

Утверждаю: _____

Согласовано: _____

"__" ____ 2016 г.

"__" ____ 2016 г.

«Введение в python»

Отчет по лабораторной работе №6

(вид документа)

писчая бумага формата А4

(вид носителя)

(количество листов)

Исполнитель:

студент группы РТ5-51

_____ Попков В.Е.

"__" ____ 2016 г.

Москва – 2016

Задания лабораторной:

1. Скрипт с подключением к БД и несколькими запросами.
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)
3. Модели вашей предметной области
4. View для отображения списка ваших сущностей

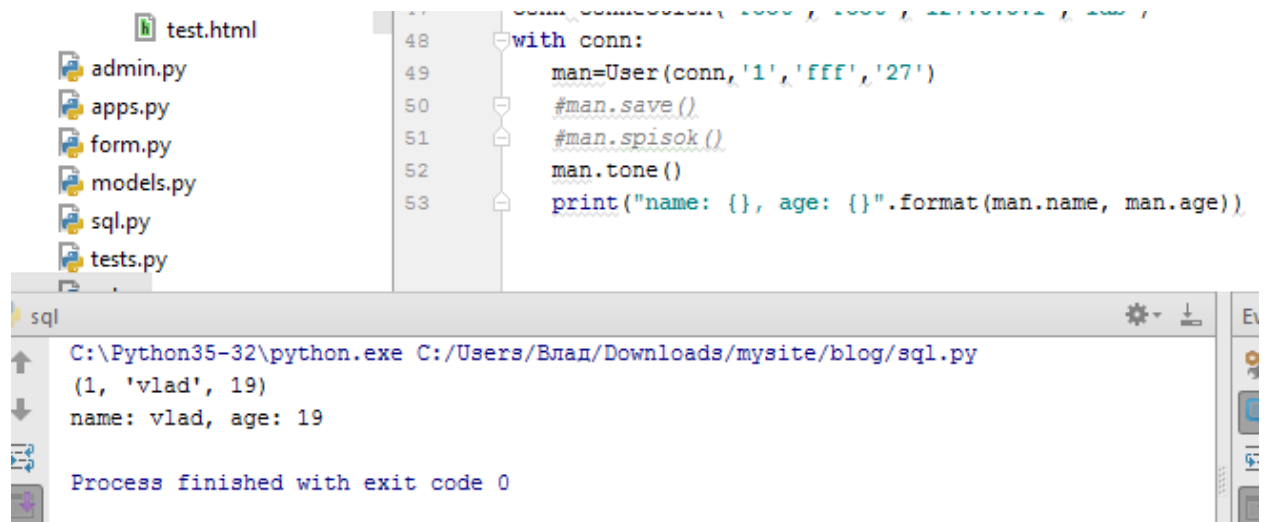
1-2)Класс Connection - для соединения с базой данных

```
from mysql import connector
class Connection():
    def __init__(self, user, password, host, database):
        self.user = user
        self.password = password
        self.host = host
        self.database = database
        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 = connector.connect(user=self.user, password=self.password, host=self.host, database=self.database)
    def disconnect(self):
        if self._connection:
            self._connection.close()
```

Класс User - функции вывода списка пользователей(spisok), создания юзера (save) и отдельного пользователя по айди(tone)

```
class User:
    def __init__(self, db_connection, id, name, age):
        self.db_connection = db_connection.connection
        self.name = name
        self.id = id
        self.age = age
    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO users (id, name, age) VALUES (%s,%s,%s)", (self.id, self.name, self.age))
        self.db_connection.commit()
        c.close()
    def spisok(self):
        c = self.db_connection.cursor()
        c.execute("SELECT * FROM users")
        takeall = c.fetchall()
        for e in takeall:
            print(e)
    def tone(self):
        c = self.db_connection.cursor()
        c.execute('SELECT * FROM users WHERE id=(%s)', [(self.id)])
        entries = c.fetchone()
        print(entries)
        self.name = entries[1]
        self.age = entries[2]
        c.close()
```

Пример работы (сама база ниже)



The screenshot shows an IDE interface. The file explorer on the left lists files: test.html, admin.py, apps.py, form.py, models.py, sql.py, and tests.py. The code editor in the center shows a Python script with the following code:

```
48 with conn:
49     man=User(conn,'1','fff','27')
50     #man.save()
51     #man.spisok()
52     man.tone()
53     print("name: {}, age: {}".format(man.name, man.age))
```

The terminal window at the bottom shows the command: `C:\Python35-32\python.exe C:/Users/Влад/Downloads/mysite/blog/sql.py` and the output: `(1, 'vlad', 19)` and `name: vlad, age: 19`. It also indicates "Process finished with exit code 0".

3)Post - модель предметной/ Man - используемая в лабе

```
from django.db import models
from django.utils import timezone

class Post(models.Model):
    author = models.ForeignKey('auth.User')
    title = models.CharField(max_length=200)
    text = models.TextField()
    created_date = models.DateTimeField(
        default=timezone.now)
    published_date = models.DateTimeField(
        blank=True, null=True)

    def publish(self):
        self.published_date = timezone.now()
        self.save()

    def __str__(self):
        return self.title# Create your models here.

class Man(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=200)
    age = models.IntegerField()
```

4) Class-based view для отображения списка сущностей (MyView(View))

```
content=[...]  
from mysql import connector  
class conn():  
    def __init__(self):  
        self.connection = connector.connect(user='root', password='root', host='127.0.0.1', database='ldb')  
    def spisok_id(self):  
        c = self.connection.cursor()  
        c.execute("SELECT id from users;")  
        return (c.fetchall())  
    def spisok_all(self):  
        c = self.connection.cursor()  
        c.execute("SELECT * from users;")  
        return (c.fetchall())  
    def age(self, id):  
        self.id=int(id)  
        c = self.connection.cursor()  
        c.execute("SELECT name, age from users WHERE id = %s"% self.id)  
        return (c.fetchall())  
class add:...  
def books(request):...  
def govorit(request, pk):...  
class MyView(View):  
    def get(self, request):  
        us = conn()  
        ta = us.spisok_all()  
        take = []  
        for i in ta:  
            take.append(i)  
        return render_to_response('blog/test.html', {'current': take})
```

Platform a

База данных в MySQL

