**Параллельный запуск процессов**

1. Реализовать класс ProcessController, который организует очередь заданий и параллельное выполнение заданий из очереди. Данный класс должен содержать следующие методы:

* set\_max\_proc — аргумент: n. Метод устанавливает ограничение: максимальное число одновременно выполняемых заданий не должно превышать n.
* start — аргументы: tasks, max\_exec\_time:

1. tasks — список заданий. Задание представляет кортеж вида: (функция, кортеж входных аргументов для функции).  
   *Пример*: tasks = [(function0, (f0\_arg0, f0\_arg1)), (function1, (f1\_arg0, f1\_arg1, f1\_arg2)), (function2, (f2\_arg0, ))]
2. max\_exec\_time — максимальное время (в секундах) работы каждого задания из списка tasks

Данный метод помещает в очередь все задания из tasks. В случае, если не достигнуто ограничение на максимальное число одновременно работающих заданий, метод запускает выполнение заданий из очереди до тех пор, пока не будет достигнуто это ограничение. Запуск задания представляет порождение нового процесса, который выполняет соответствующую функция с её аргументами. При этом каждый запущенный процесс для задания из tasks не должен работать дольше max\_exec\_time.

* wait — (без аргументов) ждать пока не завершат своё выполнение все задания из очереди.
* wait\_count — (без аргументов) возвращает число заданий, которые осталось запустить.
* alive\_count — (без аргументов) возвращает число выполняемых в данный момент заданий.

2. Придумать пример, который продемонстрирует корректность реализации класса ProcessController.

3. Написанный код на языке python должен соответствовать соглашению PEP8.