**Модуль №5. Задание №2**

Решение задач присылать ссылкой на github репозиторий, в Discord чат «с-домашняя-работа».

Структура репозитория

Корневая папка – data-structures

Подпапка – stack

файл с реализацией – stack.py

файл с тестами – test\_stack.py

Подпапка – queue

файл с реализацией – queue.py

файл с тестами – test\_queue.py

Структуры данных

**Задача №1.**

Разработайте класс «Stack» описывающий абстрактный тип данных (АТД) Стек. Класс описывается атрибутами top-вершина стека и size-размер стека. Необходимо реализовать протокол для работы с АТД Стэк, который состоит из следующих операций: добавление элемента на вершину стека (push); удаление и чтение элемента с вершины стека (pop); просмотр значения элемента на вершине стека (peek); проверка на пустоту стека (is\_empty).

**Задача №2.**

Разработайте класс «Queue» описывающий абстрактный тип данных (АТД) Очередь. Класс описывается атрибутами head-голова очереди, tail-хвост очереди и size-размер очереди. Необходимо реализовать протокол для работы с АТД Очередь, который состоит из следующих операций: добавление элемента в хвост очереди (enqueue); удаление и чтение элемента с головы очереди (dequeue); просмотр значения элемента с головы очереди (peek); проверка на пустоту стека (is\_empty).

**Задание №3.**

Для задачи №1 и №2, необходимо провести модульное тестирование:   
1. Составить тест-план с описанием тестовых случаев трех видов: проверка на корректные данные, проверка на некорректные данные, граничные проверки.

2. Составить тестовые функции по тест-плану.

**По желанию**

**Задание №2**

Карточка персоны содержит фамилию и дату рождения. Реализовать класс ListPerson (двунаправленный список) для работы с картотекой персоналий. Класс должен содержать массив карточек персон. Реализовать методы добавления и удаления карточек персон, вставки на определенную позицию, проверки на пустоту. Реализовать операции объединения двух карточек персон. Реализовать метод поиска персоны по фамилии.