1)Создать двух пользователей

from NewsPaper.models import User

user1 = User.objects.create\_user(‘Иванов Иван Иванович’)

user2 = User.objects.create\_user(‘Петров Петр Петрович’)

2) создать два объекта модели Author связанные с пользователями.

from NewsPaper.models import Author

Author.objects.create(user\_name = user1)

Author.objects.create(user\_name = user2)

3) Добавить 4 категории в модель Category.

from NewsPaper.models import Category

Category.objects.create(name\_category = ‘Спорт’)

Category.objects.create(name\_category = ‘Политика’)

Category.objects.create(name\_category = ‘Образование’)

Category.objects.create(name\_category = ‘Культура’)

4) Добавить 2 статьи и 1 новость.

from NewsPaper.models import Post

user\_name = Author.objects.get(id=1)

Post.objects.create(author\_name = user\_name,

types\_post = ‘AR’,

header = ‘Заголовок1’,

content = ‘Статья1’)

Post.objects.create(author\_name = user\_name,

types\_post = ‘AR’,

header = ‘Заголовок2’,

content = ‘Статья2’)

user\_name = Author.objects.get(id=2)

Post.objects.create(author\_name = user\_name,

types\_post = ‘NE’,

header = ‘Заголовок3’,

content = ‘Новость1’)

5) Присвоить статьям и новостям категории (как минимум к одной из них применить не менее 2х категорий).

Post.objects.get(id=1).category\_post.add(Category.objects.get(id=1))

Post.objects.get(id=1).category\_post.add(Category.objects.get(id=4))

Post.objects.get(id=2).category\_post.add(Category.objects.get(id=2))

Post.objects.get(id=3).category\_post.add(Category.objects.get(id=3))

6) Создать как минимум 4 комментария к разным объектам модели Post (в каждом объекте как минимум 1 комментарий).

from NewsPaper.models import\*

Comment.objects.create(

post\_comment = Post.objects.get(id=1),

user\_comment = Author.objects.get(id=1).user\_name,

text\_comment = ‘Всё супер’)

Comment.objects.create(

post\_comment = Post.objects.get(id=1),

user\_comment = Author.objects.get(id=1).user\_name,

text\_comment = ‘Мне понравилось’)

Comment.objects.create(

post\_comment = Post.objects.get(id=2),

user\_comment = Author.objects.get(id=3).user\_name,

text\_comment = ‘интересно очень’)

Comment.objects.create(

post\_comment = Post.objects.get(id=3),

user\_comment = Author.objects.get(id=3).user\_name,

text\_comment = ‘интересно очень’)

7) Применяя функции like(), и dislike() к статьям/новостям и комментариям, скорректировать рейтинги этих объектов.

from NewsPaper.models import\*

>>> Comment.objects.get(id=1).like(1)

>>> Comment.objects.get(id=3).like(2)

>>> Post.objects.get(id=1).dislike(1)

>>> Post.objects.get(id=3).like(1)

Проверка рейтинга пользователя

>>> Comment.objects.get(id=1).rating\_comment

Проверка рейтинга поста

>>> Post.objects.get(id=3).rating\_post

8) Обновить рейтинги пользователей

>>> u1 = Author.objects.get(id=1)

>>> u1.update\_rating(5)

>>> u1.user\_rating

>>> u2 = Author.objects.get(id=2)

>>> u2.update\_rating()

>>> u2.user\_rating

9)Вывести username и рейтинг лучшего пользователя (применяя сортировку и возвращая поля первого объекта).

>>> s = Author.objects.order\_by('-user\_rating')

>>> for i in s:

. . . i.user\_name.username

. . . i.user\_rating

10)Вывести дату добавления, username автора, рейтинг, заголовок и превью лучшей статьи, основываясь на лайках/дислайках к этой статье.

>>> p = Post.objects.order\_by('-rating\_post')

>>> for i in p[:1]:

. . . i.datetime\_post

. . . i.author\_name.user\_name

. . . i.rating\_post

. . . i.header

. . . i.preview()

11)Вывести все комментарии (дата, пользователь, рейтинг, текст) к этой статье

Post.objects.all().order\_by('-rating\_post')[0].comment\_set.values('datetime\_comment',

'user\_comment','rating\_comment','text\_comment')