

---

## Часть 1

### Протокол жизнь

Методы:

1. Новый день. () -> ()

### Протокол управление АЭС

Поля:

1. Выработанная энергия за текущий период (get). Int
2. Здоровье счетчиков (get). [Int]
3. Работает ли (get). Bool

Методы:

1. Сбросить показатели за текущий период. () -> ()
2. Добавить счетчик. (Счетчик) -> ()

### Класс АЭС: Жизнь

Управление АЭС должно быть только через протоколы. За управление АЭС будет отвечать класс центр работы АЭС.

Поля:

1. Счетчики. [Счетчики]
2. Работает ли производство. Bool
3. Показатели за последний день. Int

Методы:

1. Остановить производство. () -> (). Во время остановки производства каждый новый день ничего не меняет.
2. Запустить производство. () -> (). Во время работы производства каждый новый день меняет показатели.
3. П: Новый день. () -> (). Должен поменять показатели, если работает производство
4. Private. Почистить неработающие счетчики. () -> (). Должен чистить все счетчики, у которых здоровье < 1. Вызывается в новом дне.

### Класс Счетчик: Жизнь

Поля:

1. Private + свойство (get). Здоровье счетчика. Int. Тратится по одному в день. (0 - 20)
2. Функция вырабатываемой энергии в день относительно здоровья. (Int) -> (Int)
3. Энергия за сегодня (get). Int.

Методы:

1. П: Новый день. () -> (). Должен поменять здоровье.

---

## Часть 2

### Класс центр работы АЭС: Протокол управление АЭС

Индексирование(get). Счетчик?. По индексу должен возвращаться счетчик или nil

Поля:

1. Количество рабочих. Int. willSet { если new value < old value, то писать (можно прямо в willSet), что количество рабочих уменьшается }, didSet { если рабочих < чем счетчиков, то останавливать производство. Так же писать сколько рабочих добавилось на производство. }
2. Private. АЭС.
3. Все свойства протокола

Методы:

1. Все методы протокола

Должна возможность создать класс так (= N), где N - число рабочих. Вместе с классом создается класс АЭС

### Класс стэк с уникальными элементами. Generic

Поля:

1. Контейнер информации

Методы:

1. Push. Добавить элемент к концу контейнера. Во время добавление надо проверить, чтобы такого элемента еще не было в контейнере.
2. Pop. Взять последний элемент с конца контейнера.
3. Расширить контейнер методом + и +=, который будет добавлять в первый элементы второго.
4. Переписать вчерашний метод суммирования для текущего контейнера.
- 5\*. Переписать метод map для данного контейнера

Создать префикс операцию ^, которая будет брать строку и разбивать ее на символы, которые запишутся в класс уникального стека и функция вернет этот стек.