

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Отчет по рубежному контролю №2  
Вариант В-21

Выполнил:  
студент группы ИУ5-53Б  
Сергеев М.Ю.

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Москва, 2021 г.

### Описание предметной области:

Существуют две сущности: Оператор и Язык программирования.

Оператор имеет следующие атрибуты: номер (id), имя (name), частота употребления (frequency) и номер связанного языка программирования одно-ко-многим (lan\_id).

Язык программирования имеет следующие атрибуты: номер (id), название (name) и используемость(usability). Задание сущностей и список их значений в Python:

### Задание моделей Django ORM

```
class Operator(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=30, blank=True, null=True)
    frequency = models.IntegerField(blank=True, null=True)
    lan_id = models.ForeignKey('Language', models.DO_NOTHING, db_column='lan_id', blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'operator'

class Language(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=30, blank=True, null=True)
    usability = models.IntegerField(blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'language'
```

Для возможности добавления и удаления данных был использован rest\_framework.

Сериализаторы:

```
from rest_framework import serializers
import RK2.models as models

class LanguageSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.Language
        fields = ["id", "name"]

class OperatorSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.Operator
        fields = ["id", "name", "frequency", "lan_id"]
```

Представления:

```

class LanguageViewSet(viewsets.ModelViewSet):
    queryset = models.Language.objects.all()
    serializer_class = serializers.LanguageSerializer

class OperatorViewSet(viewsets.ModelViewSet):
    queryset = models.Operator.objects.all()
    serializer_class = serializers.OperatorSerializer

def getOperators(Language: models.Language):
    result = []
    for op in models.Operator.objects.all():
        if op.lan_id == Language:
            result.append(op)

```

## Примеры выполнения запросов

GET http://127.0.0.1:8000/Operator/
Send
200 OK
27.3 ms
277 B

JSON Auth Query Header 1 Docs
Preview Header 9 Cook
1 ...
1 [
2 {
3 "id": 1,
4 "name": "=",
5 "frequency": 10,
6 "lan\_id": 5
7 },
8 {
9 "id": 2,
10 "name": "!=",
11 "frequency": 5,
12 "lan\_id": 2
13 },
14 {
15 "id": 3,
16 "name": "/=",
17 "frequency": 5,
18 "lan\_id": 4
19 },
20 {
21 "id": 4,
22 "name": "+=",
23 "frequency": 8,
24 "lan\_id": 5
25 },
26 {
27 "id": 5,
28 "name": "++",
29 "frequency": 7,
30 "lan\_id": 6
31 },
32 ]

GET http://127.0.0.1:8000/Language/
Send
200 OK
4.89 ms
143 B

JSON Auth Query Header 1 Docs
Preview Header 9 Co
1 ...
1 [
2 {
3 "id": 1,
4 "name": "JavaScript"
5 },
6 {
7 "id": 2,
8 "name": "Java"
9 },
10 {
11 "id": 3,
12 "name": "C#"
13 },
14 {
15 "id": 4,
16 "name": "Python"
17 },
18 {
19 "id": 5,
20 "name": "PHP"
21 },
22 {
23 "id": 6,
24 "name": "C++"
25 }
26 ]

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Master</title>
  <style>
    table.collapse{
      border: 2px solid black;
      border-collapse: collapse;
      margin-top: 10px;
    }
    td.upper{
      border: 2px solid black;
      padding: 10px;
    }
    td{
      border: 2px solid black;
      padding: 5px;
    }
  </style>
</head>
<body>
{% for value in values %}
  <table class="collapse">
    <tr>
      <td class="upper" colspan="2">ID языка</td>
      <td class="upper" colspan="2">Название языка</td>
    </tr>
    <tr>
      <td class="upper" colspan="2">{{ value.0.id }}</td>
      <td class="upper" colspan="2">{{ value.0.name }}</td>
    </tr>
    <tr>
      <td>ID оператора</td>
      <td>Название</td>
      <td>Частота использования</td>
    </tr>
    {% for op in value.1 %}
      <tr>
        <td>{{ op.id }}</td>
        <td>{{ op.name }}</td>
        <td>{{ op.frequency }}</td>
      </tr>
    {% endfor %}
  </table>
{% endfor %}

```

```

        </tr>
    {% endfor %}
</table>
{% endfor %}
</body>
</html>

```

ID языка		Название языка
1		JavaScript
ID оператора	Название	Частота использования

ID языка		Название языка
2		Java
ID оператора	Название	Частота использования
2	*=	5.4

ID языка		Название языка
3		C#
ID оператора	Название	Частота использования
6	--	3.6

ID языка		Название языка
4		Python
ID оператора	Название	Частота использования
3	/=	5.1