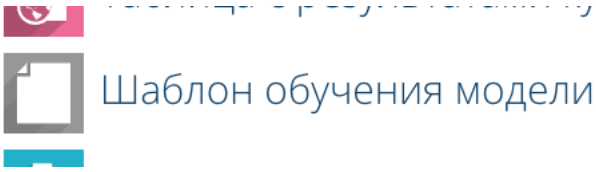
Задание 3.

Для обучения модели word2vec необходимо получить файл шаблона у преподавателя. При необходимости или наличии индивидуальных предпочтений исполнитель может использовать собственный метод обучения модели, если он удовлетворяет остальным пунктам Заданий. Для корректной работы шаблонного проекта необходимо установить библиотеки (genism, pandas), используя средства среды разработки или операционной системы. Далее необходимо создать проект, в который будет импортирован файл шаблона и собранный датасет, изменить параметры для корректной работы системы и сохранить файл .model, полученный в ходе работы проекта.

Для обучения модели необходимо скачать шаблон из соответствующего раздела курса (https://online.mospolytech.ru/course/view.php?id=12946).



Далее в любой удобной среде разработке добавьте файл шаблона в проект. После этого необходимо импортировать в проект файл датасета из предыдущего шага. Сохраните его в папку “resources” в проекте. Запишите путь к файлу в параметр.

```
10  with open('resources/название_файла_с_текстом.tsv') as f:
```

Запишите несколько слов из текстов работ в параметры методов для проверки качества работы модели. Эти слова должны содержаться в текстах датасета. Они используются для проверки того, что слова вошли в представление.

```
34  print(model.wv['тестовое_слово_по_теме_работы'])
35  # print(model.corpus_count)
36
37  # Train the model
38  model.build_vocab(response_base, update=True)
39  model.train(response_base, total_examples=model.c
40  print(model.wv["тестовое_слово_по_теме_работы"])
```

И еще три слова для сравнения близости их векторных представлений (здесь из слов 2 и 3 выбирается наиболее близкое по вектору слову 1).

```
45  print(model.wv.most_similar_to_given("слово_1", ["слово_2", "слово_3"]))
```

Например, "болезнь", ["заболевание", "инфекция"].

Укажите название файла для сохранения модели и запустите программу.

```
42    model.save("resources/model_name.model")
```

Параметры обучения модели задаются в классе World2Vec. Цель обучения состоит в том, чтобы большее число нужных для работы системы слов попали в представление. Для этого нужны проверки выше. Если слова из текстов не попадают в представление, измените значение параметра min_count. Понизьте значение для уменьшения порога вхождения.

```
23    model = Word2Vec(
24        sentences=response_base,
25        min_count=30,
26        window=2,
27        vector_size=64,
28        alpha=0.03,
29        negative=10,
30        min_alpha=0.0007,
31        sample=6e-5
32    )
```