Московский государственный технический

университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №6

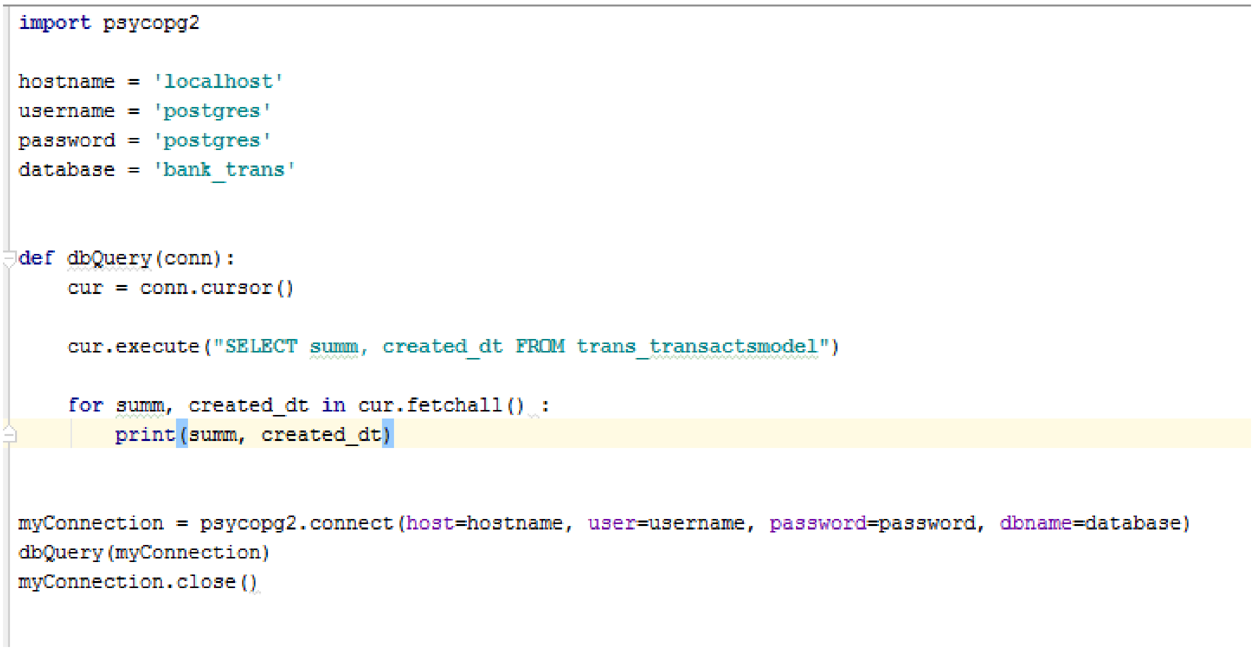
Выполнила:

студент группы ИУ5-52

Кучаева К.И.

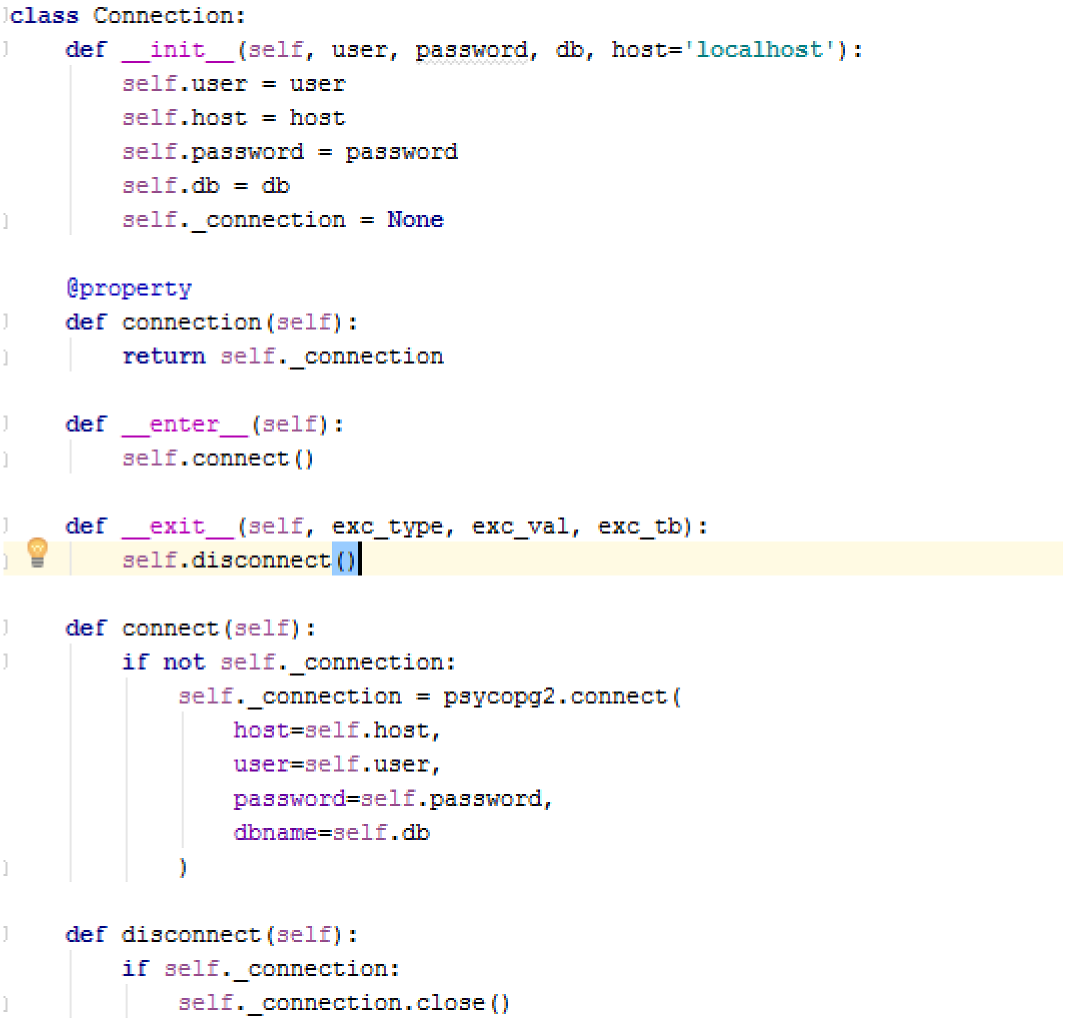
Москва, 2017 г

Скрипт с подключением к БД

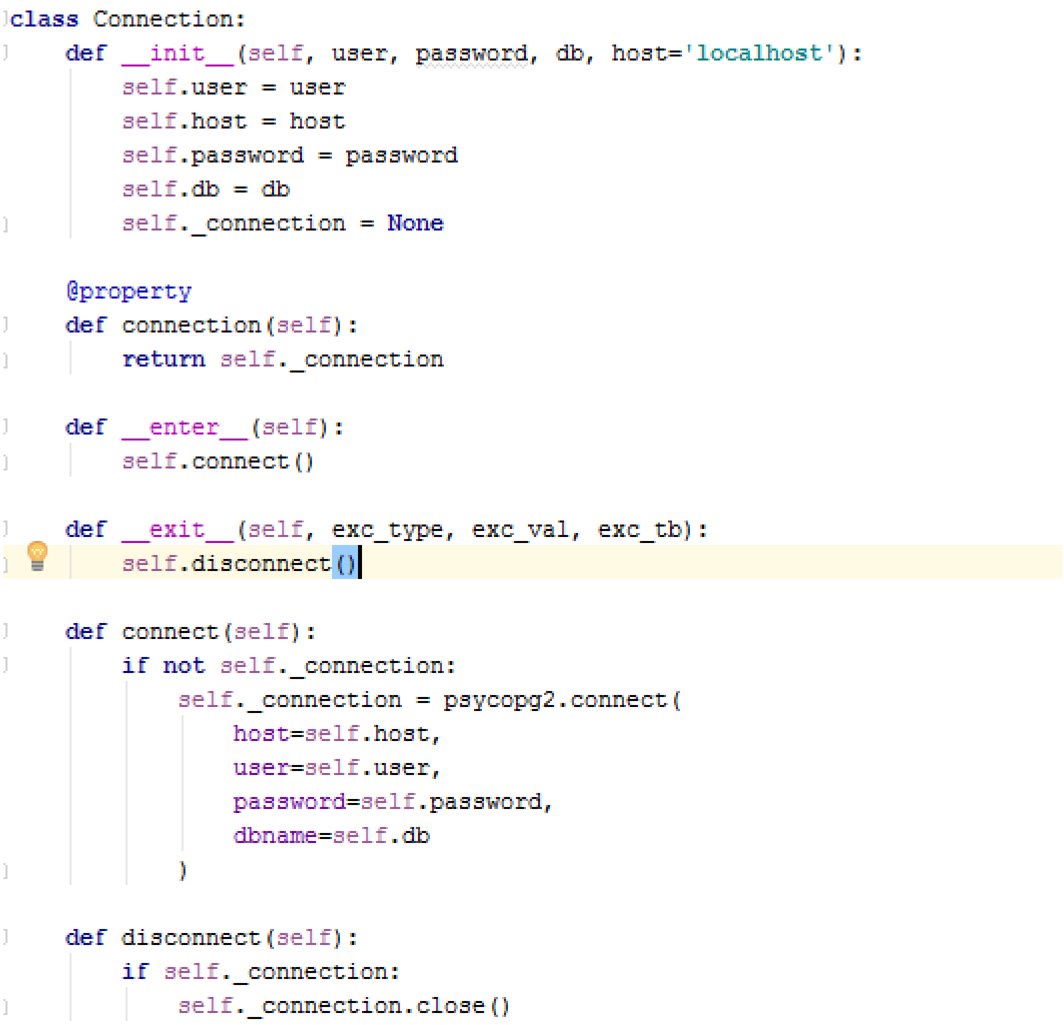


```python
import psycopg2

hostname = 'localhost'
username = 'postgres'
password = 'postgres'
database = 'bank_trans'


def dbQuery(conn):
    cur = conn.cursor()

    cur.execute("SELECT summ, created_dt FROM trans_transactsmodel")

    for summ, created_dt in cur.fetchall() :
        print(summ, created_dt)


myConnection = psycopg2.connect(host=hostname, user=username, password=password, dbname=database)
dbQuery(myConnection)
myConnection.close()
```

Класс

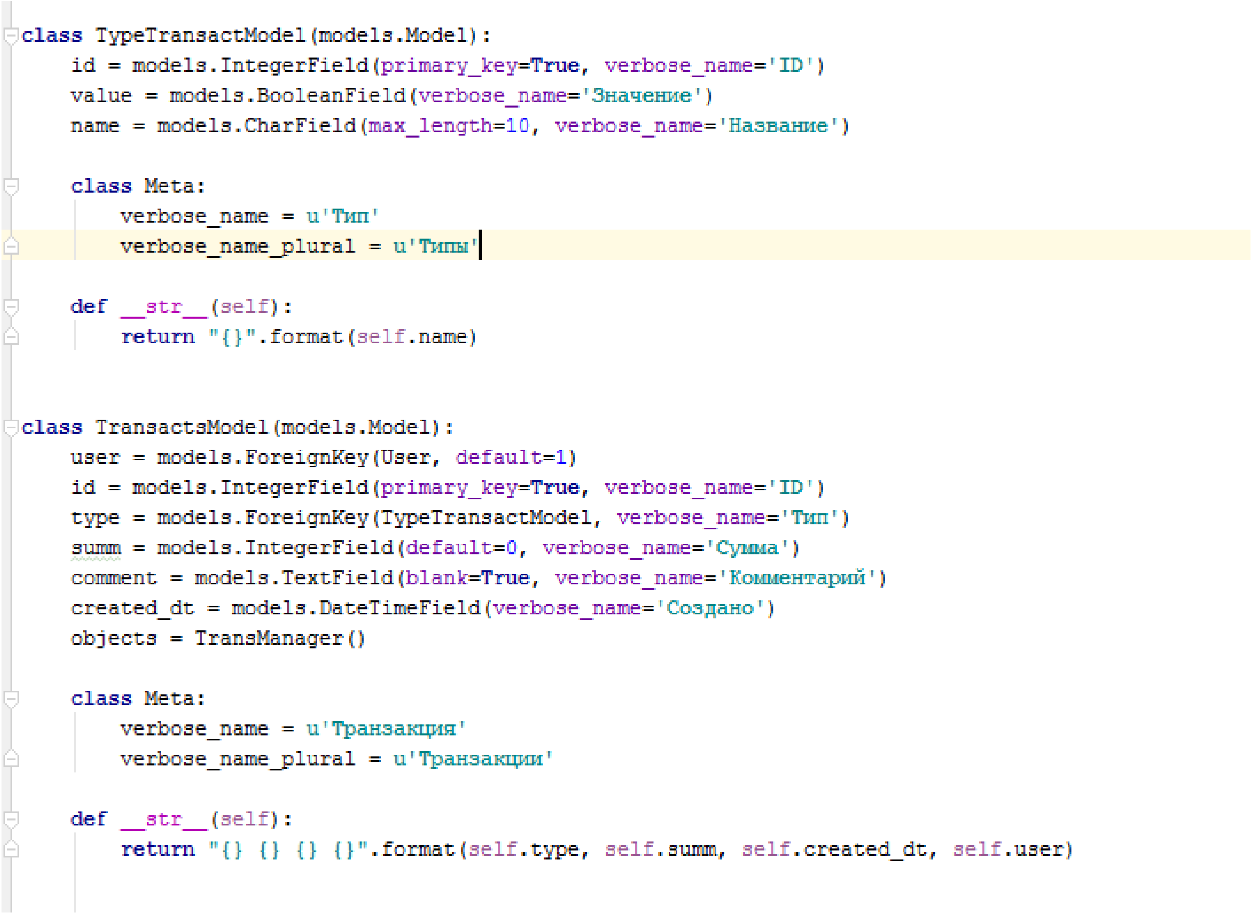


```python
class Connection:
    def __init__(self, user, password, db, host='localhost'):
        self.user = user
        self.host = host
        self.password = password
        self.db = db
        self._connection = None

    @property
    def connection(self):
        return self._connection

    def __enter__(self):
        self.connect()

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.disconnect()

    def connect(self):
        if not self._connection:
            self._connection = psycopg2.connect(
                host=self.host,
                user=self.user,
                password=self.password,
                dbname=self.db
            )

    def disconnect(self):
        if self._connection:
            self._connection.close()
```



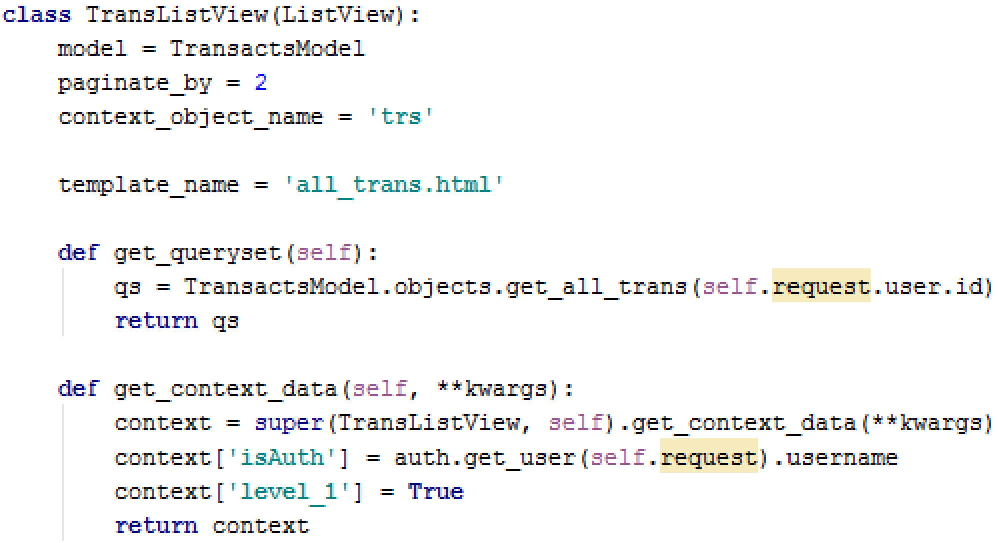
```python
class Connection:
    def __init__(self, user, password, db, host='localhost'):
        self.user = user
        self.host = host
        self.password = password
        self.db = db
        self._connection = None

    @property
    def connection(self):
        return self._connection

    def __enter__(self):
        self.connect()

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.disconnect()

    def connect(self):
        if not self._connection:
            self._connection = psycopg2.connect(
                host=self.host,
                user=self.user,
                password=self.password,
                dbname=self.db
            )

    def disconnect(self):
        if self._connection:
            self._connection.close()
```

Модели

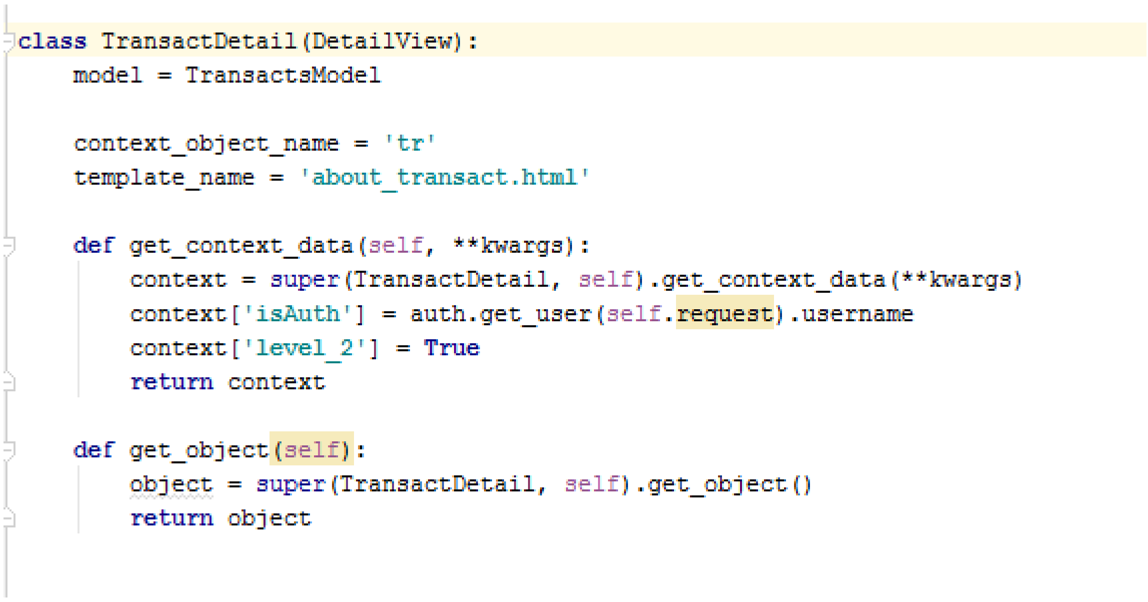


```python
class TypeTransactModel(models.Model):
    id = models.IntegerField(primary_key=True, verbose_name='ID')
    value = models.BooleanField(verbose_name='Значение')
    name = models.CharField(max_length=10, verbose_name='Название')

    class Meta:
        verbose_name = u'Тип'
        verbose_name_plural = u'Типы'

    def __str__(self):
        return "{}".format(self.name)


class TransactsModel(models.Model):
    user = models.ForeignKey(User, default=1)
    id = models.IntegerField(primary_key=True, verbose_name='ID')
    type = models.ForeignKey(TypeTransactModel, verbose_name='Тип')
    summ = models.IntegerField(default=0, verbose_name='Сумма')
    comment = models.TextField(blank=True, verbose_name='Комментарий')
    created_dt = models.DateTimeField(verbose_name='Создано')
    objects = TransManager()

    class Meta:
        verbose_name = u'Транзакция'
        verbose_name_plural = u'Транзакции'

    def __str__(self):
        return "{} {} {} {}".format(self.type, self.summ, self.created_dt, self.user)
```

VIEW

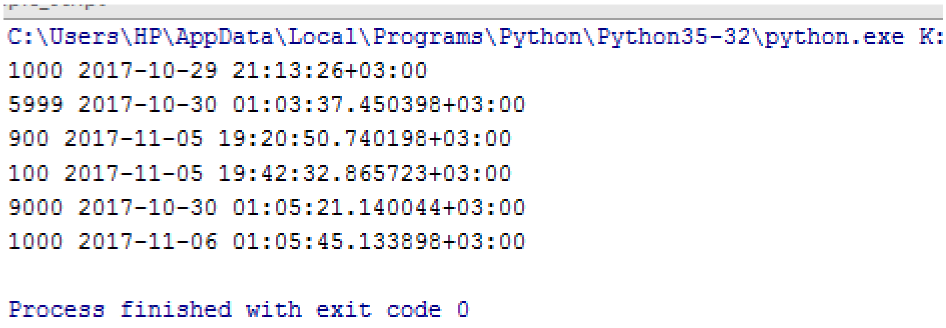


```python
class TransListView(ListView):
    model = TransactsModel
    paginate_by = 2
    context_object_name = 'trs'

    template_name = 'all_trans.html'

    def get_queryset(self):
        qs = TransactsModel.objects.get_all_trans(self.request.user.id)
        return qs

    def get_context_data(self, **kwargs):
        context = super(TransListView, self).get_context_data(**kwargs)
        context['isAuth'] = auth.get_user(self.request).username
        context['level_1'] = True
        return context
```



```python
class TransactDetail(DetailView):
    model = TransactsModel

    context_object_name = 'tr'
    template_name = 'about_transact.html'

    def get_context_data(self, **kwargs):
        context = super(TransactDetail, self).get_context_data(**kwargs)
        context['isAuth'] = auth.get_user(self.request).username
        context['level_2'] = True
        return context

    def get_object(self):
        object = super(TransactDetail, self).get_object()
        return object
```

Результат



```
C:\Users\HP\AppData\Local\Programs\Python\Python35-32\python.exe K:
1000 2017-10-29 21:13:26+03:00
5999 2017-10-30 01:03:37.450398+03:00
900 2017-11-05 19:20:50.740198+03:00
100 2017-11-05 19:42:32.865723+03:00
9000 2017-10-30 01:05:21.140044+03:00
1000 2017-11-06 01:05:45.133898+03:00

Process finished with exit code 0
```