

ООО «АКАДЕМИЯ ЛАД»

УТВЕРЖДЕНО  
Директором ООО «Академия Лад»



А.В. Усков

(Приказ №1 от 23 мая 2023 г.)

**ДОПОЛНИТЕЛЬНОЕ ОБРАЗОВАНИЕ**  
**ДОПОЛНИТЕЛЬНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**  
**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА -**  
**ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

«Программирование на Python и Django»

(Наименование программы)

144 часа

Нижний Новгород

2023

## **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**1.1.** Цель программы: совершенствование и получение новой компетенции, профессиональных знаний, которые позволят реализовать себя в сфере backend-разработки на Python и Django.

**1.2.** Нормативные документы для разработки программы повышения квалификации:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

- Приказ Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;

- Профессиональный стандарт 06.001 «Программист» утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022 № 424н.

**1.3.** Категории слушателей на обучение которых рассчитана программа повышения квалификации (далее – Программа): лица, имеющие среднее профессиональное или высшее образование.

**1.4.** Входные требования к обучающимся:

Наличие базовых навыков работы с персональным компьютером.

**1.5.** Сфера применения слушателями полученных профессиональных компетенций, умений и знаний. Знания, полученные в ходе программы могут быть использованы на предприятиях малого и среднего бизнеса, в частной практике при реализации деятельности по backend-разработке.

**1.6.** Программа реализуется исключительно с применением электронного обучения и дистанционных образовательных технологий.

## **2. ХАРАКТЕРИСТИКА ПОДГОТОВКИ ПО ПРОГРАММЕ**

**2.1.** Нормативный срок освоения программы 144 часа.

**2.2.** Срок обучения 20 недель

**2.3.** Общая трудоемкость 4 ЗЕ.

**2.4.** Режим обучения 7,2 часа в неделю.

### **3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ**

Слушатель, освоивший программу, должен:

**3.1.** обладать профессиональными компетенциями, включающими в себя способность:

ПК-1. Техническая поддержка процессов, создание (модификация) и анализ и сопровождение информационных ресурсов.

ПК-2. Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов.

ПК-3. управлять процессами и проектами по созданию (модификации) информационных ресурсов.

**3.2.** знать:

- основы веб-программирования на языке Python;
- основы backend-разработки на Django;
- инструменты backend-разработки веб-приложений;
- методы и приемы отладки программного кода.

**3.3.** уметь:

- решать задачи на Python;
- выявлять ошибки в программном коде;
- разрабатывать backend сайтов на языке python, в том числе с использованием библиотек, инструментов и фреймворков (Django);
- работать в команде.

**3.4.** владеть:

- современными информационными технологиями и навыками работы со специальным программным обеспечением для backend-разработки веб-приложений;
- технологиями backend-разработки веб-приложений с использованием языка Python и фреймворка Django.

**3.5.** Сфера применения слушателями полученных профессиональных компетенций, умений и знаний.

Знания, полученные в ходе программы могут быть использованы на предприятиях малого и среднего бизнеса, в частной практике при реализации деятельности по backend-разработке веб-приложений.

#### **4. ТРЕБОВАНИЯ К СТРУКТУРЕ ПРОГРАММЫ**

Программа предусматривает изучение следующих дисциплин:

Дисциплина 1. «Программирование на Python»

Дисциплина 2. «СУБД MySQL»

Дисциплина 3. «Фреймворк Django»

#### **5. ТРЕБОВАНИЯ К ОЦЕНКЕ КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

«Программирование на Python и Django»

Процедура **промежуточной аттестации** предусматривает выполнение практических заданий по темам дисциплин.

Процедура **итоговой аттестации** предусматривает выполнение итогового проекта в команде. При подготовке итогового проекта слушатель должен продемонстрировать полученные умения и навыки в области backend-разработки веб-приложений. Текущий контроль проверки качества освоения дисциплин программы повышения квалификации осуществляется в форме устного собеседования.

## Формы и методы контроля и оценки результатов освоения дисциплин

| № п/п | Наименование процедуры                        | Основные показатели оценки  | Формы и методы контроля и оценки  |
|-------|---|---|---|
| 1.    | Дисциплина 1.<br>«Программирование на Python» | <i>знает</i> основы программирования на языке Python», методы и приемы отладки программного кода<br><i>умеет</i> решать задачи программирования на Python».<br><i>владеет</i> современными информационными технологиями и навыками программирования на языке Python».   | Текущий контроль - устный опрос.<br>Промежуточная аттестация - выполнение практических заданий. |
| 2.    | Дисциплина 2.<br>«СУБД MySQL»                 | <i>знает</i> основы реляционной модели и SQL<br><i>умеет</i> писать запросы на SQL (выборка данных, сортировка, фильтрация, группировка, агрегатные функции).   | Текущий контроль - устный опрос.<br>Промежуточная аттестация - выполнение практических заданий. |
|       | Дисциплина 3.<br>«Фреймворк Django»           | <i>знает</i> основы backend-разработки веб-приложений на Django, инструменты разработки backend-приложений, методы и приемы отладки программного кода<br><i>умеет</i> разрабатывать backend-приложений на языке Python, в том числе с использованием библиотек, инструментов и фреймворков (Django), работать в команде<br><i>владеет</i> современными информационными технологиями и навыками работы со специальным программным обеспечением для backend-разработки веб-приложений;<br>технологиями разработки веб-приложений с использованием языка Python и фреймворка Django. | Текущий контроль - устный опрос.<br>Промежуточная аттестация - выполнение практических заданий. |
| 3     | Итоговая аттестация                           | Работа в качестве backend-разработчика на Django на стажировке.   | Итоговый проект   |

## **Примеры вопросов для текущего контроля освоения учебного материала:**

1. Что такое переменная в Python, и какой синтаксис используется для их объявления?
2. Какие базовые типы данных поддерживает Python, и какие операции можно выполнять с этими типами данных?
3. Что такое список (list) в Python, и какие методы можно использовать для работы с ним?
4. Какие различия между кортежем (tuple) и списком (list) в Python?
5. Что такое словарь (dictionary) в Python, и как получить доступ к элементам словаря?
6. Какие встроенные функции Python можно использовать для работы со строками?
7. Что такое модуль в Python, и как импортировать функции из другого модуля?
8. Как создать класс и объект в Python, и какие основные принципы ООП реализованы в этом языке?
9. Какие основные принципы наследования и полиморфизма применяются в Python?
10. Что такое исключения (exceptions) в Python, и как обрабатывать ошибки с помощью блока try-except?
11. Как создать файл в Python, записать данные в файл и прочитать данные из файла?
12. Какие библиотеки стандартной библиотеки Python вы использовали для работы с временем и датами?
13. Как реализовать сортировку списка в Python с помощью встроенных функций или методов?

14. Что такое генераторы (generators) в Python, и как они отличаются от списков или кортежей?
15. Какие библиотеки сторонних разработчиков вы использовали в своих проектах на Python, и для чего они были нужны?
16. Какие принципы следует соблюдать при написании чистого и эффективного кода на Python?
17. Что такое Django, и какую роль он играет в веб-разработке на Python?
18. Какие основные компоненты включает структура проекта на Django?
19. Как создать новое Django приложение, и как его зарегистрировать в проекте?
20. Какие шаблонные языки поддерживает Django, и какие возможности они предоставляют для создания динамических веб-страниц?
21. Что такое ORM (Object-Relational Mapping) в контексте Django, и какие преимущества он предоставляет при работе с базой данных?
22. Как настроить маршрутизацию URL-адресов в Django приложении?
23. Что такое СУБД MySQL и для чего она используется?
24. Какие типы представлений (views) существуют в Django, и как они используются для обработки запросов от пользователей?
25. Что такое миграции базы данных в Django, и зачем они необходимы?
26. Какие инструменты Django предоставляет для администрирования сайта и управления данными?
27. Как реализовать аутентификацию пользователей в Django приложении?
28. Какие механизмы безопасности предусмотрены в Django для защиты от атак на сайт?
29. Как организовать тестирование Django приложения с использованием встроенных средств фреймворка?

## Примеры практических заданий для осуществления промежуточной аттестации:

### Дисциплина 1. «Программирование на Python»

1. Напишите программу, которая запрашивает имя и возраст пользователя, и выводит приветствие и возраст пользователя в следующем году.
2. Напишите программу, которая получает на вход целое число  $n$  из диапазона от 1 до 9 включительно, и выводит результат вычисления выражения  $n^{n^2} - n^n - n$ . Например, если введенное число 3, программа должна вывести 297 ( $3^{3^2} - 3^3 - 3$ ).
3. Напишите программу, которая получает от пользователя вещественные числа  $a$  и  $b$  – длину и ширину прямоугольника, – и выводит периметр и площадь прямоугольника.
4. Напишите программу, которая получает строку с именем, отчеством и фамилией, написанными в произвольном регистре, и выводит данные в правильном формате. Например, строка `**алеКСандр СЕРГЕЕВИЧ ПушкиН**` должна быть преобразована в `**Александр Сергеевич Пушкин`.
5. Напишите программу, которая запрашивает у пользователя радиус круга и выводит на экран его площадь.
6. Создайте функцию, которая принимает на вход список чисел и возвращает сумму их квадратов.
7. Напишите программу, которая принимает на вход строку и выводит количество гласных букв в ней.
8. Создайте класс "Треугольник", у которого есть метод для вычисления площади по длинам его сторон.
9. Напишите программу, которая считывает из файла список чисел, сортирует его по возрастанию и записывает результат обратно в файл.

10. Реализуйте функцию, которая проверяет, является ли введенная пользователем строка палиндромом (читается одинаково с начала и с конца).
11. Создайте модуль для работы с геометрическими фигурами, включающий классы для круга, прямоугольника и треугольника с методами для вычисления их характеристик (площадь, периметр и т.д.).
12. Напишите программу, которая генерирует случайный пароль заданной длины для пользователя.
13. Разработайте простую игру "Крестики-нолики" для двух игроков, используя консольный интерфейс.
14. Создайте программу для анализа текста, которая считает количество слов, предложений и символов во введенном тексте.

## **Дисциплина 2. «СУБД MySQL»**

1. Создать базу данных MySQL и несколько таблиц с различными типами данных, включая числовые, строковые и даты.
2. Задание включает написание SQL-запросов для извлечения данных из таблиц, фильтрации, сортировки, группировки данных и использования агрегатных функций.
3. Выполнить запросы с использованием операторов INNER JOIN, LEFT JOIN, RIGHT JOIN и FULL JOIN для объединения данных из нескольких таблиц.
4. Изменение данных в таблицах. Задание заключается в выполнении операций добавления, обновления и удаления данных в таблицах MySQL.
5. Использование подзапросов. Написать SQL-запросы с использованием подзапросов (подзапросы в SELECT, FROM и WHERE) для выполнения сложных операций над данными.

6. Оптимизация запросов. Выполнить оптимизацию SQL-запросов с использованием индексов, объединений и других методов для улучшения производительности.

7. Работа с транзакциями: Выполнить операции COMMIT и ROLLBACK в рамках транзакций для обеспечения целостности данных при выполнении группы операций.

8. Использование хранимых процедур и функций. Задание включает создание и использование хранимых процедур и функций в MySQL для повторного использования логики обработки данных.

9. Работа с представлениями. Создать представления (views) в MySQL для удобного доступа к данным без необходимости написания сложных запросов.

10. Резервное копирование и восстановление данных. Выполнить создание резервных копий базы данных MySQL с помощью mysqldump и восстановление данных из резервных копий для обеспечения безопасности и целостности информации.

### **Дисциплина 3. «Фреймворк Django»**

1. Настройка проекта Django с использованием MySQL. Настроить проект Django для использования базы данных MySQL в качестве основного хранилища данных, включая установку необходимых драйверов, изменение настроек базы данных в файле settings.py и проверку подключения.

2. Создать несколько моделей Django, которые будут отображать таблицы в базе данных MySQL. Установить связи между моделями с использованием ForeignKey, OneToOneField и ManyToManyField. Провести миграции для создания таблиц в базе данных.

3. Выполнить несколько сырых SQL-запросов в контексте Django, используя методы raw() и execute().

4. Оптимизировать запросы к базе данных MySQL, используя инструменты Django ORM. Использовать `select_related` и `prefetch_related` для уменьшения количества запросов, `annotate` и `aggregate` для выполнения группировок и вычислений на стороне базы данных.

5. Создание моделей Django. Создать несколько моделей Django для предметной области, которая может включать в себя пользователей, профили, посты и комментарии.

6. Создание API с использованием Django REST framework. Задание заключается в создании API с помощью Django REST framework для доступа к данным, созданным в предыдущем задании.

7. Аутентификация и авторизация. Реализовать систему аутентификации и авторизации на основе Django, включая использование токенов для аутентификации API.

8. Работа с формами. Задание включает создание форм для ввода данных пользователем и их обработка с помощью Django Form.

9. Использование ORM. Выполнить запросы к базе данных с использованием ORM Django для извлечения, фильтрации и обновления данных.

10. Работа с миграциями. Создать и применить несколько миграций для изменения структуры базы данных, включая добавление новых таблиц, изменение существующих полей и удаление таблиц или полей.

### **Критерии оценки текущих практических заданий по дисциплинам 1-3.**

На основании выполненных практических заданий обучающемуся определяется оценка – «зачтено», «не зачтено».

| <b>Оценка</b> | <b>Уровень подготовки</b>  |
|---------------|--|
| Зачтено       | Хорошая подготовка. Обучающийся выполнил все предложенные задания.     |
| Не зачтено    | Подготовка недостаточная и требует дополнительного изучения материала. |

### **Описание итогового зачетного проекта**

Работа в качестве backend-разработчика на Python и Django на стажировке в ООО «Академия Лад».

### **Критерии оценки итогового проекта**

На основании выполненного итогового проекта обучающемуся определяется оценка – «зачтено», «не зачтено».

| <b>Оценка</b> | <b>Уровень подготовки</b>  |
|---------------|--|
| Зачтено       | Хорошая подготовка. Обучающийся выполнил все предложенные этапы работы над проектом. Проведено исследование пользовательского опыта, представлены его результаты. Сделаны корректные выводы. |
| Не зачтено    | Подготовка недостаточная и требует дополнительного изучения материала.   |

### **Литература**

#### **- Книги**

1. Mark Lutz. "Изучаем Python" (Learning Python).
2. Eric Matthes. "Программирование на Python" (Python Crash Course).
3. Al Sweigart. "Автоматизация рутины с помощью Python" (Automate the Boring Stuff with Python).

#### **- Источники в интернете**

1. LearnPython (<https://learn.python.ru/>) - Сайт с учебными материалами по Python на русском языке, включая уроки для начинающих и продвинутых программистов.

2. DjangoSchool (<https://djangoschool.ru/>) - Ресурс с уроками и статьями о веб-разработке на Django, включая примеры кода и практические советы.
3. PythonRu (<https://pythonru.com/>) - Сайт с обучающими материалами по Python на русском языке, включая статьи, уроки и видеоуроки.
4. WebDev (<https://webdevblog.ru/>) - Блог с статьями о веб-разработке, включая много материалов по Django и Python.
5. Tproger (<https://tproger.ru/tag/python/>) - Ресурс с новостями, статьями и уроками по программированию, включая материалы по Python и Django.
6. Stepik (<https://stepik.org/>) - Образовательная платформа с онлайн-курсами по Python и Django на русском языке.

## **6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ ОБЕСПЕЧЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

*Руководитель программы повышения квалификации:*

Усков А.В. Директор ООО «Академия Лад»

*Разработчики программы повышения квалификации:*

Лапаев А.С. Разработчик, компания Протон

Усков А.В. Директор ООО «Академия Лад»

*Составители учебно-тематического плана программы повышения квалификации:*

Лапаев А.С. Разработчик, компания Протон

Усков А.В. Директор ООО «Академия Лад»

Сведения о педагогических (научно-педагогических) работниках, участвующих в реализации программы повышения квалификации, и лицах, привлекаемых к реализации дополнительной образовательной программы на иных условиях, представлены в таблице 2.

## Преподаватели программы повышения квалификации

## «Программирование на Python и Django»

| № п/п | Наименование дисциплин (модулей)              | Фамилия, имя, отчество, | Ученая степень, ученое звание | Основное место работы, должность | Место работы и должность по совместительству (если есть) |
|-------|---|-------------------------|-------------------------------|----------------------------------|--|
| 1.    | Дисциплина 1.<br>«Программирование на Python» | Лапаев Андрей Сергеевич |                               | Разработчик, компания Протон     | Старший преподаватель (ГБОУ ВО НГИЭУ)                    |
| 2.    | Дисциплина 2.<br>«СУБД MySQL»                 | Лапаев Андрей Сергеевич |                               | Разработчик, компания Протон     | Старший преподаватель (ГБОУ ВО НГИЭУ)                    |
| 3.    | Дисциплина 3.<br>«Фреймворк Django»           | Лапаев Андрей Сергеевич |                               | Разработчик, компания Протон     | Старший преподаватель (ГБОУ ВО НГИЭУ)                    |

### 7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

| № п.п. | Наименование дисциплины (модуля)              | Материально-технические условия для реализации программ (наличие лабораторий, производственных участков и т.п. по профилю программы повышения квалификации) |
|--------|---|---|
| 1.     | Дисциплина 1.<br>«Программирование на Python» | - Система дистанционного обучения: <a href="https://getcourse.ru">https://getcourse.ru</a>  |
| 2.     | Дисциплина 2. «СУБД MySQL»                    |   |
| 3      | Дисциплина 3. «Фреймворк Django»              |   |

УЧЕБНЫЙ ПЛАН  
ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ  
 «Программирование на Python и Django»

| № пп      | Наименование модулей, разделов и тем  | Всего, час. | В том числе: |  |
|-----------|---|-------------|--------------|--|
|           |   |             | Лекции       | Практические занятия (семинары), лабораторные работы |
| <b>1.</b> | <b>Дисциплина 1. «Программирование на Python»</b>   | <b>72</b>   | <b>20</b>    | <b>50</b>  |
|           | Ввод-вывод данных   | 3           | 1            | 2  |
|           | Условный оператор   | 3           | 1            | 2  |
|           | Типы данных. Циклы for и while.   | 3           | 1            | 2  |
|           | Строки и операции со строками   | 3           | 1            | 2  |
|           | Строковый тип данных  | 3           | 1            | 2  |
|           | Списки  | 3           | 1            | 2  |
|           | Функции   | 4           | 1            | 3  |
|           | Вложенные списки. Кортежи   | 4           | 2            | 2  |
|           | Множества. Словари  | 4           | 1            | 3  |
|           | Функции (высшего порядка, map(), filter(), reduce(), анонимные функции, any(), all(), zip(), enumerate()) | 4           | 1            | 3  |
|           | Работа с файлами  | 4           | 1            | 3  |
|           | Обработка исключений  | 4           | 1            | 3  |
|           | Функции (замыкания, декораторы)   | 4           | 1            | 3  |
|           | Генераторы и итераторы  | 4           | 1            | 3  |
|           | Декораторы и контекстные менеджеры  | 4           | 1            | 3  |
|           | Введение в ООП  | 4           | 1            | 3  |
|           | Атрибуты, свойства и методы   | 4           | 1            | 3  |
|           | Магические методы   | 4           | 1            | 3  |
|           | Наследование и полиморфизм  | 4           | 1            | 3  |
|           | Промежуточная аттестация  | 2           |              |  |
| <b>2.</b> | <b>Дисциплина 2. «СУБД MySQL»</b>   | <b>24</b>   | <b>8</b>     | <b>14</b>  |

|           |  |           |           |           |
|-----------|--|-----------|-----------|-----------|
|           | Основы реляционной модели и SQL. Отношение (таблица).                        | 3         | 1         | 2         |
|           | Выборка данных. Запросы, групповые операции.                                 | 3         | 1         | 2         |
|           | Вложенные запросы. Запросы корректировки данных.                             | 3         | 1         | 2         |
|           | Запросы на выборку, запросы корректировки                                    | 3         | 1         | 2         |
|           | SQL: выборка данных, сортировка, фильтрация, группировка, агрегатные функции | 3         | 1         | 2         |
|           | Связи между таблицами  | 3         | 1         | 2         |
|           | Запросы на выборку, соединение таблиц  | 2         | 1         | 1         |
|           | Запросы корректировки, соединение таблиц                                     | 2         | 1         | 1         |
|           | Промежуточная аттестация   | 2         |           |           |
| <b>3.</b> | <b>Дисциплина 3. «Фреймворк Django»</b>                                      | <b>46</b> | <b>20</b> | <b>26</b> |
|           | Основные понятия Django. Модель MTV.   | 2         | 1         | 1         |
|           | Связи. Ввод данных. Статические файлы  | 2         | 1         | 1         |
|           | Настройка проекта  | 2         | 1         | 1         |
|           | Модели: поля, связи между моделями, интернет-адрес модели, валидация модели. | 2         | 1         | 1         |
|           | Миграции. Запись данных.   | 2         | 1         | 1         |
|           | Выборка данных. Маршрутизация  | 2         | 1         | 1         |
|           | Контроллеры-функции и контроллеры-классы                                     | 2         | 1         | 1         |
|           | Шаблоны и статические файлы. Пагинатор                                       | 2         | 1         | 1         |
|           | Формы и наборы форм, связанные с моделями                                    | 2         | 1         | 1         |
|           | Разграничение доступа  | 2         | 1         | 1         |
|           | Модели: расширенные инструменты  | 3         | 2         | 1         |
|           | Формы и наборы форм: расширенные инструменты                                 | 3         | 1         | 2         |
|           | Поддержка баз данных PostgreSQL и библиотека django-localflavor              | 3         | 1         | 2         |
|           | Шаблоны: расширенные инструменты и Bootstrap 5                               | 3         | 1         | 2         |

|                     |   |          |    |    |
|---------------------|---|----------|----|----|
|                     | Обработка выгруженных файлов  | 3        | 1  | 2  |
|                     | Разграничение доступа: расширенные инструменты и Python Social Auth | 3        | 1  | 2  |
|                     | Посредники и обработчики контекста                                  | 3        | 1  | 2  |
|                     | Библиотека Django REST framework                                    | 3        | 1  | 2  |
|                     | Публикация веб-сайта  | 2        | 1  | 1  |
| Итоговая аттестация |   | <b>2</b> |    |    |
| Итого               |   | 144      | 48 | 90 |

