

Утверждаю: \_\_\_\_\_  
" \_\_\_\_ " \_\_\_\_\_ 2016 г.

Согласовано: \_\_\_\_\_  
" \_\_\_\_ " \_\_\_\_\_ 2016 г.

**«Введение в Python»**

\_\_\_\_\_  
Отчет по лабораторной работе №4

(вид документа)

\_\_\_\_\_  
писчая бумага формата А4

(вид носителя)

\_\_\_\_\_  
(количество листов)

Исполнитель: студент группы РТ5-51

\_\_\_\_\_  
Сергеев Н.Д.

" \_\_\_\_ " \_\_\_\_\_ 2016 г.

1)Скрипт с подключением к БД и несколькими запросами.

```
import MySQLdb

db=MySQLdb.connect(
    host='localhost',
    user='root',
    password='17021942',
    db='lab'
)

cur=db.cursor()

cur.execute('INSERT INTO table1 VALUES (id,"stroka")')

cur.commit()

cur.execute('SELECT * FROM table1;')

entries=cur.fetchall()

for e in entries:
    print (e)

cur.close()
db.close()
```

Результат выполнения

(5L, 'ura')  
(6L, 'rabotaet')  
(8L, 'str')  
(12L, 'sdds')  
(14L, 'stroka')  
(15L, 'stroka')  
(16L, 'str42')  
(17L, 'stroka')

2)Набор классов вашей предметной области с привязкой к СУБД (класс должен уметь хотя бы получать нужные записи из БД и преобразовывать их в объекты этого класса)

```
import MySQLdb
```

```
class Connection:
```

```
    def __init__(self, user, password, db, host='localhost'):
        self.user=user
        self.password=password
        self.host=host
        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=MySQLdb.connect(
            host=self.host,
            user=self.user,
            password=self.password,
            db=self.db
        )
def disconnect(self):
    if self._connection:
        self._connection.close()

```

class Lane:

```

def __init__(self,db_connection,lan):
    self.db_connection=db_connection.connection
    self.lan=lan

def sel(self):
    c=self.db_connection.cursor()
    c.execute('SELECT * FROM table1 WHERE id=(%s);',[int(self.param)])
    entries=c.fetchall()
    self.text=entries[1]
    c.close()

def save(self):
    c=self.db_connection.cursor()
    c.execute('INSERT INTO table1(lane) VALUES (%s);',[self.lan])
    self.param=c.lastrowid

    self.db_connection.commit()
    c.close()

```

```
con=Connection('root','17021942','lab')
```

with con:

```

    lane=Lane(con,'Tdsffkdslnfdsl')
    lane.save()
    lane.sel()
    print("id: {}, stroka: {}".format(lane.param, lane.text))

```

Результат:

```

asword@ASPC:~/lab6$ python cod2.py
id: 33, stroka: Tdsffkdslnfdsl
asword@ASPC:~/lab6$

```

3)Модели вашей  
предметной области

```

from __future__ import unicode_literals

from django.db import models

# Create your models here.
class Book(models.Model):
    name=models.CharField(max_length=100, default='1')

    def __str__(self):
        return self.name

```

4)View для отображения списка ваших сущностей

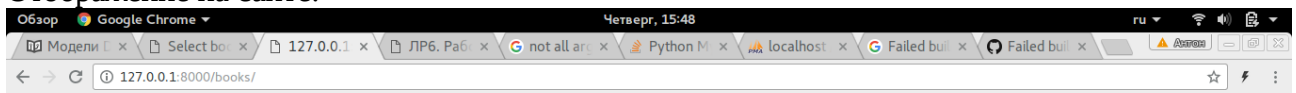
```
class BookListView(View):
```

```

    def get(self, *args, **kwargs):
        books = Book.objects.all()
        return render(self.request,'lab/main.html',{'lane':books})

```

Отображение на сайте:



**d**

Книга2: Название: 1

Книга3: Название: 1