СРЕДСТВА ПРОГРАММНОЙ РАЗРАБОТКИ, ВАРИАНТ НА БУКВУ "Е"

**Exercise\_3**

Напишите программу, которая запрашивает у пользователя число, а затем выводит "Четное", если число четное, или "Нечетное", если число нечетное

|  |
| --- |
| # Запрос числа у пользователя  number = int(input("Введите число: "))  # Проверка на четность  if number % 2 == 0:  print("Четное")  else:  print("Нечетное") |

После ввода числа она определит, является ли оно четным или нечетным и выведет соответствующее сообщение.

**Exercise\_6**

Напишите программу, которая создает словарь, заполняет его случайными элементами и сортирует его по значениям.

|  |
| --- |
| import random  import operator  # Создаем словарь и заполняем его случайными элементами  my\_dict = {i: random.randint(1, 100) for i in range(10)}  # Выводим исходный словарь  print("Исходный словарь:")  print(my\_dict)  # Сортируем словарь по значениям  sorted\_dict = dict(sorted(my\_dict.items(), key=operator.itemgetter(1)))  # Выводим отсортированный словарь  print("\nОтсортированный словарь по значениям:")  print(sorted\_dict) |

Программа создает словарь с 10 случайными ключами и значениями от 1 до 100. Затем она сортирует словарь по значениям и выводит как исходный, так и отсортированный словарь.

**Exercise\_9**

Создайте класс "Автомобиль" с атрибутами "марка" и "год выпуска". Создайте объекты, представляющие разные автомобили, и выведите информацию о них

|  |
| --- |
| class Car:  def \_\_init\_\_(self, brand, year):  self.brand = brand  self.year = year  # Создаем объекты, представляющие разные автомобили  car1 = Car("Toyota", 2020)  car2 = Car("Honda", 2018)  car3 = Car("Ford", 2022)  # Выводим информацию о каждом автомобиле  print("Информация о первом автомобиле:")  print(f"Марка: {car1.brand}")  print(f"Год выпуска: {car1.year}")  print("\nИнформация о втором автомобиле:")  print(f"Марка: {car2.brand}")  print(f"Год выпуска: {car2.year}")  print("\nИнформация о третьем автомобиле:")  print(f"Марка: {car3.brand}")  print(f"Год выпуска: {car3.year}") |

Этот код создает класс Car с атрибутами "марка" и "год выпуска". Затем создаются три объекта этого класса, представляющих разные автомобили, и выводится информация о каждом из них.

**Exercise\_15**

Создайте модель «Human» с полями «name», «surname», «date\_birth» и «place\_residence». Определите соответствующие типы полей и их параметры. Создайте миграции и примените их к базе данных.

Для создания модели "Human" с использованием Django и проведения миграций, вам нужно создать Django-приложение и определить модель в файле models.py. Затем вы создадите и примените миграции, чтобы создать соответствующую таблицу в базе данных.

Создайте новый Django-проект, если у вас его еще нет:

|  |
| --- |
| django-admin startproject myproject  cd myproject |

Создайте новое Django-приложение:

|  |
| --- |
| python manage.py startapp myapp |

В файле models.py вашего нового приложения (myapp/models.py), определите модель "Human":

|  |
| --- |
| # models.py  from django.db import models  class Human(models.Model):  name = models.CharField(max\_length=255)  surname = models.CharField(max\_length=255)  date\_birth = models.DateField()  place\_residence = models.CharField(max\_length=255)  def \_\_str\_\_(self):  return f"{self. name} {self. surname} {self. date\_birth} ({self. place\_residence})" |

В файле myproject/settings.py добавьте ваше приложение в INSTALLED\_APPS:

|  |
| --- |
| INSTALLED\_APPS = [  # ...  'myapp',  # ...  ] |

Создайте миграции для вашей модели и примените миграции к базе данных:

|  |
| --- |
| python manage.py makemigrations  python manage.py migrate |

Теперь у вас есть таблица в базе данных, представляющая модель "Human" с полями "name", "surname", "date\_birth" и "place\_residence". Вы можете использовать эту модель для сохранения данных о людях в вашем приложении Django.

**Exercise\_18**

На основе модели "Human" создайте форму для добавления данных о людях в базу данных. Форма должна включать поля для ввода "name", "surname", "date\_birth" и "place\_residence". Реализуйте представление для обработки данных из этой формы и сохранения их в базе данных.

Чтобы создать форму и представление для добавления данных о людях в базу данных, вы можете использовать Django Forms и Views. Вот пример кода:

В файле forms.py вашего приложения (myapp/forms.py), создайте форму на основе модели "Human":

|  |
| --- |
| # forms.py  from django import forms  from .models import Human  class HumanForm(forms.ModelForm):  class Meta:  model = Human  fields = ['name', 'surname', 'date\_birth', 'place\_residence'] |

В файле views.py вашего приложения (myapp/views.py), создайте представление для обработки данных из формы и их сохранения в базе данных:

|  |
| --- |
| # views.py  from django.shortcuts import render, redirect  from .forms import HumanForm  def add\_human(request):  if request.method == 'POST':  form = HumanForm(request.POST)  if form.is\_valid():  form.save()  return redirect('success') # Перенаправление на страницу успеха или другую страницу  else:  form = HumanForm()  return render(request, 'add\_human.html', {'form': form}) |

Создайте HTML-шаблон для отображения формы. В файле templates/add\_human.html:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Add Human</title>  </head>  <body>  <h2>Add Human</h2>  <form method="post" action="{% url 'add\_human' %}">  {% csrf\_token %}  {{ form.as\_p }}  <button type="submit">Submit</button>  </form>  </body>  </html> |

В файле urls.py вашего приложения (myapp/urls.py), определите URL-путь для представления:

|  |
| --- |
| # urls.py  from django.urls import path  from .views import add\_human  urlpatterns = [  path('add\_human/', add\_human, name='add\_human'),  # Добавьте другие URL-пути, если необходимо  ] |

Добавьте URL-пути вашего приложения в основной файл urls.py (myproject/urls.py):

|  |
| --- |
| from django.contrib import admin  from django.urls import path, include  urlpatterns = [  path('admin/', admin.site.urls),  path('', include('myapp.urls')),  ] |

Теперь, когда вы посетите http://localhost:8000/add\_human/, вы увидите форму для добавления данных о человеке, а введенные данные будут сохраняться в базе данных.