

Для выполнения каждого из заданий необходимо написать код, по желанию сопровождаемый комментариями.

1) В Django проекте (версии пакетов считать последними на данный момент времени) имеется много Django моделей с определенным и конечным набором экземпляров в БД. Данные модели повсеместно используются в коде проекта. Конкретные экземпляры моделей извлекаются из БД через методы моделей по полю id. Пример такой модели:

```
class DeliveryState(models.Model):

    class Meta:
        verbose_name = u"Состояние доставки"
        verbose_name_plural = u"Состояния доставок"

    STATE_NEW = 1 # Новая
    STATE_ISSUED = 2 # Выдана курьеру
    STATE_DELIVERED = 3 # Доставлена
    STATE_HANDED = 4 # Курьер сдал
    STATE_REFUSED = 5 # Отказ
    STATE_PAID_REFUSED = 6 # Отказ с оплатой курьеру
    STATE_COMPLETE = 7 # Завершена
    STATE_NONE = 8 # Не определено

    @classmethod
    def get_new(cls):
        return cls.objects.get(pk=cls.STATE_NEW)

    @classmethod
    def get_issued(cls):
        return cls.objects.get(pk=cls.STATE_ISSUED)

    @classmethod
    def get_delivered(cls):
        return cls.objects.get(pk=cls.STATE_DELIVERED)

    @classmethod
    def get_handed(cls):
        return cls.objects.get(pk=cls.STATE_HANDED)

    @classmethod
    def get_refused(cls):
        return cls.objects.get(pk=cls.STATE_REFUSED)

    @classmethod
    def get_paid_refused(cls):
        return cls.objects.get(pk=cls.STATE_PAID_REFUSED)

    @classmethod
    def get_complete(cls):
        return cls.objects.get(pk=cls.STATE_COMPLETE)

    @classmethod
    def get_none(cls):
        return cls.objects.get(pk=cls.STATE_NONE)
```

В последующем, такие методы используются в коде:

```
...
Delivery.objects.filter(state=DeliveryState.get_new())
...
if delivery.state == DeliveryState.get_refused():
    ...
```

Каким образом можно добиться того, что бы интерфейс работы с моделью `DeliveryState` не изменился (т.е. была возможность написать, к примеру, `DeliveryState.get_new()`), но в описании модели `DeliveryState` не пришлось бы декларативно описывать каждый метод `get_new`, `get_issued`, таким образом избавившись от однотипного кода.

Приведите конкретные реализации решения проблемы, которые можно использовать повсеместно в проекте (не только для модели `DeliveryState`)

2) В Django проекте (версии пакетов считать последними на данный момент времени) используется модель `Lead` и набор состояний, в котором может находиться экземпляр этой модели. Набор состояний определяется связью с другой моделью `LeadState`. Набор экземпляров модели `LeadState` строго определен:

```
class LeadState(models.Model):

    # pk экземпляров модели
    STATE_NEW = 1 # Новый
    STATE_IN_PROGRESS = 2 # В работе
    STATE_POSTPONED = 3 # Приостановлен
    STATE_DONE = 4 # Завершен

    name = models.CharField(
        u"Название",
        max_length=50,
        unique=True,
    )

class Lead(models.Model):

    name = models.CharField(
        max_length=255,
        db_index=True,
        verbose_name=u"Имя",
    )

    state = models.ForeignKey(
        LeadState,
        on_delete=models.PROTECT,
        default=LeadState.STATE_NEW,
        verbose_name=u"Состояние",
    )
```

Возможность перехода из одного состояния в другое строго определена (с учетом направления перехода):

```
Новый -> В работе
В работе -> Приостановлен
В работе -> Завершен
Приостановлен -> В работе
Приостановлен -> Завершен
```

Другие переходы невозможны. В момент перехода из одного состояния в другое нужно вызвать методы, который реализует дополнительную бизнес логику (произвольные методы класса модели `Lead` для каждого из типов переходов, заранее определенные. Учсть, что код данных методов может быть произвольный: от изменения полей модели до отправки email и SMS). Смена состояния и выполнение бизнес логики должно быть согласованным с точки зрения данных.

Каким образом вы бы реализовали эту задачу? Необходимо написать код для решения данной задачи в рамках нового Django проекта, покрыть его unit тестами (не обязательно 100% покрытие).

3) В рамках Django проекта есть метод (не важно какого класса) покупки уникального товара (модель `Product`) с неким уникальным идентификатором `item_id`:

```
@classmethod
def buy(cls, user, item_id):
    product_qs = Product.objects.filter(item_id=item_id)
    if product_qs.exists():
        product = product_qs[0]

    if product.available:
        # списание средств со счета пользователя
        user.withdraw(product.price)

        # информация о купленном товаре
        send_email_to_user_of_buy_product(user)

        product.available = False
        product.buyer = user
        product.save()

    return True
else:
    return False
```

Есть ли в реализации данного метода потенциальные проблемы(проблема)? Интерфейс метода не брать во внимание.

4) Дана вложенная структура данных (Python dict), описывающая свойства тарифа, например: <https://pastebin.com/1r4H0CwT>

Реализуйте функцию, которая будет выводить в консоль описание такой структуры данные в удобочитаемом виде. Например, для вышеуказанного описания тарифа функция должна вывести в консоль

```
root
  Внутрисетевой роуминг
    Internet
      Значение А
    MMS
      Входящие: 0.00
      Исходящие: 6.45
      Междугородние
        Входящие: 0.00
        Исходящие
          Значение В
      Местные
        Входящие: 0.00
  Домашняя сеть
    Internet
      Значение С
```

Важно: структура данных может иметь произвольный уровень вложенности, но структура данных такая, как в примере.