

Отчет по РК № 2 по курсу
"Разработка Интернет-Приложений"

Выполнил:
Студент группы
ИУ5-51Б
Федоров И.Н.

Москва, МГТУ – 2021

Задание:

Рубежный контроль представляет собой разработку веб-приложения с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

№ варианта	Класс 1	Класс 2
21	Оператор	Язык программирования

"Язык программирования" и "Оператор" связаны соотношением один-ко-многим. Выведите список всех связанных операторов и языков программирования, отсортированный по названию языка программирования, сортировка по операторам произвольная.

"Язык программирования" и "Оператор" связаны соотношением один-ко-многим. Выведите список языков программирования с суммарным числом операндов в операторах в каждом языке программирования, отсортированный по суммарному числу операндов в порядке убывания.

Текст программы:

rk2/urls.py

```
"""rk2 URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.2/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
```

```

1. Add an import:  from other_app.views import Home
2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include
from operators import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name="main"),
    path('operator/', include([
        path('', views.read_operator, name='read_operator'),
        path('create/', views.create_operator, name="create_operator"),
        path('update/<int:operator_id>/', views.update_operator,
name="update_operator"),
        path('delete/<int:operator_id>/', views.delete_operator,
name="delete_operator"),
    ])),
    path('prog_lang/', include([
        path('', views.read_prog_lang, name='read_prog_lang'),
        path('create/', views.create_prog_lang, name="create_prog_lang"),
        path('update/<int:prog_lang_id>/',
            views.update_prog_lang, name="update_prog_lang"),
        path('delete/<int:prog_lang_id>/',
            views.delete_prog_lang, name="delete_prog_lang"),
    ])),
    path('report/', views.report, name="report"),
]

```

books/models.py

```

from django.db import models

class ProgrammingLanguage(models.Model):
    name = models.CharField(max_length=256, verbose_name="Programming language
name")

    def __str__(self):
        return self.name

class Operator(models.Model):
    name = models.CharField(max_length=256, verbose_name="Operator name")

```

```

        num_of_operands = models.PositiveIntegerField(verbose_name="Number of
operands")
        syntax = models.CharField(max_length=256, verbose_name="Operator syntax")
        prog_lang = models.ForeignKey(
            ProgrammingLanguage,
            on_delete=models.SET_DEFAULT,
            null=True,
            default=None,
            related_name="operators"
        )

    def __str__(self):
        return self.name

```

books/views.py

```

from django.db.models.aggregates import Avg, Sum
from django.http.response import HttpResponse
from django.shortcuts import get_list_or_404, get_object_or_404, redirect,
render
from .models import ProgrammingLanguage, Operator
from django.db import models

def index(request):
    return render(request, 'index.html')

# operators

def read_operator(request):
    operators = Operator.objects.all()
    return render(request, 'operator/operator_list.html', {'operators':
operators})

def create_operator(request):
    if request.method == 'GET':
        prog_langs = ProgrammingLanguage.objects.all()
        return render(request, 'operator/create_operator.html', {"prog_langs":
prog_langs})
    else:
        dto = {}
        for key in request.POST:
            if key in Operator.__dict__:
                dto[key] = request.POST[key]

```

```

        dto['prog_lang'] = get_object_or_404(ProgrammingLanguage,
pk=request.POST['prog_lang'])
        new_operator = Operator(**dto)
        new_operator.save()
        return redirect('read_operator')

def update_operator(request, operator_id):
    if request.method == 'GET':
        prog_langs = ProgrammingLanguage.objects.all()
        operator = get_object_or_404(Operator, pk=operator_id)
        return render(request, 'operator/update_operator.html', {"operator":
operator, "prog_langs": prog_langs})
    else:
        operator = get_object_or_404(Operator, pk=operator_id)
        for key in request.POST:
            if key in operator.__dict__ and key != 'prog_lang':
                setattr(operator, key, request.POST[key])
        if 'prog_lang' in request.POST:
            setattr(operator, 'prog_lang', get_object_or_404(
                ProgrammingLanguage, pk=request.POST['prog_lang']))
        operator.save()
        return redirect('read_operator')

def delete_operator(request, operator_id):
    operator = get_object_or_404(Operator, pk=operator_id)
    operator.delete()
    return redirect(request.META.get('HTTP_REFERER'))

# prog_langs

def read_prog_lang(request):
    prog_langs = ProgrammingLanguage.objects.all()
    return render(request, 'prog_lang/prog_lang_list.html', {'prog_langs':
prog_langs})

def create_prog_lang(request):
    if request.method == 'GET':
        return render(request, 'prog_lang/create_prog_lang.html')
    else:
        new_prog_lang = ProgrammingLanguage(name=request.POST['name'])
        new_prog_lang.save()
        return redirect('read_prog_lang')

def update_prog_lang(request, prog_lang_id):

```

```

    if request.method == 'GET':
        prog_lang = get_object_or_404(ProgrammingLanguage, pk=prog_lang_id)
        return render(request, 'prog_lang/update_prog_lang.html',
{"prog_lang": prog_lang})
    else:
        prog_lang = get_object_or_404(ProgrammingLanguage, pk=prog_lang_id)
        for key in request.POST:
            if key in prog_lang.__dict__:
                setattr(prog_lang, key, request.POST[key])
        prog_lang.save()
        return redirect('read_prog_lang')

def delete_prog_lang(request, prog_lang_id):
    prog_lang = get_object_or_404(ProgrammingLanguage, pk=prog_lang_id)
    prog_lang.delete()
    return redirect(request.META.get('HTTP_REFERER'))

# REPORT

def report(request):
    sorted_operators = Operator.objects.all().order_by("prog_lang__name")
    num_of_operators = []
    for prog_lang in ProgrammingLanguage.objects.all():
        num_of_operators.append({"prog_lang": prog_lang, "num_of_operators":
Operator.objects.filter(
            prog_lang=prog_lang.pk).aggregate(Sum('num_of_operands'))['num_of_
operands__sum']})
    return render(request, 'report.html', {"sorted_operators":
sorted_operators, "num_of_operators": sorted(num_of_operators, key=lambda x: -
1*x['num_of_operators'])})

```

templates/inc/base.html

```

{% load static %}

<!doctype html>
<html lang="en" class="h-100">

<head>
    <!-- ===== CHARSET AND VIEWPORT ===== -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <!-- ===== CACHE TURN OFF ===== -->

```

```

<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<!-- ===== COPY RIGHT ===== -->
<meta name="author" content="IU5-51 Pavel Strikhar">
<title>
    IU5-51 Strikhar RK2
</title>

<!-- ===== BOOTSTRAP CORE CSS ===== -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <!-- ===== SHARED STYLES ===== -->
    <link rel="stylesheet" href="{% static 'css/navbar.css' %}">
</head>

<body class="d-flex flex-column h-100">
    <header>
        <!-- Navbar -->
        {% include 'inc/navbar.html' %}
    </header>

    <!-- Begin page content -->
    <main role="main" class="flex-shrink-0">
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <!-- Content -->
                    {% block content %}
                    {% endblock content %}
                </div>
            </div>
        </div>
    </main>

    <footer class="footer mt-auto py-3">
        <!-- Footer -->
    </footer>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
ka7Sk0GlIn4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</html>

```

templates/inc/navbar.html

```
{% load static %}
<nav class="navbar navbar-expand-sm navbar-dark bg-dark p-3">
    <a class="navbar-brand ml-5" href="{% url 'main' %}">Strikhar</a>

    <ul class="navbar-nav ml-auto">
        <li class="nav-item">
            <a href="{% url 'main' %}" class="nav-link">Main Page</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'read_operator' %}" class="nav-link">All
operators</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'create_operator' %}" class="nav-link">Create
operator</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'read_prog_lang' %}" class="nav-link">All
prog_langs</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'create_prog_lang' %}" class="nav-link">Create
prog_lang</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'report' %}" class="nav-link">Report</a>
        </li>
    </ul>
</nav>
```

templates/operator/operator_card.html

```
{% load static %}

<div class="card mb-3">
    <div class="row g-0">
        <div class="col-md-4">
            
        </div>
        <div class="col-md-8">
            <div class="card-body">
                <h5 class="card-title">{{ operator.name }}</h5>
                <p class="card-text">{{ operator.prog_lang.name }}</p>
                <p class="card-text">
                    Оператор.
                </p>
            </div>
        </div>
    </div>
</div>
```



```

        <p class="card-text"><small class="text-muted">Синтаксис: {{
operator.syntax }}</small></p>
        <p class="card-text"><small class="text-muted">Кол-во
операндов: {{ operator.num_of_operands }}</small></p>
        <a href="{% url 'delete_operator' operator.pk %}" class="btn
btn-danger"> Delete </a>
        <a href="{% url 'update_operator' operator.pk %}" class="btn
btn-primary"> Update </a>
    </div>
</div>
</div>
</div>

```

templates/operator/operator_form.html

```

<div class="mb-3">
    <label for="operatorName" class="form-label">operator Name</label>
    <input type="text" name="name" value="{{ operator.name }}"
required="required" class="form-control"
        id="operatorName" />
</div>

<div class="mb-3">
    <label for="operatorSyntax" class="form-label">operator Syntax</label>
    <input type="text" name="syntax" value="{{ operator.syntax }}"
required="required" class="form-control"
        id="operatorSyntax" />
</div>

<div class="mb-3">
    <label for="numOfOperands" class="form-label">numOfOperands</label>
    <input type="number" min="0" name="num_of_operands" value="{{
operator.num_of_operands }}" required="required" class="form-control"
        id="numOfOperands" />
</div>

<div class="mb-3">
    <label for="progLang" class="form-label">Select Programming
Language</label>
    <select name="prog_lang" class="form-select" aria-label="Default select
example" id="progLang">
        {% for prog_lang in prog_langs %}
        <option value="{{ prog_lang.pk }}">{{ prog_lang.name }}</option>
        {% endfor %}
    </select>
</div>

```

templates/operator/operator_list.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">All operators</h1>
</div>
<!-- Page Content -->
<div class="d-flex flex-column">
    {% for operator in operators %} {% include 'operator/operator_card.html'
with operator=operator %} {% endfor %}
</div>
{% endblock content %}
```

templates/operator/create_operator.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">New operator</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'create_operator' %}" method="POST" class="form">
        {% csrf_token %} {% include 'operator/operator_form.html' with
operator=operator %}
        <button type="submit" class="btn btn-primary">Create</button>
    </form>
</div>
{% endblock content %}
```

templates/operator/update_operator.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Update operator</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'update_operator' operator.pk %}" method="POST"
class="form">
        {% csrf_token %} {% include 'operator/operator_form.html' with
operator=operator %}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock content %}
```

templates/prog_lang/prog_lang_card.html

```
{% load static %}

<div class="card m-3" style="width: 18rem">
  
  <div class="card-body">
    <h5 class="card-title">{{ prog_lang.name }}</h5>
    <p class="card-text">
      Это язык программирования и он этому рад.
    </p>
    <a href="{% url 'delete_prog_lang' prog_lang.pk %}" class="btn btn-danger"> Delete </a>
    <a href="{% url 'update_prog_lang' prog_lang.pk %}" class="btn btn-primary"> Update </a>
  </div>
</div>
```

templates/prog_lang/prog_lang_form.html

```
<div class="mb-3">
  <label for="prog_langName" class="form-label">prog_lang Name</label>
  <input
    type="text"
    name="name"
    value="{{ prog_lang.name }}"
    required="required"
    class="form-control"
    id="prog_langName"
  />
</div>
```

templates/prog_lang/prog_lang_list.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
  <h1 class="text-center mt-4">All prog_langs</h1>
</div>
<!-- Page Content -->
<div class="d-flex flex-wrap">
  {% for prog_lang in prog_langs %} {% include
'prog_lang/prog_lang_card.html' with prog_lang=prog_lang %} {% endfor %}
</div>
{% endblock content %}
```

templates/prog_lang/create_prog_lang.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">New prog_lang</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'create_prog_lang' %}" method="POST" class="form">
        {% csrf_token %} {% include 'prog_lang/prog_lang_form.html' with
prog_lang=prog_lang %}
        <button type="submit" class="btn btn-primary">Create</button>
    </form>
</div>
{% endblock content %}
```

templates/prog_lang/update_prog_lang.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Update prog_lang</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'update_prog_lang' prog_lang.pk %}" method="POST"
class="form">
        {% csrf_token %} {% include 'prog_lang/prog_lang_form.html' with
prog_lang=prog_lang %}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock content %}
```

templates/report.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Report Page</h1>
</div>
<!-- Page Content -->
<div>
    <h1>Sorted Languages with Operators</h1>
    {% for operator in sorted_operators %} {% include
'operator/operator_card.html' with operator=operator %} {% endfor %}
</div>
```

```

</div>

<div>
  <h1>Sum of operands in Programming Language</h1>
  <div class="d-flex flex-wrap">
    {% for operands in num_of_operators %}
      <div class="d-flex flex-column align-center" >
        <h3 class="text-center">Number of operands {{
operands.num_of_operators }}</h3>
        {% include 'prog_lang/prog_lang_card.html' with
prog_lang=operands.prog_lang %}
      </div>
    {% endfor %}
  </div>
</div>
{% endblock content %}

```

templates/index.html

```

{% extends 'inc/base.html' %}

{% load static %}

{% block content %}
<div class="container">
  <h1 class="text-center mt-4">Рубежный контроль №2</h1>
</div>
<!-- Page Content -->
<div id="wrapper">

<div class="dust"></div>

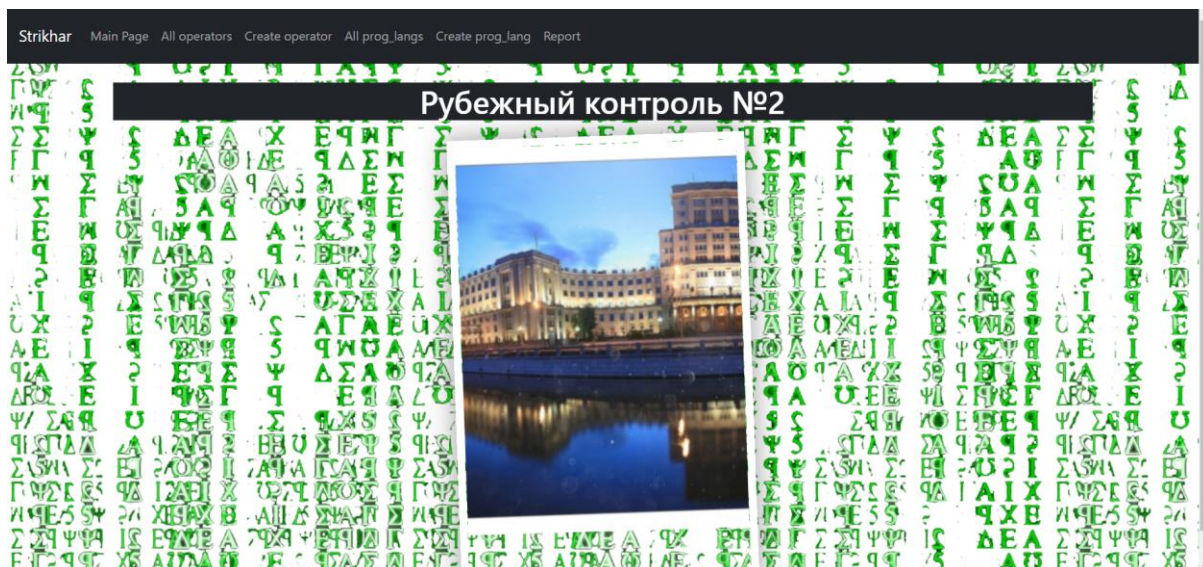
<div class="foreground"></div>

<div class="midground"></div>

<div class="background"></div>
</div><!--wrapper-->
{% endblock content %}

```

Результат работы:



Operators:



Strikhar Main Page All operators Create operator All prog_langs Create prog_lang Report

New operator

operator Name

operator Syntax

numOfOperands

Select Programming Language

superMegaUltraNiceLang

Create

Strikhar Main Page All operators Create operator All prog_langs Create prog_lang Report

Strikhar Main Page All operators Create operator All prog_langs Create prog_lang Report

Update operator

operator Name

Operatorik(menyatel)

operator Syntax

numOfOperands

2

Select Programming Language

superMegaUltraNiceLang

Save

Strikhar Main Page All operators Create operator All prog_langs Create prog_lang Report

Programming languages:

Strikhar Main Page All operators Create operator All prog_langs Create prog_lang Report

All prog_langs	
<div> <div>superMegaUltraNiceLang</div> <div> <div>Это язык программирования и он этому рад.</div> <div>Delete Update</div> </div> </div>	<div> <div>Lingvistik++</div> <div> <div>Это язык программирования и он этому рад.</div> <div>Delete Update</div> </div> </div>

Strikhar Main Page All operators Create operator All prog_langs Create prog_lang Report

Strikhar

Main Page

All operators

Create operator

All prog_langs

Create prog_lang

Report

Update prog_lang

prog_lang Name

superMegaUltraNiceLang

Save

Strikhar

Main Page

All operators

Create operator

All prog_langs

Create prog_lang

Report

New prog_lang

prog_lang Name

Create

localhost:8000(prog_lang)/

Report:

Strikhar

Main Page

All operators

Create operator


All prog_langs

Create prog_lang

Report

Report Page

Sorted Languages with Operators



© CanStockPhoto.com

Uravnyatel

Lingvistik++


Оператор.

Синтаксис: ===

Кол-во операндов: 2

Delete

Update



SUMMIROVANIE

Lingvistik++


Оператор.

Синтаксис: ++++++

Кол-во операндов: 2

Delete

Update



© CanStockPhoto.com

Operatorik(menyatel)


superMegaUltraNiceLang

Оператор.

Синтаксис: ~"

Кол-во операндов: 2

Delete Update



© CanStockPhoto.com

Оператор Второй по званию

superMegaUltraNiceLang


Оператор.

Синтаксис: !!!

Кол-во операндов: 1

Delete Update

Sum of operands in Programming Language

Number of operands 4	Number of operands 3
	
Lingvistik++	superMegaUltraNiceLang
Это язык программирования и он этому рад.	Это язык программирования и он этому рад.
Delete Update	Delete Update