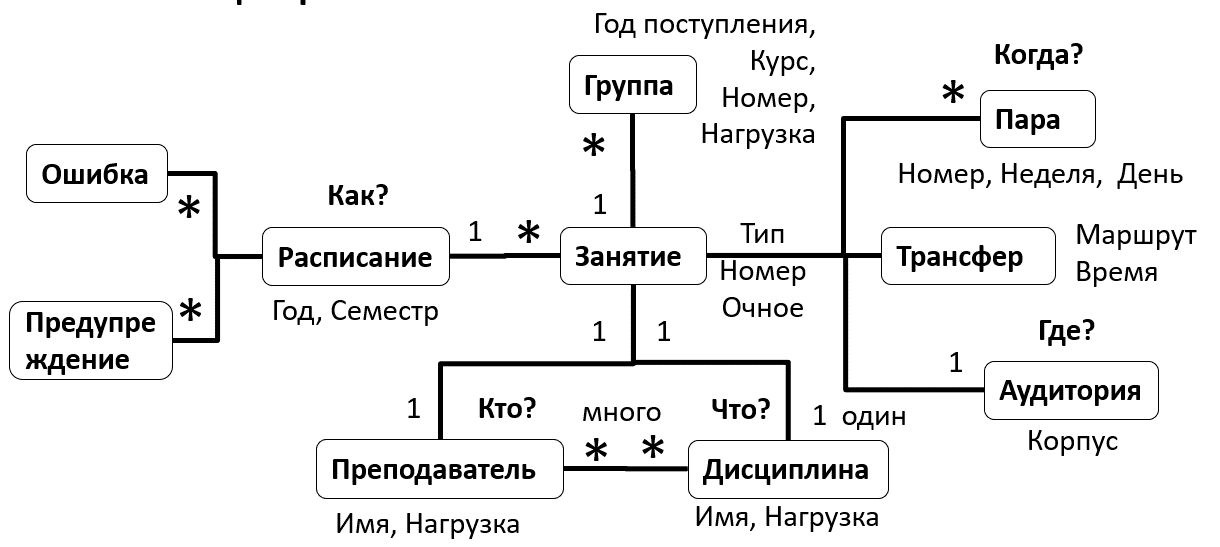
**Детальное описание процесса**

1. Разработчик опрашивает Студентов и Преподавателей, выясняя нежелательные явления (НЯ) в расписании.
2. Разработчик обобщает НЯ, определяет корневые ограничения и проектирует модель их устранения.
3. Разработчик реализует полученную модель в виде программного приложения Расписание (Приложение) на языке Python.
4. Эксперт вносить Ресурсы в Расписание.
5. Эксперт планирует учебный процесс распределяя Ресурсы
6. Приложение показывает ошибки и предупреждения, заложенные Разработчиком.
7. Эксперт устраняет ошибки и минимизирует предупреждения.
8. После утверждения общего расписания Приложение рассылает индивидуальные расписания Субъектам.
9. Эксперт сопровождает расписание, внося изменения (замена: преподавателя, аудитории, трансфера).
10. Приложение, после изменения Ресурса, оповещает Субъектов, которых касается данное изменение.

**Стандартные расширения (ветвления) процесса:**

5а. На втором этапе проекта возможно автоматизация распределения Ресурсов с помощь нейронной сети.

**Диаграмма классов приложения Расписание**



**Цели двух этапов автоматизации расписания:**

**1. Проверка расписания:** исключение ошибок и минимизация предупреждений

* 1. Ограничения по типу занятий (Предупреждение)
  2. Ограничения по трансферу (Ошибка)
  3. Персонализация расписания (Разовое решение)
  4. Окна между занятиями (Предупреждение)
  5. Ограничения занятости ресурсов (Ошибка)

1. **Составление расписания:** распределение ресурсов (нейронная сеть)





https://medlineplus.gov/ency/article/003399.htm?\_gl=1\*1s50tgd\*\_ga\*NTE5MzExNzkuMTcxMzU0NzAzNA..\*\_ga\_7147EPK006\*MTcxMzU0NzAzMy4xLjAuMTcxMzU0NzAzMy4wLjAuMA..\*\_ga\_P1FPTH9PL4\*MTcxMzU0NzAzMy4xLjAuMTcxMzU0NzAzMy4wLjAuMA..

## Normal Results

For resting heart rate:

* Newborns 0 to 1 month old: 70 to 190 beats per minute
* Infants 1 to 11 months old: 80 to 160 beats per minute
* Children 1 to 2 years old: 80 to 130 beats per minute
* Children 3 to 4 years old: 80 to 120 beats per minute
* Children 5 to 6 years old: 75 to 115 beats per minute
* Children 7 to 9 years old: 70 to 110 beats per minute
* Children 10 years and older, and adults (including seniors): 60 to 100 beats per minute
* Well-trained athletes: 40 to 60 beats per minute

<https://translated.turbopages.org/proxy_u/en-ru.ru.ae00833a-6622a65e-e71d8e92-74722d776562/https/medlineplus.gov/ency/article/003399.htm?__ya_mt_enable_static_translations=1>

## Нормальные результаты

Для определения частоты сердечных сокращений в состоянии покоя:

* У новорожденных в возрасте от 0 до 1 месяца: от 70 до 190 ударов в минуту
* У младенцев в возрасте от 1 до 11 месяцев: от 80 до 160 ударов в минуту
* У детей от 1 до 2 лет: от 80 до 130 ударов в минуту
* У детей от 3 до 4 лет: от 80 до 120 ударов в минуту
* У детей 5-6 лет: от 75 до 115 ударов в минуту
* У детей от 7 до 9 лет: от 70 до 110 ударов в минуту
* У детей 10 лет и старше и взрослых (включая пожилых людей): от 60 до 100 ударов в минуту
* У хорошо тренированных спортсменов: от 40 до 60 ударов в минуту

