

# Graph SLAM Homework

Mikhail Korobkin, Dmitri Kovalenko

*Yandex SDC*

2022.03.23

## Задача 1

Правильно реализованный класс `LandmarkObservationEdge` инициализирует поля `inf`, `error` и метод `compute_error`. Важно считать ошибку в локальной системе координат.

Для верной реализации, юнит-тест будет проходить:

```
nosetests -s unit.test_edges:test_landmark_observation_edge
```

## Задача 2

Обеспечить загрузку рёбер-наблюдений маяков. За этот процесс отвечает метод `LandmarkConstraintBuilder.add_event`, конструируя объекты `LandmarkObservationEdge`.

В фабрике `LandmarkConstraintBuilder` важно различать ситуации, когда наблюдение  $i$ -го маяка встретилось впервые. В этом случае, фабрике следует создать и вершину-маяк `Landmark`, и ребро с его наблюдением.

Важно поддерживать ассоциацию между вершинами-маяками и всеми ребрами, с ними связанными, через объекты `Feature`.

Далее, показать сходимость оптимизационной задачи с маяками при наличии шума. Взять `timeline_N.json`, где  $N = m \bmod 55$ ,  $m$  - номер вашей строчки в списке, в который вам следует вписать себя самостоятельно. <https://tinyurl.com/4h8yb8jj>  
При верной реализации юнит-тест должен проходить:

```
nosetests -s unit.test_graph:test_graph_optimization_with_landmarks
```

Выложить в систему anytask выражение, складывающееся из маяков после оптимизации и картинки с визуализацией.