Министерство науки и высшего образования Российской Федерации

Тольяттинский государственный университет

Институт математики, физики и информационных технологий

Кафедра «Прикладная математика и информатика»

**А.В. Герасимов, О.М. ГУЩИНА**

**средства программной разработки**

Учебно-методическое пособие

для студентов очной и заочной форм обучения

Тольятти

Издательство ТГУ

2023

УДК 004.2

ББК 32.972

Рецензенты:

д.т.н., профессор кафедры прикладной математики и информатики

Тольяттинского государственного университета С.В. Мкртычев;

к.п.н., доцент кафедры транспорта и технологий нефтегазового комплекса филиала Тюменского индустриального университета

в городе Ноябрьск С.В. Лаптева

Герасимов А.В. Средства программной разработки: учеб.-метод. пособие / А.В. Герасимов, О.М. Гущина. – Тольятти : ТГУ, 2023. – 233 с.

Учебно-методическое пособие содержит теоретические представления в области создания алгоритмов и компьютерных программ, которые можно практически применять в цифровой среде для взаимодействия с обществом и решения цифровых задач в профессиональной деятельности.

Учебно-методическое пособие предназначено для использования в образовательном процессе и может быть полезно студентам как ИТ так и не ИТ-направлений направлений подготовки, профессорско-преподавательскому составу высших учебных заведений, а также любому желающему получить знания в области программирования на языке Python, научиться использовать фреймворки при создании программных продуктов, подключать базу данных, осуществлять сборку, отладку и развертывание веб-приложения, а также настраивать права доступа у пользователей.

Рекомендовано к изданию научно-методическим советом Тольяттинского государственного университета.

© ФГБОУ ВО «Тольяттинский государственный университет», 2023

© А.В. Герасимов, О.М. Гущина, 2023

**Содержание**

[Введение 4](#_Toc141712056)

[Тема 1. «Установка Python 3, установка и настройка редактора кода Visual Studio Code» 5](#_Toc141712057)

[Тема 2. «Переменные, операции с ними, базовый ввод / вывод» 6](#_Toc141712058)

[Тема 3. «Строки, циклы, условия» 9](#_Toc141712059)

[Тема 4. «Коллекции объектов (списки, кортежи, словари, sets)» 12](#_Toc141712060)

[Тема 5. «Работа с файлами» 15](#_Toc141712061)

[Тема 6. «Функции, модули, исключения» 17](#_Toc141712062)

[Тема 7. «Понятие классов и их создание» 20](#_Toc141712063)

[Тема 8. «Объекты и их атрибуты. Создание объектов» 22](#_Toc141712064)

[Тема 9. «Наследование. Использование декораторов» 25](#_Toc141712065)

[Тема 10. «Работа с виртуальным окружением. Установка библиотек» 28](#_Toc141712066)

[Тема 11. «Создание проекта в Django» 30](#_Toc141712067)

[Тема 12. «Создание веб-страницы в Django» 32](#_Toc141712068)

[Тема 13. «Использование шаблонов в Django» 34](#_Toc141712069)

[Тема 14. «Создание моделей в Django» 36](#_Toc141712070)

[Тема 15. «Внесение изменения в БД проекта, используя миграцию» 38](#_Toc141712071)

[Тема 16. «Получение доступа к данным в проекте с использованием Django ORM» 41](#_Toc141712072)

[Тема 17. «Работа с данными, хранящимися в разных таблицах с использованием Django ORM» 43](#_Toc141712073)

[Тема 18. «Работа с формами в Django» 45](#_Toc141712074)

[Тема 19. «Административная панель Django Admin» 48](#_Toc141712075)

[Тема 20. «Регистрация, авторизация пользователей» 50](#_Toc141712076)

[Тема 21. «Сборка и отладка веб-приложения» 52](#_Toc141712077)

[Тема 22. «Развертывание приложения. Подготовка к публикации» 54](#_Toc141712078)

[Заключение 57](#_Toc141712079)

[Библиографический список: 58](#_Toc141712080)

[Глоссарий 59](#_Toc141712081)

# **Введение**

Добро пожаловать в учебно-методическое пособие по дисциплине "Средства программной разработки". Это пособие разработано для помощи в изучении основ программирования на языке Python и создании веб-приложений с использованием фреймворка Django. Здесь Вы найдёте подробные указания по проведению практических занятий, материалы для самостоятельного изучения и примеры заданий.

Python – один из наиболее популярных языков программирования в современном мире. Он используется для разработки веб-приложений, анализа данных, машинного обучения, научных расчётов и многого другого. Django является одним из наиболее популярных фреймворков для веб-разработки на Python. Работа в этой сфере требует глубокого понимания принципов работы этих инструментов, и наши практические работы помогут вам освоить эти принципы.

В этом пособии мы начнём с установки и настройки Python 3 и редактора кода Visual Studio Code. Затем мы постепенно изучим основные концепции и механизмы языка Python, включая работу с переменными, коллекциями объектов, файлами, функциями и классами. Целью будет разработка модульного и структурированного кода, что является ключом к созданию эффективных и надёжных программ.

Далее перейдём к изучению разработки веб-приложений на Django. Вы создадите проект, научитесь создавать веб-страницы, использовать шаблоны, работать с базой данных и формами, а также освоите основы работы с административной панелью Django. Будут рассмотрены вопросы регистрации и авторизации пользователей, сборки и отладки веб-приложения и, наконец, развертывание и подготовка приложения к публикации.

# **Тема 1. «Установка Python 3, установка и настройка редактора кода Visual Studio Code»**

**Учебные вопросы по теме**

1. Какие основные шаги установки Python 3 на разных операционных системах?
2. Какие основные проблемы могут возникнуть при установке Python 3 и как их решить?
3. Как проверить успешность установки Python 3 и его version control?
4. Какие преимущества использования Python 3 по сравнению с другими версиями Python?
5. Как установить и настроить Visual Studio Code для работы с Python 3?
6. Какие основные возможности и функции предлагает Visual Studio Code для разработки на Python?

**Методические материалы к занятию:**

1. Как установить Python 3 на различных операционных системах? Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9
2. Как установить и настроить Visual Studio Code для работы с Python 3? Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Программирование на Python и инструменты разработки». Понимание этих понятий поможет вам лучше понять цель и назначение инструментов, которые вы будете устанавливать.
2. Изучите документацию Python 3 (https://docs.python.org/3/) и Visual Studio Code (https://code.visualstudio.com/docs). Это поможет вам лучше понять, как работают эти инструменты и как их можно настроить для максимальной эффективности.
3. Самостоятельно установите Python 3 на свой компьютер.
4. После установки Python 3, установите Visual Studio Code. Обратите внимание на раздел документации Visual Studio Code, посвященный установке.
5. После установки Visual Studio Code, произведите базовую настройку этого редактора для работы с Python 3. В официальной документации Visual Studio Code есть раздел, посвященный этой теме.
6. Создайте простую программу на Python в Visual Studio Code, чтобы проверить, что всё работает правильно. Это может быть классическая программа "Hello, World!" или что-то более сложное, если вы уже знакомы с Python.

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 2. «Переменные, операции с ними, базовый ввод / вывод»**

**Учебные вопросы по теме**

1. Что такое переменная в Python и какие правила существуют для её названия?
2. Какие типы данных существуют в Python и как можно определить тип переменной?
3. Как происходит присваивание значений переменным в Python? Как можно изменить значение переменной?
4. Что представляют собой операции с переменными и какие операции с переменными вы знаете?
5. Как в Python осуществляется арифметические операции (сложение, вычитание, умножение, деление, возведение в степень, получение остатка от деления) с переменными?
6. Как в Python осуществляется сравнение переменных?
7. Что такое базовый ввод / вывод данных в Python и какие функции используются для этого?
8. Как использовать функцию input() для ввода данных пользователем? Какой тип данных она возвращает?
9. Как использовать функцию print() для вывода данных? Как форматировать вывод?
10. Как в Python осуществляется преобразование типов данных?
11. Какие ошибки могут возникнуть при работе с переменными и как их обрабатывать?

**Методические материалы к занятию:**

1. Что такое переменная в Python? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
2. Какие операции можно выполнять с переменными? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
3. Как работает базовый ввод / вывод данных в Python? Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте существующие понятия категории «Переменные, операции с ними, базовый ввод / вывод в Python». Понимание этих понятий улучшит ваше понимание функций и использования переменных в Python, а также основных механизмов ввода и вывода данных.
2. Изучите документацию Python 3 (https://docs.python.org/3/), с особенным вниманием к разделам, посвященным переменным и операциям с ними, а также вводу и выводу данных. Это поможет вам лучше понять, как эти элементы работают в Python и как их можно использовать для решения различных задач.
3. Самостоятельно напишите небольшие программы на Python 3, используя различные типы переменных и выполняя с ними различные операции.
4. После практической работы с переменными, освойте механизмы базового ввода и вывода данных в Python. Это включает в себя использование функций input() и print().
5. После ознакомления с механизмами ввода и вывода данных, попрактикуйтесь в их использовании в своих программах. Напишите программу, которая запрашивает данные от пользователя, проводит с ними какие-то операции и выводит результат.
6. Создайте несколько программ на Python, которые демонстрируют различные операции с переменными и использование ввода / вывода данных, чтобы убедиться, что вы правильно поняли и смогли применить полученные знания.

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 3. «Строки, циклы, условия»**

**Учебные вопросы по теме**

1. Что такое строка в Python и какие основные операции можно выполнять со строками?
2. Как объявить строку в Python? Какие есть различия в использовании одинарных, двойных и тройных кавычек?
3. Какие особенности имеет операция сложения и умножения строк в Python?
4. Что такое циклы в Python и какие типы циклов представлены в этом языке программирования?
5. Какие сценарии могут быть упрощены с использованием циклов? Можете ли вы привести примеры задач, которые решаются с помощью циклов?
6. Как использовать условные операторы в Python? В чем разница между if, elif и else?
7. Как можно использовать циклы и условные операторы вместе? Приведите примеры.

**Методические материалы к занятию:**

1. Что такое строки в Python и как с ними работать? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
2. Как работают циклы и условные операторы в Python? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
3. Как применять строки, циклы и условные операторы в реальных задачах программирования? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Базовые структуры Python»: строки, циклы и условия. Укажите основные характеристики каждого из этих понятий, их особенности и области применения в программировании на Python.
2. Проанализируйте документ "Python Language Reference". Уделите внимание разделам, касающимся работа со строками, циклами и условными операторами.
3. Проведите исследование кода на Python, в котором активно используются строки, циклы и условные операторы. Отметьте, какие задачи решаются с помощью каждого из этих элементов.
4. Разработайте собственные небольшие программы на Python, в которых будут применяться строки, циклы и условные операторы. Следите за правильностью синтаксиса и структуры кода.
5. В каждой подгруппе выберите наиболее интересные программы, разработанные ее участниками, и проведите их коллективный анализ. Обсудите, какие конструкции были использованы и почему, как можно улучшить или оптимизировать код.
6. Осуществите самостоятельное изучение материалов, указанных в методических рекомендациях. Постарайтесь углубить свои знания о строках, циклах и условных операторах в Python.
7. Напишите программу на Python, которая выполняет обработку и анализ введенных пользователем данных с использованием переменных, строк, циклов и условий на основе представленных вариантов.

|  |  |
| --- | --- |
| **Вариант (первая буква фамилии)** | **Задание** |
| А-Д | Напишите программу, которая запрашивает у пользователя его имя, а затем выводит приветственное сообщение с использованием этого имени |
| Е-М | Напишите программу, которая запрашивает у пользователя число, а затем выводит "Четное", если число четное, или "Нечетное", если число нечетное |
| Н-С | Напишите программу, которая запрашивает у пользователя строку, затем выводит эту строку в обратном порядке, используя цикл |
| Т-Ш | Напишите программу, которая запрашивает у пользователя два числа, затем предлагает пользователю выбрать операцию (сложение, вычитание, умножение или деление) и выводит результат выбранной операции |
| Э-Я | Напишите программу, которая запрашивает у пользователя ввод целого числа, затем выводит простые числа от 2 до введенного числа |

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 4. «Коллекции объектов (списки, кортежи, словари, sets)»**

**Учебные вопросы по теме**

1. Каковы основные различия между списками, кортежами, словарями и множествами в Python? Под какими условиями вы бы выбрали использовать один тип коллекции над другим?
2. Как вы можете управлять памятью при работе с большими коллекциями в Python? Какие методы или техники вы бы использовали?
3. Что такое list comprehension в Python и как он может быть использован для упрощения создания списков? Можете ли вы привести примеры?
4. Какие операции можно выполнить с коллекциями в Python? Какие из этих операций являются наиболее эффективными с точки зрения производительности для каждого типа коллекции?
5. Можете ли вы объяснить концепцию hashing в контексте словарей и множеств в Python? Как это влияет на производительность этих структур данных?
6. Как Python обрабатывает коллекции, которые содержат различные типы данных? Каковы преимущества и недостатки такого подхода?
7. Какие инструменты и модули Python могут быть полезными при работе с коллекциями? Например, вы можете обсудить встроенные функции, такие как len(), max(), min(), sorted(), или модули, такие как collections.

**Методические материалы к занятию:**

1. В чем основные отличия и сходства между списками (lists), кортежами (tuples), словарями (dictionaries) и множествами (sets) в Python? Какие особенности использования этих типов данных следует учитывать при программировании? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
2. Каковы преимущества использования list comprehension в Python и в каких ситуациях этот метод может быть наиболее полезным? Можете ли вы привести пример использования list comprehension? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
3. Какие специальные функции Python вы бы использовали для работы с различными типами коллекций (списки, кортежи, словари, sets) и как они помогают в программировании? Можете ли вы привести примеры их использования? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Коллекции объектов в Python»: списки, кортежи, словари и множества. Укажите основные характеристики каждого из этих понятий, их особенности и области применения в программировании на Python.
2. Проанализируйте документ "Python Language Reference". Уделите внимание разделам, касающимся работы со списками, кортежами, словарями и множествами.
3. Проведите исследование кода на Python, в котором активно используются списки, кортежи, словари и множества. Отметьте, какие задачи решаются с помощью каждого из этих элементов.
4. Разработайте собственные небольшие программы на Python, в которых будут применяться списки, кортежи, словари и множества. Следите за правильностью синтаксиса и структуры кода.
5. Осуществите самостоятельное изучение материалов, указанных в методических рекомендациях. Постарайтесь углубить свои знания о списках, кортежах, словарях и множествах в Python.

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 5. «Работа с файлами»**

**Учебные вопросы по теме**

1. Какие различия между режимами чтения, записи и добавления при работе с файлами в Python, и в каких случаях они применяются?
2. Что такое контекстные менеджеры в Python, и как они используются при работе с файлами? Каковы преимущества их использования?
3. Как в Python обрабатываются исключения при работе с файлами? Какие типы исключений могут возникнуть и как их обрабатывать?
4. Как в Python читать и записывать файлы в формате CSV и JSON? В каких случаях может быть полезно использование этих форматов?
5. Как работать с файлами большого размера в Python, не загружая их полностью в память?
6. Как осуществляется работа с файловой системой в Python (например, создание, удаление, переименование файлов и директорий, обход дерева директорий)?

**Методические материалы к занятию:**

1. Что такое файлы в контексте Python и какие операции мы можем с ними выполнять? Какие особенности работы с файлами следует учитывать при программировании на Python? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
2. Какие модули Python вы бы использовали для работы с файлами и в каких ситуациях каждый из них может быть наиболее полезным? Можете ли вы привести пример использования этих модулей? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
3. Как работать с исключениями при работе с файлами в Python? Какие основные типы исключений могут возникнуть и как их обрабатывать? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Работа с файлами в Python»: открытие, чтение, запись и закрытие файлов. Укажите основные характеристики каждого из этих понятий, их особенности и области применения в программировании на Python.
2. Проанализируйте документ "Python Language Reference". Уделите внимание разделам, касающимся работы с файлами, включая обработку исключений и использование контекстных менеджеров.
3. Проведите исследование кода на Python, в котором активно используется работа с файлами. Отметьте, какие задачи решаются с помощью чтения и записи файлов, и как обрабатываются возникающие исключения.
4. Разработайте собственные небольшие программы на Python, в которых будет осуществляться работа с файлами. Следите за правильностью синтаксиса и структуры кода, уделите внимание корректной обработке возможных исключений.
5. Осуществите самостоятельное изучение материалов, указанных в методических рекомендациях. Постарайтесь углубить свои знания о работе с файлами в Python, включая применение различных форматов файлов, таких как CSV и JSON.

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 6. «Функции, модули, исключения»**

**Учебные вопросы по теме**

1. Каковы основные преимущества использования функций в Python и в каких случаях их стоит применять?
2. Что такое рекурсивные функции в Python и в каких случаях их использование наиболее эффективно?
3. Каковы основные принципы работы с модулями в Python? Как импортировать и использовать функции или переменные из других модулей?
4. Что такое исключения в Python и какова их роль в программировании? Какие типы исключений вы знаете и в каких случаях они могут возникать?
5. Как обрабатываются исключения в Python? Каковы основные стратегии их обработки и в каких случаях стоит применять каждую из них?
6. Что такое собственные исключения и в каких случаях стоит создавать их? Как создать и использовать собственные исключения в Python?
7. Как использование функций, модулей и исключений помогает улучшить читаемость, масштабируемость и отладку кода на Python?

**Методические материалы к занятию:**

1. Что такое функции в контексте Python и какие операции мы можем с ними выполнять? Какие особенности работы с функциями следует учитывать при программировании на Python?иСеверанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
2. Какие модули Python вы бы использовали для определения и использования функций и в каких ситуациях каждый из них может быть наиболее полезным? Можете ли вы привести пример использования этих модулей? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
3. Как работать с исключениями при работе с функциями и модулями в Python? Какие основные типы исключений могут возникнуть и как их обрабатывать? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Функции, модули и исключения в Python»: определение функций, импорт модулей, вызов функций из модулей, обработка исключений. Укажите основные характеристики каждого из этих понятий, их особенности и области применения в программировании на Python;
2. Проанализируйте документ "Python Language Reference". Уделите внимание разделам, касающимся определения функций, импорта модулей, вызова функций из модулей и обработки исключений.
3. Проведите исследование кода на Python, в котором активно используются функции, модули и обрабатываются исключения. Отметьте, какие задачи решаются с помощью этих элементов, и как обрабатываются возникающие исключения.
4. Разработайте собственные небольшие программы на Python, в которых будет осуществляться определение функций, импорт модулей и обработка исключений. Следите за правильностью синтаксиса и структуры кода, уделите внимание корректной обработке возможных исключений.
5. Напишите программу на Python, демонстрирующую работу с основными структурами данных (списки, словари, кортежи, множества) и возможностями языка на основе представленных вариантов.

|  |  |
| --- | --- |
| **Вариант (первая буква фамилии)** | **Задание** |
| А-Д | Напишите программу, которая создает пустой список и заполняет его случайными элементами. Затем программа выводит элементы списка в обратном порядке. |
| Е-М | Напишите программу, которая создает словарь, заполняет его случайными элементами и сортирует его по значениям. |
| Н-С | Напишите программу, которая создает два списка, заполняет их случайными элементами, и создает словарь, используя в качестве значений элементы первого списка, а в качестве ключей элементы из второго. |
| Т-Ш | Напишите программу, которая создает список, заполняет его случайными элементами, и сохраняет этот список в текстовом файле. |
| Э-Я | Напишите функцию, которая принимает три числа в качестве аргументов, представляющих коэффициенты квадратного уравнения (ax^2 + bx + c), и возвращает его корни (решение уравнения). |

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7

2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8

3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 7. «Понятие классов и их создание»**

**Учебные вопросы по теме**

1. Что такое класс в Python и какова его роль в объектно-ориентированном программировании?
2. Как создать класс в Python? Можете ли вы привести примеры определения простого класса и класса с методами и атрибутами?
3. Что такое методы класса и как они отличаются от обычных функций в Python? Можете ли вы объяснить концепцию "self" в методах класса?
4. Каковы принципы работы с атрибутами класса и экземпляра в Python? Как управлять доступом к атрибутам класса?
5. Что такое конструктор класса в Python и какова его роль при создании объектов класса?
6. Какова роль наследования в объектно-ориентированном программировании на Python? Как создать подкласс в Python и переопределить методы базового класса?
7. Как Python поддерживает полиморфизм в объектно-ориентированном программировании? Можете ли вы привести примеры полиморфизма в Python?

**Методические материалы к занятию:**

1. Что такое классы в Python и каков процесс их создания? Какие особенности работы с классами следует учитывать при программировании на Python? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
2. Какие основные составляющие класса в Python и как они используются? Какие основные принципы и правила следует соблюдать при создании классов? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
3. Каковы основные принципы наследования классов в Python? Как реализовывать множественное наследование и каковы возможные подводные камни этого процесса? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Понятие классов и их создание в Python»: создание классов, наследование, инкапсуляция, полиморфизм. Укажите основные характеристики каждого из этих понятий, их особенности и области применения в программировании на Python.
2. Проанализируйте документ "Python Language Reference". Уделите внимание разделам, касающимся создания и использования классов, включая применение конструкторов, методов и свойств классов.
3. Проведите исследование кода на Python, в котором активно используется работа с классами. Отметьте, какие задачи решаются с помощью классов, как реализовано наследование и полиморфизм, и как обрабатываются возникающие исключения.
4. Разработайте собственные небольшие программы на Python, в которых будет осуществляться работа с классами. Следите за правильностью синтаксиса и структуры кода, уделите внимание корректной обработке возможных исключений.

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 8. «Объекты и их атрибуты. Создание объектов»**

**Учебные вопросы по теме**

1. Что такое объект в контексте программирования на Python? Как вы определите его и отличите от других сущностей?
2. Какие основные атрибуты объекта вы можете назвать в Python? Что они представляют собой и какова их роль?
3. Каким образом создаются объекты в Python? Можете ли вы привести примеры?
4. Как работать с атрибутами объекта после его создания? Как можно изменять, добавлять или удалять атрибуты?
5. Какие существуют ограничения или специфические правила при работе с объектами и их атрибутами в Python?
6. В чем разница между атрибутами класса и атрибутами объекта в Python? Как это влияет на работу с ними?
7. Как можно использовать объекты и их атрибуты для решения реальных задач в программировании? Можете ли вы привести примеры из своего опыта или известных вам проектов?

**Методические материалы к занятию:**

1. Что такое объекты и атрибуты в контексте Python и какие операции мы можем с ними выполнять? Какие особенности работы с объектами и их атрибутами следует учитывать при программировании на Python? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
2. Каким образом происходит создание объектов в Python и какие подходы к этому процессу вы бы рекомендовали? В каких ситуациях каждый из них может быть наиболее полезным? Можете ли вы привести пример использования этих подходов? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.
3. Как работать с исключениями при создании объектов и определении их атрибутов в Python? Какие основные типы исключений могут возникнуть и как их обрабатывать? Северанс, Ч. Р. Python для всех / Ч. Р. Северанс ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2022. — 262 с. — ISBN 978-5-93700-104-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241115. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Объекты и их атрибуты в Python»: создание объектов, атрибуты объектов, методы объектов. Укажите основные характеристики каждого из этих понятий, их особенности и области применения в программировании на Python.
2. Проанализируйте документ "Python Language Reference". Уделите внимание разделам, касающимся создания и использования объектов, включая определение и доступ к атрибутам объектов.
3. Проведите исследование кода на Python, в котором активно используется работа с объектами и их атрибутами. Отметьте, какие задачи решаются с помощью объектов и их атрибутов, как реализован доступ и изменение атрибутов объектов, и как обрабатываются возникающие исключения.
4. Разработайте собственные небольшие программы на Python, в которых будет осуществляться работа с объектами и их атрибутами. Следите за правильностью синтаксиса и структуры кода, уделите внимание корректной обработке возможных исключений.

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 9. «Наследование. Использование декораторов»**

**Учебные вопросы по теме**

1. Что такое наследование в объектно-ориентированном программировании и как оно реализуется в Python?
2. Какие типы наследования существуют в Python и в чем заключаются их особенности?
3. Что такое декораторы в Python и какова их основная функция? Какие типы декораторов вы знаете?
4. Как можно использовать декораторы для расширения функциональности методов в классах Python?
5. Как работает механизм наследования в Python, когда применяются декораторы?
6. Какие есть общие принципы и лучшие практики при работе с наследованием и декораторами в Python?

**Методические материалы к занятию:**

1. Что такое наследование и декораторы в контексте Python и какие основные принципы их использования? Какие особенности работы с наследованием и декораторами следует учитывать при программировании на Python? Мартелл, М. А. Python в системном администрировании : руководство для программистов / М. А. Мартелл ; перевод с английского В. С. Макеева. — Москва : ДМК Пресс, 2022. — 440 с. — ISBN 978-5-94074-861-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/130741. — Режим доступа: для авториз. пользователей.
2. Какие основные примеры использования наследования и декораторов вы можете привести в контексте программирования на Python? В каких случаях каждый из них будет наиболее эффективным? Мартелл, М. А. Python в системном администрировании : руководство для программистов / М. А. Мартелл ; перевод с английского В. С. Макеева. — Москва : ДМК Пресс, 2022. — 440 с. — ISBN 978-5-94074-861-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/130741. — Режим доступа: для авториз. пользователей.
3. Как решать возникающие проблемы и обрабатывать возможные исключения при работе с наследованием и декораторами в Python? Какие основные типы исключений могут возникнуть и как их обрабатывать? Мартелл, М. А. Python в системном администрировании : руководство для программистов / М. А. Мартелл ; перевод с английского В. С. Макеева. — Москва : ДМК Пресс, 2022. — 440 с. — ISBN 978-5-94074-861-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/130741. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Наследование и декораторы в Python»: принципы наследования, виды наследования, использование и написание декораторов. Укажите основные характеристики каждого из этих понятий, их особенности и области применения в программировании на Python.
2. Проанализируйте документ "Python Language Reference". Уделите внимание разделам, касающимся наследования классов и использования декораторов.
3. Проведите исследование кода на Python, в котором активно используется наследование и декораторы. Отметьте, какие задачи решаются с помощью этих понятий, как реализована иерархия классов, и как обрабатываются возникающие исключения при использовании декораторов.
4. Разработайте собственные небольшие программы на Python, в которых будет осуществляться использование наследования и декораторов. Следите за правильностью синтаксиса и структуры кода, уделите внимание корректной обработке возможных исключений.
5. Напишите программу на Python, демонстрирующую использование классов, объектов, наследования и декораторов на основе представленных вариантов:

|  |  |
| --- | --- |
| **Вариант (первая буква фамилии)** | **Задание** |
| А-Д | Создайте класс "Студент", который содержит атрибуты "имя" и "возраст". Создайте объекты этого класса, представляющие разных студентов, и выведите информацию о них |
| Е-М | Создайте класс "Автомобиль" с атрибутами "марка" и "год выпуска". Создайте объекты, представляющие разные автомобили, и выведите информацию о них |
| Н-С | Создайте класс "Товар" с атрибутами "название" и "цена". Создайте объекты этого класса, представляющие разные товары, и выведите информацию о них |
| Т-Ш | Создайте базовый класс "Транспортное средство" с методами для вычисления максимальной скорости и вместимости. Затем создайте производные классы, представляющие разные виды транспорта (например, "Автомобиль" и "Самолет"), и реализуйте соответствующие методы для каждого виде транспорта |
| Э-Я | Создайте класс "Планета" с атрибутами "название" и "диаметр". Создайте объекты этого класса, представляющие разные планеты, и выведите информацию о них |

**Рекомендуемая литература:**

1. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
2. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
3. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Тема 10. «Работа с виртуальным окружением. Установка библиотек»**

**Учебные вопросы по теме**

1. Что такое виртуальное окружение в Python и для чего оно нужно?
2. Как создать виртуальное окружение в Python с использованием модуля venv? Опишите процесс пошагово.
3. Какие есть альтернативы стандартному модулю venv для создания виртуального окружения в Python, и в чем их особенности?
4. Как установить библиотеки в виртуальное окружение Python? Что изменится, если установить библиотеки глобально?
5. Какие команды используются для активации и деактивации виртуального окружения в Python в различных операционных системах?
6. Что такое файл requirements.txt и как его использовать для установки библиотек в Python? Как создать этот файл для уже существующего проекта?
7. Какие проблемы и сложности могут возникнуть при работе с виртуальными окружениями и установкой библиотек в Python, и как их решать?

**Методические материалы к занятию:**

1. Что такое виртуальное окружение и установка библиотек в контексте Python и какие основные принципы их использования? Какие особенности работы с виртуальным окружением и установкой библиотек следует учитывать при программировании на Python? Янцев, В. В. Web-программирование на Python / В. В. Янцев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 180 с. — ISBN 978-5-507-46546-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/310289. — Режим доступа: для авториз. пользователей.
2. Какие основные примеры использования виртуального окружения и установки библиотек вы можете привести в контексте программирования на Python? В каких случаях каждый из них будет наиболее эффективным? Янцев, В. В. Web-программирование на Python / В. В. Янцев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 180 с. — ISBN 978-5-507-46546-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/310289. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Работа с виртуальным окружением. Установка библиотек в Python»: виртуальное окружение, установка библиотек, менеджер пакетов pip. Укажите основные характеристики каждого из этих понятий, их особенности и области применения в программировании на Python.
2. Проанализируйте документ "Python Packaging User Guide". Уделите внимание разделам, касающимся создания виртуального окружения и установки библиотек, включая применение менеджера пакетов pip.
3. Проведите исследование кода на Python, в котором активно используется работа с виртуальными окружениями и установка библиотек. Отметьте, какие задачи решаются с помощью виртуальных окружений, как осуществляется установка и обновление библиотек, и как управляется зависимостями между ними.
4. Разработайте собственные небольшие проекты на Python, в которых будет осуществляться работа с виртуальным окружением и установка библиотек. Следите за правильностью создания и использования виртуального окружения, корректной установкой и обновлением библиотек.

**Рекомендуемая литература:**

Янцев, В. В. Web-программирование на Python / В. В. Янцев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 180 с. — ISBN 978-5-507-46546-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/310289. — Режим доступа: для авториз. пользователей.

# **Тема 11. «Создание проекта в Django»**

**Учебные вопросы по теме**

1. Что такое Django и для чего он используется в разработке веб-приложений?
2. Как создать новый проект в Django? Какие команды необходимо использовать?
3. Какова структура проекта Django? Какие основные файлы и папки в нем содержатся и какую роль они играют?
4. Как создать и настроить новое приложение в рамках проекта Django?
5. Какова роль файла settings.py в проекте Django и как его можно настроить для своего приложения?

**Методические материалы к занятию:**

1. Что такое Django и какие основные компоненты этого фреймворка необходимы для создания проекта? Какие особенности работы с Django следует учитывать при создании проекта? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Как создать новый проект в Django? Какие основные шаги необходимы для начала работы над новым проектом и какие инструменты Django используются на каждом из этих шагов? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как решать возникающие проблемы и обрабатывать возможные ошибки при создании проекта в Django? Какие основные типы ошибок могут возникнуть и как их обрабатывать? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Создание проекта в Django»: работа с Django, создание и настройка проекта.
2. Проанализируйте документацию Django. Уделите внимание разделам, касающимся создания и настройки проекта.
3. Проведите исследование проекта на Django, в котором активно используются вышеуказанные элементы. Отметьте, какие задачи решаются с помощью этих элементов, как реализована работа с базой данных, как обрабатываются пользовательские запросы и как представляется пользовательский интерфейс.
4. Разработайте собственный небольшой проект на Django, в котором будет осуществляться работа с основными элементами фреймворка. Следите за правильностью структуры проекта и кода, уделите внимание корректной настройке и обработке возможных ошибок.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 12. «Создание веб-страницы в Django»**

**Учебные вопросы по теме**

1. Что такое HttpResponse в Django и как его использовать для отправки HTTP ответов?
2. Какие альтернативы HttpResponse доступны в Django для отправки HTTP ответов, и в каких случаях их использовать?
3. Как использовать HttpResponse для отправки различных типов контента, например, JSON или HTML?
4. Как можно настроить статус коды и заголовки HTTP с использованием HttpResponse в Django?

**Методические материалы к занятию:**

1. Что такое Django и какие основные компоненты этого фреймворка необходимы для создания веб-страницы? Какие особенности работы с Django следует учитывать при создании веб-страницы? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Как создать новую веб-страницу в Django? Какие основные шаги необходимы для начала работы над новой веб-страницей и какие инструменты Django используются на каждом из этих шагов? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Осуществите анализ существующих понятий категории «Создание веб-страницы в Django»: работа с Django, создание веб-страницы, настройка веб-страницы.
2. Проанализируйте документацию Django. Уделите внимание разделам, касающимся создания и настройки веб-страницы, использования базовых компонентов фреймворка, таких как модели, представления и шаблоны.
3. Проведите исследование проекта на Django, в котором активно используются вышеуказанные элементы для создания веб-страницы. Отметьте, какие задачи решаются с помощью этих элементов, как реализована работа с базой данных, как обрабатываются пользовательские запросы и как представляется пользовательский интерфейс.
4. Разработайте собственную веб-страницу на Django, в которой будет осуществляться работа с основными элементами фреймворка. Следите за правильностью структуры проекта и кода, уделите внимание корректной настройке и обработке возможных ошибок.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 13. «Использование шаблонов в Django»**

**Учебные вопросы по теме**

1. Что такое система шаблонов Django и какова её роль в процессе создания веб-страницы?
2. Как создать и использовать базовый шаблон в Django?
3. Какие синтаксические структуры используются в шаблонизаторе Django и каково их назначение?
4. Как передать данные из представления (view) в шаблон и как их использовать внутри шаблона?
5. Как работает наследование шаблонов в Django и как можно его использовать для повторного использования кода?
6. Какие встроенные теги и фильтры доступны в системе шаблонов Django и как их использовать для манипуляций с данными в шаблоне?
7. Как создать собственные теги и фильтры шаблонов в Django?

**Методические материалы к занятию:**

1. Что такое система шаблонов Django и как она взаимодействует с другими компонентами фреймворка? Какие особенности работы со шаблонами в Django следует учитывать при создании веб-страницы? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Как использовать шаблоны при создании веб-страницы в Django? Какие основные шаги необходимы для работы со шаблонами и какие инструменты Django используются на каждом из этих шагов? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «использование шаблонов в Django»: что такое система шаблонов Django, какие элементы шаблона и какие особенности их использования при создании веб-страниц.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с системой шаблонов. Попробуйте понять, каким образом шаблоны влияют на процесс создания веб-страниц, как они связаны с другими компонентами Django.
3. Проведите исследование существующего проекта на Django, в котором активно используется система шаблонов. Выделите, какие задачи решаются с помощью шаблонов, как они взаимодействуют с другими элементами приложения, в чем их роль при формировании пользовательского интерфейса.
4. Разработайте свою веб-страницу на Django, активно используя систему шаблонов. Следите за корректностью организации шаблонов, правильностью их настройки и применения. Уделите внимание обработке возможных ошибок, связанных с работой шаблонов.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 14. «Создание моделей в Django»**

**Учебные вопросы по теме**

1. Что такое модель в контексте фреймворка Django и какую роль она играет в структуре веб-приложения?
2. Какие типы данных поддерживаются в моделях Django и как их использовать при определении полей модели?
3. Что такое связи между моделями в Django (One-to-One, One-to-Many, Many-to-Many)?

**Методические материалы к занятию:**

1. Что такое модели в Django и как они взаимодействуют с другими компонентами фреймворка? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности работы с моделями в Django следует учитывать при создании и обработке данных? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как создать модель в Django? Какие основные шаги необходимы для работы с моделями и какие инструменты Django используются на каждом из этих шагов? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «создание моделей в Django»: что такое модели Django, какие элементы модели и какие особенности их использования при создании веб-приложений.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с созданием и работой с моделями. Попробуйте понять, каким образом модели влияют на процесс создания веб-приложений, как они связаны с другими компонентами Django.
3. Проведите исследование существующего проекта на Django, в котором активно используются модели. Выделите, какие задачи решаются с помощью моделей, как они взаимодействуют с другими элементами приложения, в чем их роль при формировании структуры данных.
4. Разработайте свою модель в Django, активно используя возможности фреймворка. Следите за корректностью определения модели, правильностью ее настройки и применения. Уделите внимание обработке возможных ошибок, связанных с работой моделей.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 15. «Внесение изменения в БД проекта, используя миграцию»**

**Учебные вопросы по теме**

1. Что такое миграция в контексте работы с базой данных в Django?
2. Какие шаги необходимо предпринять для создания новой миграции в Django?
3. Как применить миграцию в Django и какие команды для этого используются?
4. Что происходит, если вы внесете изменения в модель, но не создадите новую миграцию?

**Методические материалы к занятию:**

1. Что такое миграции в Django и как они взаимодействуют с другими компонентами фреймворка? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности работы с миграциями в Django следует учитывать при внесении изменений в БД? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как создать и применить миграцию в Django? Какие основные шаги необходимы для работы с миграциями и какие инструменты Django используются на каждом из этих шагов? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «миграции в Django»: что такое миграции, как они используются для внесения изменений в структуру БД, какие основные команды и процедуры связаны с миграциями.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с миграциями. Попробуйте понять, как миграции взаимодействуют с моделями и другими элементами фреймворка.
3. Проведите исследование существующего проекта на Django, в котором активно используются миграции. Выделите, какие задачи решаются с помощью миграций, как они помогают обновлять структуру БД и как они взаимодействуют с другими компонентами приложения.
4. Самостоятельно внесите изменения в структуру БД вашего проекта на Django, используя миграции. Следите за тем, чтобы правильно создать и применить миграции, а также учтите возможные ошибки и трудности, которые могут возникнуть в этом процессе.
5. На основе представленного варианта в таблице, соответствующего первой букве вашей фамилии, создайте модель данных в веб-приложении Django и примените изменения в базу данных.

|  |  |
| --- | --- |
| **Вариант (первая буква фамилии)** | **Задание** |
| А-Д | Создайте модель «Article» для веб-приложения. Модель должна содержать поля, такие как «title», «content», «date\_published» и «author». Затем создайте миграции для этой модели и примените их к базе данных. |
| Е-М | 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. |
| Н-С | Создайте модель «Auto». Модель должна содержать поля, такие как «brand», «model», «year» и «color». Определите соответствующие типы полей и их параметры. Затем создайте миграции для этой модели и примените их к базе данных. |
| Т-Ш | Создайте модель «Product». Модель должна содержать следующие поля: «name», «price», «description» и «quantity». Определите соответствующие типы полей и их параметры. Затем создайте миграции для этой модели и примените их к базе данных. |
| Э-Я | Создайте модель «Task». Модель должна содержать следующие поля: «title», «description», «due\_date». Определите соответствующие типы полей и их параметры. Затем создайте миграции для этой модели и примените их к базе данных. |

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 16. «Получение доступа к данным в проекте с использованием Django ORM»**

**Учебные вопросы по теме**

1. Что такое ORM и какова его роль в фреймворке Django?
2. Какие основные преимущества использования Django ORM по сравнению с написанием SQL запросов вручную?
3. Какие виды операций над данными можно выполнить с помощью Django ORM?
4. Как получить доступ к данным определенной модели с использованием Django ORM

**Методические материалы к занятию:**

1. Что такое Django ORM и как он взаимодействует с другими компонентами фреймворка? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности работы с Django ORM следует учитывать при получении и обработке данных? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как использовать Django ORM для получения данных из БД? Какие основные шаги необходимы для работы с Django ORM и какие инструменты Django используются на каждом из этих шагов? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «работа с Django ORM»: что такое ORM, как Django ORM предоставляет доступ к данным в БД, какие основные методы используются для получения данных.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с Django ORM. Попробуйте понять, как ORM взаимодействует с моделями и другими элементами фреймворка.
3. Проведите исследование существующего проекта на Django, в котором активно используется Django ORM. Выделите, какие задачи решаются с его помощью, какие методы используются для получения данных, и как они взаимодействуют с другими компонентами приложения.
4. Самостоятельно реализуйте доступ к данным в вашем проекте на Django, используя Django ORM. Следите за тем, чтобы корректно использовать методы ORM, а также учтите возможные ошибки и трудности, которые могут возникнуть в этом процессе. Обращайте внимание на оптимизацию запросов и безопасность доступа к данным.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 17. «Работа с данными, хранящимися в разных таблицах с использованием Django ORM»**

**Учебные вопросы по теме**

1. Какие типы связей между моделями (таблицами) существуют в Django?
2. Как осуществляется выборка данных из связанных таблиц с использованием Django ORM?
3. Как можно создавать и обновлять связанные записи в Django ORM?
4. Как обрабатываются многие ко многим (ManyToMany) отношения в Django ORM?
5. Какие возможности предоставляет Django ORM для работы с "обратными" связями?

**Методические материалы к занятию:**

1. Что такое Django ORM и как он взаимодействует с данными из разных таблиц в базе данных? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности работы с Django ORM следует учитывать при извлечении и обработке данных из разных таблиц? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как использовать Django ORM для извлечения, обновления и фильтрации данных из разных таблиц в базе данных? Какие основные шаги и инструменты Django используются при работе с Django ORM? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «работа с Django ORM в разных таблицах»: как Django ORM предоставляет доступ к данным в разных таблицах, какие основные методы используются для получения данных.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с Django ORM. Попробуйте понять, как ORM взаимодействует с моделями и другими элементами фреймворка.
3. Проведите исследование существующего проекта на Django, в котором активно используется Django ORM. Выделите, какие задачи решаются с его помощью, какие методы используются для получения данных, и как они взаимодействуют с другими компонентами приложения.
4. Самостоятельно реализуйте доступ к данным из разных таблиц в вашем проекте на Django, используя Django ORM. Следите за тем, чтобы корректно использовать методы ORM, а также учтите возможные ошибки и трудности, которые могут возникнуть в этом процессе. Обращайте внимание на оптимизацию запросов и безопасность доступа к данным

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 18. «Работа с формами в Django»**

**Учебные вопросы по теме**

1. Что такое форма в контексте веб-разработки и для чего она используется?
2. Как Django поддерживает работу с формами?
3. Как создать форму в Django?
4. Какие есть основные типы полей (field classes) для форм в Django?
5. Что такое CSRF-токен и как его использовать при работе с формами в Django?

**Методические материалы к занятию:**

1. Что такое формы в Django и как они используются для обработки ввода пользователя? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности работы с формами в Django следует учитывать при создании и валидации форм? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как использовать формы в Django для обработки и валидации ввода пользователя? Какие основные шаги и инструменты Django используются при работе с формами? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «работа с формами в Django»: как Django обрабатывает формы, какие основные классы и методы используются для создания и валидации форм.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с формами в Django. Попробуйте понять, как формы взаимодействуют с моделями и другими элементами фреймворка.
3. Проведите исследование существующего проекта на Django, в котором активно используются формы. Выделите, какие задачи решаются с их помощью, какие методы используются для создания и валидации форм, и как они взаимодействуют с другими компонентами приложения.
4. Самостоятельно реализуйте обработку форм в вашем проекте на Django, используя встроенные инструменты Django для работы с формами. Следите за тем, чтобы корректно использовать классы и методы для работы с формами, а также учтите возможные ошибки и трудности, которые могут возникнуть в этом процессе. Обращайте внимание на валидацию данных и безопасность обработки ввода пользователя.
5. На основе представленного варианта в таблице, соответствующего первой букве вашей фамилии, создайте форму для добавления данных в базу данных Django-приложения.

|  |  |
| --- | --- |
| **Вариант (первая буква фамилии)** | **Задание** |
| А-Д | На основе модели "Article" создайте форму для добавления данных о статьях в базу данных. Форма должна включать поля для ввода "title", "content", "date\_published" и "author". Реализуйте представление для обработки данных из этой формы и сохранения их в базе данных. |
| Е-М | 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. |
| Н-С | На основе модели "Auto" создайте форму для добавления данных об автомобилях в базу данных. Форма должна включать поля для ввода "brand", "model", "year" и "color". Реализуйте представление для обработки данных из этой формы и сохранения их в базе данных. |
| Т-Ш | На основе модели "Product" создайте форму для добавления данных о продуктах в базу данных. Форма должна включать поля для ввода "name", "price", "description" и "quantity". Реализуйте представление для обработки данных из этой формы и сохранения их в базе данных. |
| Э-Я | На основе модели "Task" создайте форму для добавления данных о задачах в базу данных. Форма должна включать поля для ввода "title", "description" и "due\_date". Реализуйте представление для обработки данных из этой формы и сохранения их в базе данных. |

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 19. «Административная панель Django Admin»**

**Учебные вопросы по теме**

1. Что такое Django Admin и для каких целей он обычно используется?
2. Как настроить Django Admin для использования в вашем проекте?
3. Какие основные функции предоставляет Django Admin?
4. Как добавить модели из вашего приложения в Django Admin?
5. Как настроить интерфейс Django Admin для определённой модели?

**Методические материалы к занятию:**

1. Что такое Django Admin и как оно используется для управления данными приложения? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности работы с Django Admin следует учитывать при настройке и управлении данными? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как использовать Django Admin для управления данными в приложении? Какие основные шаги и инструменты Django используются при работе с Django Admin? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «работа с административной панелью Django Admin»: как Django Admin позволяет управлять данными приложения, какие основные классы и методы используются для настройки административной панели.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с Django Admin. Попробуйте понять, как административная панель взаимодействует с моделями и другими элементами фреймворка.
3. Проведите исследование существующего проекта на Django, в котором активно используется Django Admin. Выделите, какие задачи решаются с его помощью, какие методы используются для настройки и управления данными, и как они взаимодействуют с другими компонентами приложения.
4. Самостоятельно настройте Django Admin для управления данными в вашем проекте на Django, используя встроенные инструменты Django для работы с административной панелью. Следите за тем, чтобы корректно использовать классы и методы для настройки Django Admin, а также учтите возможные ошибки и трудности, которые могут возникнуть в этом процессе. Обращайте внимание на безопасность и эффективность управления данными.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 20. «Регистрация, авторизация пользователей»**

**Учебные вопросы по теме**

1. Что такое системы регистрации и авторизации пользователей в Django?
2. Как настроить в Django процесс регистрации новых пользователей?
3. Как в Django реализуется механизм авторизации пользователей?
4. Какие средства предоставляет Django для управления пользователями (создание, удаление, изменение ролей и привилегий)?
5. Какие шаги необходимо предпринять для внедрения в Django системы регистрации и авторизации пользователей?

**Методические материалы к занятию:**

1. Что такое система регистрации и авторизации в Django и как она используется для управления доступом пользователей к приложению? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности работы с системой регистрации и авторизации Django следует учитывать при создании и управлении пользовательскими аккаунтами? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как использовать систему регистрации и авторизации Django для управления доступом пользователей к различным частям приложения? Какие основные шаги и инструменты Django используются при работе с регистрацией и авторизацией пользователей? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «регистрация и авторизация пользователей в Django»: как Django обрабатывает процесс регистрации и авторизации пользователей, какие основные классы и методы используются для этих процессов.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с регистрацией и авторизацией пользователей. Попробуйте понять, как системы регистрации и авторизации взаимодействуют с моделями пользователей и другими элементами фреймворка.
3. Проведите исследование существующего проекта на Django, в котором активно используются регистрация и авторизация пользователей. Выделите, какие задачи решаются с их помощью, какие методы используются для создания, валидации и управления сессиями пользователей, и как они взаимодействуют с другими компонентами приложения.
4. Самостоятельно реализуйте системы регистрации и авторизации в вашем проекте на Django, используя встроенные инструменты Django для работы с пользователями. Следите за тем, чтобы корректно использовать классы и методы для работы с регистрацией и авторизацией, а также учтите возможные ошибки и трудности, которые могут возникнуть в этом процессе. Обращайте внимание на безопасность и конфиденциальность данных пользователей.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 21. «Сборка и отладка веб-приложения»**

**Учебные вопросы по теме**

1. Что такое процесс сборки в контексте веб-приложения Django?
2. Какие инструменты используются для сборки веб-приложения на Django?
3. Как проводить отладку веб-приложения на Django?
4. Какие существуют способы отладки кода в Django?

**Методические материалы к занятию:**

1. Что такое процесс сборки и отладки в Django и как он используется для обеспечения эффективной работы веб-приложения? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности процесса сборки и отладки в Django следует учитывать при разработке и оптимизации веб-приложения? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как использовать инструменты сборки и отладки Django для обеспечения стабильности и высокой производительности веб-приложения? Какие основные шаги и инструменты Django используются при работе со сборкой и отладкой? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «сборка и отладка веб-приложения Django»: как Django обрабатывает процесс сборки и отладки, какие основные инструменты и методы используются для этих процессов.
2. Изучите документацию Django, особое внимание уделите разделам, связанным со сборкой и отладкой приложений. Попробуйте понять, как процессы сборки и отладки взаимодействуют с различными компонентами фреймворка.
3. Проведите исследование существующего проекта на Django, в котором активно используется процесс сборки и отладки. Выделите, какие задачи решаются с их помощью, какие инструменты и методы используются для сборки и отладки, и как они взаимодействуют с другими компонентами приложения.
4. Самостоятельно реализуйте процесс сборки и отладки в вашем проекте на Django, используя встроенные инструменты Django. Следите за тем, чтобы корректно использовать инструменты и методы для работы со сборкой и отладкой, а также учтите возможные ошибки и трудности, которые могут возникнуть в этом процессе.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Тема 22. «Развертывание приложения. Подготовка к публикации»**

**Учебные вопросы по теме**

1. Что такое развертывание веб-приложения Django?
2. Какие шаги включает подготовка веб-приложения Django к публикации?
3. Какие инструменты можно использовать для развертывания веб-приложения Django?
4. Как настроить проект для развертывания веб-приложения Django?

**Методические материалы к занятию:**

1. Что такое процесс развертывания веб-приложения Django и как он используется для обеспечения его доступности и стабильной работы? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Какие особенности процесса развертывания веб-приложения Django следует учитывать при его подготовке к публикации и оптимизации его работы? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
3. Как использовать инструменты и методики развертывания Django для обеспечения стабильности и высокой доступности веб-приложения? Какие основные шаги и инструменты Django используются при подготовке веб-приложения к публикации? Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

**Методические указания по выполнению самостоятельной работы студентов:**

1. Проанализируйте базовые понятия категории «развертывание веб-приложения Django»: как Django обрабатывает процесс развертывания, какие основные инструменты и методы используются для этого процесса.
2. Изучите документацию Django, особое внимание уделите разделам, связанным с развертыванием приложений. Попробуйте понять, как процесс развертывания взаимодействует с различными компонентами фреймворка.
3. Проведите исследование существующего проекта на Django, в котором активно используется процесс развертывания. Выделите, какие задачи решаются с его помощью, какие инструменты и методы используются для развертывания, и как они взаимодействуют с другими компонентами приложения.
4. Самостоятельно реализуйте процесс развертывания в вашем проекте на Django, используя встроенные инструменты Django. Следите за тем, чтобы корректно использовать инструменты и методы для работы с развертыванием, а также учтите возможные ошибки и трудности, которые могут возникнуть в этом процессе.

**Рекомендуемая литература:**

Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.

# **Заключение**

В этом пособии были раскрыты различные темы, каждая из которых содержала практические занятия, обсуждение теоретических вопросов, методические указания и рекомендации по выполнению самостоятельной работы, а также литературу для глубокого погружения в тему.

В пособии представлены 22 темы, которые включают в себя полный цикл разработки веб-приложений на Python с использованием фреймворка Django: от установки Python и настройки редактора кода, изучения основ работы с переменными, строками, циклами и условиями, а также обзора основных типов коллекций до более продвинутых тем, таких как работа с файлами, использование функций, создание модулей и обработка исключений.

Разделы, посвященные объектно-ориентированному программированию, знакомят с понятиями классов, объектов, их атрибутов и наследования. Также рассматривается использование декораторов.

Большая часть пособия посвящена работе с Django. Темы касаются создании проектов, работы с веб-страницами, использовании шаблонов и создании моделей. Отдельно отражены вопросы миграции и работы с Django ORM.

Помимо этого, было уделено внимание работе с формами, использованию административной панели Django, организации процесса регистрации и авторизации пользователей. Завершается пособие разделами, которые затрагивают сборку и отладку веб-приложения, а также его подготовку и развертывание для публикации в сети.

# **Библиографический список:**

1. Меле, А. Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711. — Режим доступа: для авториз. пользователей.
2. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс. – Москва : ДМК Пресс, 2022. – 262 с. – ISBN 978-5-93700-104-7
3. Matthes, E. Python Crash Course / E. Matthes. – San Francisco : No Starch Press, 2019. – 544 с. – ISBN 978-1-59327-928-8
4. Sweigart, A. Automate the Boring Stuff with Python / A. Sweigart. – San Francisco : No Starch Press, 2019. – 592 с. – ISBN 978-1-59327-992-9

# **Глоссарий**

**Python** – высокоуровневый интерпретируемый язык программирования общего назначения. Он поддерживает несколько парадигм программирования, включая структурное, объектно-ориентированное и функциональное программирование. Python отличается чистым и понятным синтаксисом, что способствует его популярности.

**Функция** – блок организованного, многократно используемого кода, который используется для выполнения конкретного задания. Функции обеспечивают лучшую модульность для вашего приложения и высокую степень повторного использования кода.

**Класс** – шаблон для создания объектов (конкретных данных).

**Объект** – экземпляр класса.

**Модуль** – файл, содержащий функции, определения и выражения Python. Модуль представляет собой способ организации связанных кодов Python.

**Исключение** – событие, происходящее во время выполнения программы, которое нарушает нормальный ход выполнения инструкций программы.

**Django** – фреймворк для разработки веб-приложений на языке Python, который следует принципу "Не повторяйся" (DRY - Don't Repeat Yourself) и использует паттерн проектирования "Модель-Представление-Контроллер" (MVC).

**Проект** – верхнеуровневая организация, представляющая собой полное веб-приложение, которое состоит из одного или нескольких приложений.

**Приложение** – подсистема Django, которая может быть использована в одном или нескольких проектах. Приложение может содержать модели, представления, шаблоны и другие компоненты.

**Модель** – определение структуры базы данных, включая функциональные и операционные данные.

**Представление (View)** – компонент, который обрабатывает запрос и формирует ответ. В Django представление это Python-функция или класс, который принимает HTTP-запрос и возвращает HTTP-ответ.

**Шаблон** – файл, определяющий структуру выходного документа (обычно это HTML).

**URL диспетчер (URLconf)** – механизм, с помощью которого Django направляет HTTP-запросы к соответствующим представлениям.

**Middleware** – серия хуков, которые Django применяет к обработке запросов и ответов.

**ORM (Object-Relational Mapping)** – компонент Django для работы с базами данных. С его помощью вы можете взаимодействовать с базой данных, как с обычными Python-объектами.