



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ РАДИОТЕХНИЧЕСКИЙ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

## ОТЧЕТ ПО РУБЕЖНОМУ КОНТРОЛЮ №2

Студент Столярова Ольга Денисовна  
*фамилия, имя, отчество*

Группа РТ5-51Б

Название предприятия МГТУ им. Н. Э. Баумана

Студент Столярова О.Д.

Преподаватель Гапанюк Ю.Е.

2021 г.

## Задание:

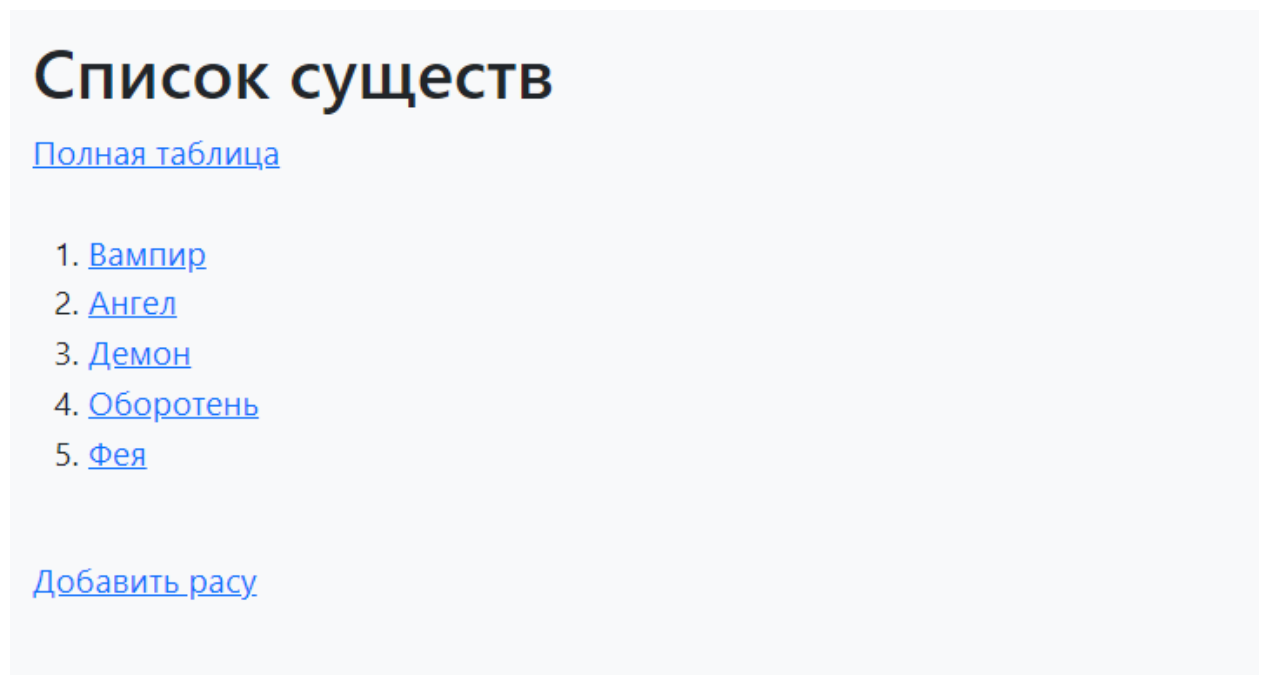
Рубежный контроль представляет собой разработку веб-приложения с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

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.

## Экранные формы с примерами выполнения программы:

Форма для просмотра списка рас:

<http://127.0.0.1:8000/>



Форма для добавления расы:

<http://127.0.0.1:8000/create/>

## Форма для добавления расы

Название расы

Описание

[Добавить](#)

Форма для просмотра отчета со всей информацией из бд:  
<http://127.0.0.1:8000/rep/>

### Список существ

#### 1. Вампир

Это мифические существа, как правило, фольклорная нечисть, нежить, питающаяся кровью людей и животных ради получения из неё питательных веществ, новой силы. Она позволяет вампирам раз свои сверхъестественные способности и исцелять раны.

##### [Способности](#)

- Вампиры могут летать как в своей физической форме, так и в форме летучих мышей, птиц.
- Вампиры могут проникать сквозь любого размера щели, просачиваться сквозь землю.
- Скорость.

#### 2. Ангел

Чистые, непорочные создания. Противоположность демонов.

##### [Способности](#)

- Телепортация.
- Телекинез.
- Способность читать мысли.

#### 3. Демон

Эти личности, как вы прекрасно знаете, являются адскими созданиями. Однако, даже несмотря на то, что они являются таковыми, не у всех кровавый характер. Все они разные.

##### [Способности](#)

- Одержимость.
- Скорость, ловкость и прочие физические характеристики выше человеческих в несколько раз.

#### 4.оборотень

Мифические существа, люди, способные временно менять свой облик магическим путём, превращаясь из человека в животное и наоборот.

##### [Способности](#)

#### 5. Фея

Многие из нас знают, кто такие феи и с чем их едят. Фей много, на любой вкус и цвет. Одна фея владеет только одним видом магии. Фея Пламени никогда не сможет использовать магию Воды.

##### [Способности](#)

Форма для просмотра информации об определенной расе:

<http://127.0.0.1:8000/2/>

## Вампир

**Описание:** Это мифические существа, как правило, фольклорная нечисть, нежить, питающаяся кровью людей и животных ради получения из неё питательных веществ, новой силы. Она позволяет вампирам развивать свои сверхъестественные способности и исцелять раны.

[Способности](#)

[Изменить](#)

[Удалить](#)

Форма для изменения информации о расе:

<http://127.0.0.1:8000/2/update/>

## Форма для редактирования расы

Вампир

Это мифические существа, как правило, фольклорная нечисть, нежить, питающаяся кровью людей и животных ради получения из неё питатель

Сохранить

Форма для просмотра способностей расы (связь один ко многим: одна раса — несколько способностей):

<http://127.0.0.1:8000/2/cans/>

## Список способностей

1. Вампиры могут летать как в своей физической форме, так и в форме летучих мышей, птиц. [Изменить](#) [Удалить](#)
2. Вампиры могут проникать сквозь любого размера щели, просачиваться сквозь землю. [Изменить](#) [Удалить](#)
3. Скорость. [Изменить](#) [Удалить](#)

[Добавить способность](#)

Форма для добавления способностей:

<http://127.0.0.1:8000/2/cans/create/>

## Форма для добавления расы

ID расы

Способность

[Добавить](#)

### Список существ

1. Вампир
2. Ангел
3. Демон
- 4.оборотень
5. Фея

Форма для редактирования способности:

<http://127.0.0.1:8000/update/4/>

## Форма для редактирования способности

ID расы

Способность

Список существ

1. Вампир
2. Ангел
3. Демон
4. Оборотень
5. Фея

## Код программы:

### RK/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('main.urls')),
]
```

### RK/settings.py

```
"""
Django settings for lr5 project.

Generated by 'django-admin startproject' using Django 4.0.

For more information on this file, see
https://docs.djangoproject.com/en/4.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.0/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-
p(z8&hr)=ax2xxz40##*%&3c3=46)v0_sunb#*v=!qp2_=tx5t'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition
```

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'main'
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'lr5.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'lr5.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'creat',
        'USER': 'root',
        'PASSWORD': '20012011',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {

```

```

        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

## Main/urls.py

```

from django.urls import path
from . import views
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('', views.master, name='master'),
    path('<int:cr_id>/', views.detail, name='detail'),
    path('<int:cr_id>/cans/', views.can, name='cann'),
    path('rep/', views.report, name='report'),
    path('create/', views.create, name='create'),
    path('<int:cr_id>/update/', views.update, name='update'),
    path('<int:cr_id>/delete/', views.delete, name='delete'),
    path('<int:cr_id>/cans/create/', views.create1, name='create1'),
    path('update/<int:cr_id>', views.update1),
    path('delete/<int:cr_id>', views.delete1)
]

```

## main/views.py

```

from django.shortcuts import render, redirect, HttpResponseRedirect
from .models import Rases
from .models import Cans
from .forms import RasesForm, CansForm

```

```

from django.http import HttpResponseRedirect, HttpResponseNotFound

def master(request):
    crs = {'crs': Rases.objects.order_by('id')}
    return render(request, 'main/list.html', crs)

def detail(request, cr_id):
    cr = Rases.objects.get(id=cr_id)
    return render(request, 'main/detail.html', {'cr': cr})

def can(request, cr_id):
    cr = Rases.objects.get(id=cr_id)
    c = Cans.objects.all()
    return render(request, 'main/second.html', {'c': c, 'cr': cr})

def report(request):
    cr = Rases.objects.all()
    c = Cans.objects.all()
    return render(request, 'main/rep.html', {'c': c, 'cr': cr})

def create(request):
    error= ''
    if request.method == 'POST':
        form = RasesForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('master')
        else:
            error = 'Данные введены некорректно!'

    form = RasesForm()

    data = {
        'form': form,
        'error': error
    }
    return render(request, 'main/create.html', data)

def update(request, cr_id):
    try:
        cr = Rases.objects.get(id=cr_id)

        if request.method == "POST":
            cr.name = request.POST.get("name")
            cr.discription = request.POST.get("discription")
            cr.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "main/update.html", {"cr": cr})
    except cr.DoesNotExist:
        return HttpResponseNotFound("<h2>Person not found</h2>")

def delete(request, cr_id):
    try:
        cr = Rases.objects.get(id=cr_id)
        cr.delete()
        return HttpResponseRedirect("/")

```



```

except cr.DoesNotExist:
    return HttpResponseRedirect("<h2>Person not found</h2>")

def createl(request, cr_id):
    cr = Rases.objects.all()
    error = ''
    if request.method == 'POST':
        form = CansForm(request.POST)
        if form.is_valid():
            form.save()
        else:
            error = 'Данные введены некорректно!'

    form = CansForm()

    data = {
        'form': form,
        'error': error,
        'cr': cr,
    }
    return render(request, 'main/createl.html', data)

def updatel(request, cr_id):
    try:
        cr = Cans.objects.get(id=cr_id)
        ras = Rases.objects.all()
        if request.method == "POST":
            cr.idrases = request.POST.get("idrases")
            cr.cans = request.POST.get("cans")
            cr.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "main/update2.html", {"cr": cr, "c": ras})
    except cr.DoesNotExist:
        return HttpResponseRedirect("<h2>Person not found</h2>")

def deletel(request, cr_id):
    try:
        cr = Cans.objects.get(id=cr_id)
        cr.delete()
        return HttpResponseRedirect("/")
    except cr.DoesNotExist:
        return HttpResponseRedirect("<h2>Person not found</h2>")

```

## main/models.py

```

from django.db import models

class Rases(models.Model):
    name = models.CharField(max_length=30)
    discription = models.CharField(max_length=255)

    class Meta:
        managed = False
        db_table = 'rases'

class Cans(models.Model):
    idrases = models.IntegerField()
    cans = models.CharField(max_length=100)

```

```

class Meta:
    managed = False
    db_table = 'cans'

```

## main/forms.py

```

from .models import Rases, Cans
from django.forms import ModelForm, TextInput, Textarea

class RasesForm(ModelForm):
    class Meta:
        model = Rases
        fields = ['name', 'discription']

        widgets = {
            "name": TextInput(attrs = {
                'class': 'form-control',
                'placeholder' : 'Название расы'
            }),
            "discription": Textarea(attrs={
                'class': 'form-control',
                'placeholder': 'Описание'
            }),
        }

class CansForm(ModelForm):
    class Meta:
        model = Cans
        fields = ['idrases', 'cans']

        widgets = {
            "idrases": TextInput(attrs = {
                'class': 'form-control',
                'placeholder' : 'ID расы'
            }),
            "cans": Textarea(attrs={
                'class': 'form-control',
                'placeholder': 'Способность'
            }),
        }

```

## base.html

```

<!DOCTYPE html>
<html lang="ru" class="h-100">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock%}</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
">
</head>
<body>
<div class="p-3 mb-2 bg-light text-dark">
    {% block content %}{% endblock %}
</div>
</body>
</html>

```

## List.html

```
{% extends 'main/base.html' %}

{% block title %} Фантастические существа {% endblock %}

{% block styles %}
body {
    background-color: orange;
    color: black;
    font-family: 'Calibri';
}
li {
    font-size: 35px;
    margin-bottom: 30px;
}
li:hover {
    font-weight: bold;
}
a {
    text-decoration: none;
    color: black;
}
{% endblock %}

{% block content %}
<body bgcolor="#ffffe0">

    <big><h1>Список существ</h1>
    <div><a href="{% url 'report' %}">Полная таблица</a></div>
<br>
    <ol>
        {% for i in crs %}
        <li><a href="{% url 'detail' i.id %}">{{i.name}}</a></li>
        {% endfor %}
    </ol>

    <br>
    <div><a href="{% url 'create' %}">Добавить расу</a></div> <br>

    </big>

{% endblock %}
```

## Detail.html

```
<{% extends 'main/base.html' %}

{% block styles %}
body {
    color: black;
    font-family: 'Calibri';
}
li {
    font-size: 30px;
    margin-bottom: 15px;
}
li:hover {
    font-weight: bold;
}
a {
    text-decoration: none;
    color: black;
}
h1 {
```

```

        font-family: 'Times New Roman', Times, serif; /* Гарнитура текста */
        font-size: 500%; /* Размер шрифта в процентах */
    }

{% endblock %}

{% block title %} Пача{% endblock %}

{% block content %}
<body bgcolor="#ffffe0">
    <big><h1>{{ cr.name }}</h1>
    <div><strong>Описание:</strong> {{ cr.discription }}</div>
<br>
    <div><a href="{% url 'cann' cr.id %}">Способности</a></div> <br>
    <div><a href="{% url 'update' cr.id %}"><font color = "red">Изменить
</font></a></div>
    <div><a href="delete"><font color = "red">Удалить</font></a></div>
    </big>
{% endblock %}

```

## Rep.html

```

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

{% block title %} Фантастические существа {% endblock %}

{% block styles %}
body {
    color: black;
    font-family: 'Calibri';
}
li {
    font-size: 30px;
    margin-bottom: 15px;
}
li:hover {
    font-weight: bold;
}
a {
    text-decoration: none;
    color: black;
}
{% endblock %}

{% block content %}
<body bgcolor="#ffffe0">
    <big><h1>Список существ</h1>
    <ol>
        {% for i in cr %}
        <li>
            <dev><strong>{{i.name}}</strong></dev>
            <br>
            <dev>{{i.discription}}</dev>
            <br>
            <dev><u>Способности</u></dev>
            <ul>
                {% for j in c %}
                {% if j.idrases == i.id %}
                <li>{{j.cans}}</li>
                {% endif %}
                {% endfor %}
            </ul>
            <br>

```

```

        </li>
        {% endfor %}
    </ol></big>
{% endblock %}

```

## Create.html

```

<{% extends 'main/base.html' %}

{% block styles %}
body {
    color: black;
    font-family: 'Calibri';
}
li {
    font-size: 30px;
    margin-bottom: 15px;
}
li:hover {
    font-weight: bold;
}
a {
    text-decoration: none;
    color: black;
}
h1 {
    font-family: 'Times New Roman', Times, serif; /* Гарнитура текста */
    font-size: 500%; /* Размер шрифта в процентах */
}

{% endblock %}

{% block title %}Добавление записи{% endblock %}

{% block content %}

<h1>Форма для добавления расы</h1>
<form method = "post">
    {% csrf_token %}
    {{ form.name }}
    <br>
    <span>{{error}}</span>
    {{ form.discription }}
    <button class="btn btn-succes" type="submit"><font color =
"red">Добавить</font></button>
</form>
{% endblock %}

```