|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Понедельник** | **Вторник** | **Среда** | **Четверг** | **Пятница** | **Суббота** | **Воскресение** |
|  |  |  |  |  |  |  |
| **Место для пометок – «Изменение основного расписания»** | | | | | | |
|  |  |  |  |  |  |  |
|  | | | | | | |

Общие пометки (не относятся к конкретному дню):

1) **Текущие задачи:**

* Повторить всё: **Python** весь – типы данных и их методы, функции (args, kwargs) и замыкания, lambda, map, filter, reduce, модули sys, библиотеки встроенные (работа с файлами, датой и временем), регулярки(видео YouTube в STUDIES), генераторы+итераторы(видео YouTube в STUDIES), декораторы, ООП, Исключение, Метаклассы, логирование, **PostgreSQL, Django+MTV+тестирование, DRF, Docker, Git, разворачивание проекта+установка всех необходимых библиотек (flake8, isort,** . env-file**), модулей, requirements, Redis, Celery, архитектура backend (видео в избранном в VK) – до 03.12.2023**
* ~~Подредактировать интернет-магазин (админка, кэш, и т.д.), попробовать установить свои проекты с GitHub в вирт. окружениях и посмотреть, как это будет работать и выглядеть~~
* ~~Перенести в докер свои приложения~~
* ~~Сделать Readme в приложениях~~
* **Посмотреть курс по Pytest (ссылка в VK в избранном)**
* Начать откликаться на вакансии + решать в день по 1-3 задачи с LeetCode
* Начать делать проект с ботом (асинхронное программирование + подрубить его к проекту django), параллельно учить дальше

**Обучение в целом:**

**Изучить:**

1) Nginx, Gunicorn

2) Bash

3) как Python работает с оперативной памятью

4) Тестирование в Python – pytest, Unittest, т.д.

5) Хеш-таблицы. Структуры данных: списки, деревья, графы. Алгоритмы и структуры данных. Графы, деревья, алгоритмы на графы (поиск в ширину, в глубину и алгоритм дейкстры) Книга: Algorithms + Data Structures = Programs Оценка сложности алгоритма

6) Регулярные выражения

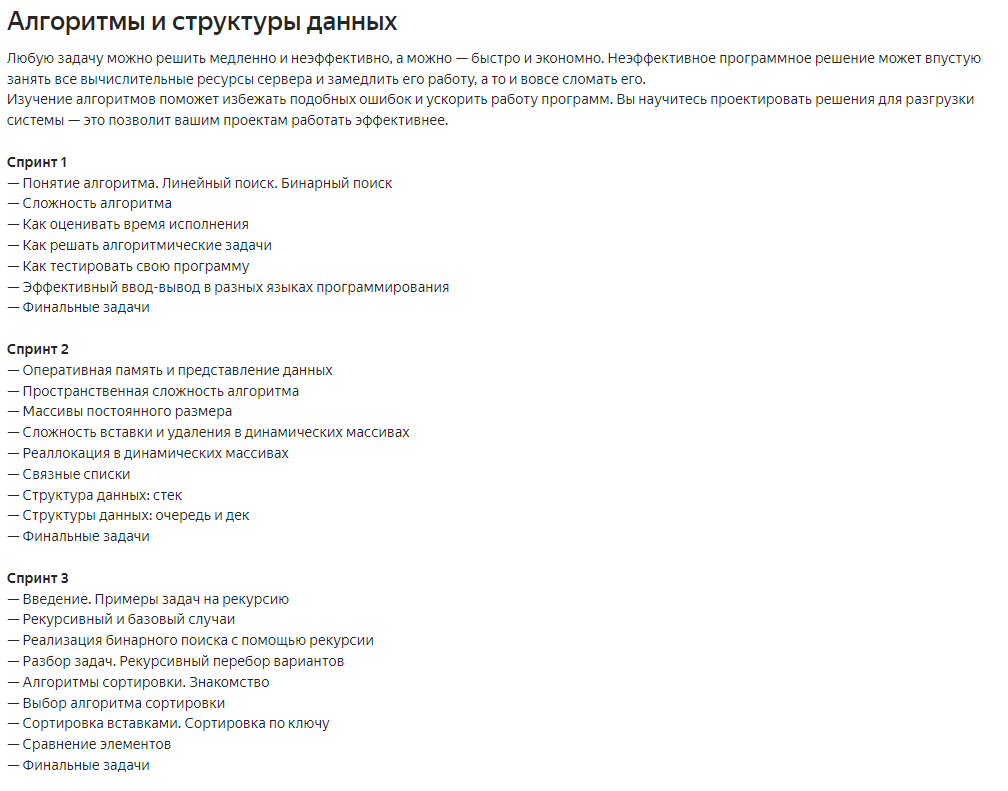
7) Асинхронность в Python

8) Рекурсия

9) Веб-фрейморк - FastAPI

10) JS – база

11) Flask



**‼‼ ПРОЙДЕННЫЕ (НА ПОСТОЯННОЕ ПОВТОРЕНИЕ) ‼‼**

1) **Python** весь – типы данных и их методы, функции (args, kwargs) и замыкания, lambda, map, filter, reduce, встроенные функции all(), any(), zip(), enumerate(), модуль sys, библиотеки встроенные (работа с файлами, датой и временем), (модули, пакеты, импорт, \_\_name\_\_ = \_\_main\_\_), библиотека request, модуль secrets и string для генерации паролей, регулярки, генераторы и итераторы, декораторы, JSON, работа с разными форматами, логирование

2) ООП (курс – основы + принципы ООП (инкапсуляция, наследование, полиморфизм), геттеры, сеттеры, деллиторы, области видимости, защищенные и приватные атрибуты, classmethod, staticmethod, свойства (property), вычисляемые свойства (тоже через property, наследование, множественное наследование, магические методы (в том числе hash и eq), super()+ делегирование, \_\_new\_\_ и \_\_call\_\_ (Паттерн Singleton), Паттерн Моносостояние, дескрипторы

3) SOLID, KISS, DRY, шаблоны проектирования, чистый код, отладка кода

4) Исключение (try, except, finaly, else; иерархия исключений, создание собственных исключений, наследуясь от Exception)

5) Хеширование в Python

6) Метаклассы (что это, зачем нужны, пример использование в части API ORM Django – на моделях)

7) PostgreSQL и в целом БД

8) Виртуальное окружение (создание через консоль + через Pycharm) + файл-установщик (freeze> name\_file.txt) … + теорию посмотреть

9) Celery + Redis (кеширование данных, создание асинхронных действий – отправка email для подтверждения почты)

10) Использование Postman программы для выполнения запросов

11) WEB: HTTP – запросы (статусы 100-500 …)/ HTTPS, OSI, TCP/IP, Soap, WebSockets, REST

12) Использование пакетов – flake8 - форматер, isort – литер

13) Скрытие данных (ключей и прочего от пользователей) – файлом. env и добавлением gitignore, requirements.txt

14) Декораторы

15) Django REST Framework (DRF), документация swager, Cookies, Авторизация, Аутентификация.

16) Chrome developer tools – панель разработчика в браузере

17) API, Rest API

18) Git, GitHub

19) Linux (основные команды). Bash - скрипты

20) Docker

21) Основы HTML, CSS, Bootstrap

**Проекты для портфолио в GitHub:**

1) ☺ Сделать хорошую админку в проекте **store + подредактировать для нормального вида**

2) ☺ Попробовать добавить form в проект **converter\_curr**

3) ☺ Сделать сайт-блог, привязать **DRF** к нему, оформить admin panel (**commit for API: api work version with admin panel**).

4) Сделать скрипт – парсер какого-то сайта (например avito – видео есть в ютубе)

5) Телеграм-бот / начало есть – продолжить курс

