

**Филиал АНО ДТ Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб в г. Норильске»**

СОГЛАСОВАНО:

Заведующий по учебной части

 Н. В. Грицюк

Протокол № 12
от «3» июня 2024 г.

УТВЕРЖДАЮ:

Директор филиала

 Е. А. Дыптан

Приказ № 02.02-59
от «3» июня 2024 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА технической направленности
«Python 3. Создание клиент-серверных приложений на Django. Промышленный уровень»

Срок реализации: 1 год

Возраст обучающихся: 15-17 лет

Составитель программы: **Теников Л. Е.**,
педагог дополнительного образования

г. Норильск, 2024 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Python 3. Создание клиент-серверных приложений на Django. Промышленный уровень» имеет техническую направленность и разработана в соответствии с основными нормативно-правовыми документами: Федеральным Законом «Об образовании» от 29.12.2012 г. № 273-ФЗ; Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам от 09.11.2018 г. № 196; Целевой моделью развития региональных систем дополнительного образования детей от 03.09.2019 г. № 467; Концепцией развития дополнительного образования детей до 2030 года утвержденной распоряжением Правительства Российской Федерации от 31.03.2022 №678-р; Санитарно-эпидемиологическими требованиями к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи от 28.09.2020 г. № 28.

Программа является логическим продолжением (третьим годом обучения) программ «Python 3 для начинающих» (первый год обучения) и «Python 3 для новичков. Продвинутый Python» (второй год обучения).

В современном мире веб-программирование является важным и востребованным навыком. Одним из популярных фреймворков для веб-разработки является Django.

Django — это мощный и гибкий фреймворк, написанный на языке программирования Python. Он предоставляет разработчикам все необходимые инструменты и функциональность для создания высококачественных веб-приложений. Django обладает простым и интуитивно понятным синтаксисом, что делает его доступным для разработчиков разного уровня опыта.

Основные преимущества Django включают в себя встроенную административную панель, автоматическую генерацию форм и аутентификацию пользователей. Фреймворк также обеспечивает безопасность приложений и защиту от распространенных уязвимостей. С помощью Django разработчики могут создавать разнообразные веб-приложения, включая блоги, интернет-магазины, социальные сети и многое другое. Фреймворк предоставляет гибкую архитектуру и множество расширений, что позволяет разработчикам быстро и эффективно создавать функциональные и привлекательные веб-приложения.

Изучение Django открывает широкие возможности для карьерного роста в сфере веб-разработки. Знание этого фреймворка позволяет разработчикам создавать профессиональные веб-приложения, привлекать клиентов и успешно конкурировать на рынке.

Таким образом, Django является отличным выбором для разработки веб-приложений. Он предоставляет все необходимые инструменты и функциональность для создания высококачественных и мощных веб-приложений.

1.1. АКТУАЛЬНОСТЬ

Умение создавать веб-сайты на Django, которое появится после изучения данной программы, является актуальным и востребованным навыком по нескольким причинам:

1. популярность Django: Django является одним из самых популярных фреймворков для веб-разработки. Он имеет большое сообщество разработчиков, активную поддержку и обширную документацию. Благодаря этому, разработчики, владеющие Django, могут легко найти поддержку и ресурсы для решения возникающих вопросов и проблем;

2. продуктивность разработки: Django предоставляет разработчикам мощные инструменты и функциональность, которые значительно упрощают и ускоряют процесс создания веб-сайтов. Встроенная административная панель, автоматическая генерация форм и аутентификация пользователей позволяют сосредоточиться на бизнес-логике приложения, минимизируя время, затраченное на разработку базовых функций;

3. безопасность и надежность: Django обеспечивает высокий уровень безопасности веб-приложений. Он предоставляет встроенные механизмы защиты от распространенных уязвимостей, таких как атаки CSRF и XSS. Фреймворк также обладает строгими правилами организации кода, что способствует созданию надежных и безопасных приложений;

4. масштабируемость: Django предоставляет гибкую архитектуру, которая позволяет разрабатывать масштабируемые веб-приложения. Фреймворк поддерживает разделение приложений на модули, что упрощает их разработку, тестирование и поддержку. Благодаря этому веб-сайты на Django могут легко масштабироваться и адаптироваться к растущим потребностям бизнеса;

5. рыночный спрос: Веб-разработка является одной из самых востребованных областей в IT-индустрии. Компании и предприниматели нуждаются в профессионалах, способных создавать качественные и функциональные веб-сайты. Умение работать с Django открывает широкие возможности для карьерного роста и повышения конкурентоспособности на рынке труда;

Таким образом, умение создавать веб-сайты на Django актуально и востребовано в современном мире веб-разработки. Оно позволяет разработчикам быть продуктивными, создавать безопасные и масштабируемые приложения, а также отвечает на растущий спрос на профессионалов веб-разработки.

1.2. ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ

Программа обучения программированию на Python для детей имеет несколько уникальных особенностей, которые делают ее эффективной и интересной:

1. интерактивность и игровой подход: программой предусмотрены интерактивные задания и проекты, которые делают обучение увлекательным и мотивирующим. Дети могут учиться, играя, что стимулирует их интерес к программированию;

2. проектная ориентированность: программа строится вокруг реальных проектов, которые имеют практическое применение. Это позволяет обучающимся применять свои знания в творческих задачах и создавать свои собственные программные продукты;

3. работа в команде: обучающиеся имеют возможность работать в команде над проектами и обмениваться опытом. Это помогает развивать социальные навыки и стимулирует творческое мышление.

Все эти особенности создают обучающую среду, которая не только передает технические навыки программирования, но и развивает творческое мышление и радость от обучения у детей.

1.3. АДРЕСАТ ПРОГРАММЫ

Ввиду специфики ДООП, к освоению допускаются лица, успешно прошедшие обучение по программе «Python 3 для новичков. Продвинутый Python».

Программа «Python 3. Создание клиент-серверных приложений на Django. Промышленный уровень» ориентирована на обучающихся от 15 до 17 лет. Занятия проводятся в группе из 12 человек.

Набор на программу осуществляется в соответствии с Положением о наборе в филиале АНО ДТ «Красноярский «Кванториум» в г. Норильске «Центр цифрового образования детей IT-Куб г. Норильск».

1.4. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ

Программа обладает рядом ключевых особенностей, которые делают ее педагогически обоснованной и эффективной:

1. индивидуальный подход: программа учитывает индивидуальные потребности и темп усвоения каждого ученика. Это позволяет обеспечить более эффективное обучение и лучшее понимание материала;

2. развитие критического мышления: обучение программированию способствует развитию критического мышления. Обучающиеся учатся анализировать, решать проблемы и развивать алгоритмическое мышление, что является важным аспектом их образовательного прогресса;

3. творческий рост: программа активно вовлекает детей в творческий процесс. Работа над проектами и решение практических задач способствуют раскрытию их творческого потенциала, что стимулирует интерес и удовлетворение от обучения;

4. практические навыки: программа ориентирована на практическое применение знаний. Обучающиеся не только получают теоретические сведения, но и могут успешно применять их в создании реальных программных продуктов;

5. стимулирование интереса: игровой подход и интересные задачи создают стимул для обучения. Школьники не только учатся, но и получают удовольствие от процесса изучения программирования, что делает обучение более продуктивным и запоминающимся;

6. коллективное обучение: программа предоставляет возможность для коллективного обучения, что способствует развитию социальных навыков, умению работать в команде и обмену знаниями и опытом;

7. подготовка к будущему: обучение программированию на Python дает школьникам конкретные навыки, востребованные в современном мире. Они получают базовые знания программирования и готовность к дальнейшей углубленной учебе или даже будущей карьере в IT-сфере.

Все эти аспекты делают программу обучения для детей педагогически обоснованной, обеспечивая комплексный и целенаправленный подход к обучению.

1.5. СРОК ОСВОЕНИЯ ПРОГРАММЫ

Программа рассчитана на 144 учебных часа. Срок освоения программы – 1 учебный год.

1.6. ФОРМА ОБУЧЕНИЯ, ГРАФИК ЗАНЯТИЙ

Форма обучения: очная.

Режим занятий: 2 раза в неделю по 2 академических часа (1 ак. час = 40 минут) с обязательным перерывом.

1.7. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Целью программы является формирование углубленных практических навыков при работе с языком программирования Python в контексте разработки клиент-серверных приложений. Для достижения поставленной цели планируется решить следующие задачи:

- сформировать и развить навыки алгоритмического и логического мышления, необходимые для разработки веб-приложений;
- познакомить с основными принципами и этапами программирования и разработки программного обеспечения для веб-приложений;

- обучить использованию интегрированных сред разработки, специфичных для веб-разработки;
- освоить конструкции языка программирования Python, применяемые в веб-разработке;
- познакомить с популярными модулями Python, используемыми в веб-разработке, такими как Flask, Django, SQLAlchemy и другими;
- научить применять объектно-ориентированный стиль написания кода при разработке веб-приложений;
- привить навыки создания новых классов данных, специфичных для веб-приложений;
- познакомить с процессом разработки веб-проекта и алгоритмом его создания;
- развить у учащихся инициативность, самостоятельность и стремление к получению качественного результата в веб-разработке;
- воспитать социально значимые качества личности, такие как ответственность, коммуникабельность, добросовестность, пытливость ума и критичность мышления;
- развить логическое, абстрактное и образное мышление, необходимые для эффективной веб-разработки;
- сформировать самостоятельность и творческий подход к решению задач веб-разработки с использованием средств вычислительной техники;
- привить навыки поиска информации в сети Интернет, анализа выбранной информации и использования ее при разработке веб-приложений.

Обучение начинается с повторения основ языка Python и его синтаксиса, а затем происходит переход к разработке веб-приложений. Учащиеся узнают, как работать с веб-фреймворками, такими как Flask или Django, и как использовать их для создания веб-приложений. Они также изучают работу с базами данных, обработку форм, аутентификацию пользователей и другие аспекты веб-разработки. В заключение, обучающиеся продолжают знакомиться с принципами объектно-ориентированного программирования и учиться применять их в контексте веб-разработки.

1.8. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

В результате освоения программы обучающиеся должны освоить:

Предметные результаты:

- сформированы и развиты навыки алгоритмического и логического мышления, необходимые для разработки веб-приложений;
- понимание основных принципов и этапов программирования и разработки программного обеспечения для веб-приложений;
- навыки использования интегрированных сред разработки, специфичных для создания веб-сервисов;

- приобретены навыки использования конструкций языка программирования Python в веб-разработке;
- получен опыт работы с популярными модулями Python, используемыми в веб-разработке, такими как Flask, Django, SQLAlchemy и другими;
- получен опыт применения объектно-ориентированного стиля написания кода при разработке веб-приложений;
- приобретены навыки создания новых классов данных, специфичных для веб-приложений;
- получены специфические знания для работы с процессом проектирования веб-сервисов и алгоритмом их создания.

Метапредметные результаты:

- развиты логическое, абстрактное и образное мышление, необходимые для эффективной веб-разработки;
- сформирована самостоятельность и творческий подход к решению задач веб-разработки с использованием средств вычислительной техники;
- приобретены навыки поиска информации в сети Интернет, анализа выбранной информации и грамотное её использование при разработке веб-приложений.

Личностные результаты

- развиты инициативность, самостоятельность и стремление к получению качественного результата в веб-разработке;
- воспитаны социально значимые качества личности, такие как ответственность, коммуникабельность, добросовестность, пытливость ума и критичность мышления.

1.9. ИТОГОВАЯ АТТЕСТАЦИЯ ОБУЧАЮЩИХСЯ

Система контроля знаний и умений обучающихся представляется в виде:

1. Входной контроль осуществляется при комплектовании группы в начале учебного года. Цель – определить исходный уровень знаний обучающихся, определить формы и методы работы с обучающимися. Форма контроля: тестирование.
2. Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и

др. Они активизируют, стимулируют работу обучающихся, позволяют более полно проявлять полученные знания, умения, навыки.

3. Промежуточная аттестация осуществляется в конце I полугодия учебного года. Форма контроля: тестирование, решение практических задач средствами языка программирования и механизмами веб-фреймворка.

4. Итоговая аттестация осуществляется в конце учебного года. Форма контроля: защита проекта. Проект является одним из видов самостоятельной работы, предусмотренной в ходе обучения по программе. Педагог-наставник оказывает консультационную помощь в выполнении проекта.

Индивидуальный (групповой) проект оценивается формируемой комиссией. Состав комиссии (не менее 3-х человек): педагог-наставник, администрация учебной организации, приветствуется привлечение IT-профессионалов, представителей высших и других учебных заведений.

Компонентами оценки индивидуального (группового) проекта являются (по мере убывания значимости): качество индивидуального проекта, отзыв руководителя проекта, уровень презентации и защиты проекта. Если проект выполнен группой обучающихся, то при оценивании учитывается не только уровень исполнения проекта в целом, но и личный вклад каждого из авторов. Решение принимается коллегиально.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Тема занятия	Количество часов			Форма заня- тий	Формы атте- стации / кон- троля
		Всего	Теория	Прак- тика		
Раздел 1. Повторение пройденного материала (6 часов)						
1.1	Повторение синтаксиса Python	2	1	1	Лекция / Практика	
1.2	Повторение функционального программирования	2	1	1	Лекция / Практика	
1.3	Повторение ООП	2	1	1	Лекция / Практика	
Раздел 2. Система контроля версий git (17 часов)						
2.1	Введение в системы контроля версий	2	1	1	Лекция / Практика	
2.2	Начало работы с git	2	1	1	Лекция / Практика	
2.3	Работа с изменениями	4	2	2	Лекция / Практика	
2.4	Ветки в git	5	2	3	Лекция / Практика	
2.5	Git и GitHub	4	1	3	Лекция / Практика	
Раздел 3. Веб-технологии и базовые сведения об HTML (12 часов)						
3.1	Базовые сведения о веб-технологиях	2	1	1	Лекция / Практика	
3.2	Введение в HTML	2	1	1	Лекция / Практика	
3.3	Введение в CSS	2	1	1	Лекция / Практика	
3.4	Введение в JavaScript	2	1	1	Лекция / Практика	
3.5	Создание простой веб-страницы	4	2	2	Лекция / Практика	
Раздел 4. Макетирование веб-страниц при помощи Bootstrap 5 (13 часов)						
4.1	Введение в фреймворк BS 5	2	1	1	Лекция / Практика	

4.2	Контейнеры и сетка Bootstrap	2	1	1	Лекция / Практика	
4.3	Верстка макета	2	1	1	Лекция / Практика	
4.4	Задание цвета, отступов и выравнивания содержимого страницы	3	1	2	Лекция / Практика	
4.5	Таблицы BS	2	1	1	Лекция / Практика	
4.6	Пример использования адаптивных контейнеров	2	1	1	Лекция / Практика	Тестирование
Раздел 5. Фреймворк Django (54 часа)						
5.1	Введение в Django	2	1	1	Лекция / Практика	
5.2	Структура приложений на Django. Паттерн MVC	2	1	1	Лекция / Практика	
5.3	Создание приложения. Последовательность создания проекта	2	1	1	Лекция / Практика	
5.4	Обработка запросов пользователя	8	2	6	Лекция / Практика	
5.5	Шаблоны	6	3	3	Лекция / Практика	
5.6	Формы	8	2	4	Лекция / Практика	
5.7	Модели данных	12	6	6	Лекция / Практика	
5.8	Работа с админ-панелью	6	2	4	Лекция / Практика	
5.9	Права и группы прав пользователей	8	2	6	Лекция / Практика	Тестирование
Раздел 6. Публикация веб-приложения в сети Интернет (10 часов)						
6.1	Подготовка инфраструктуры сайта перед публикацией	4	2	2	Лекция / Практика	
6.2	Размещение на хостинге	6	3	3	Лекция / Практика	
Раздел 7. Итоговый проект (34 часа)						

7.1	Выполнение проекта	32	2	30	Практика	
7.2	Защита проекта	2	2	0	Защита проекта	
Итого часов		144	51	93		

3. СОДЕРЖАНИЕ ПРОГРАММЫ

Раздел 1. Повторение пройденного материала (6 часов)

Тема 1.1: Повторение синтаксиса Python (2 часа)

Теория: Типы данных, переменные, функции, условия, циклы, ООП.

Практическая работа: Решение задач на пройденный ранее материал.

Тема 1.2: Повторение функционального программирования (2 часа).

Теория: Структурный стиль программирования.

Практическая работа: Решение задач.

Тема 1.3: Повторение ООП (2 часа).

Теория: Объектно-ориентированная парадигма программирования. Абстракция, инкапсуляция, наследование, полиморфизм.

Практическая работа: Написание кода игры в ООП стиле.

Раздел 2. Система контроля версий git (17 часов)

Тема 2.1: Введение в систему контроля версий Git (2 часа).

Теория: Теория о системе контроля версий. Отличие Git от других VCS.

Практическая работа: Установка Git на Linux, macOS, Windows.

Тема 2.2: Начало работы с git (2 часа).

Теория: Работа в локальном репозитории, фиксация изменений, игнорирование файлов.

Практическая работа: Создание репозитория, создание файла readme.md.

Тема 2.3: Работа с изменениями (4 часа)

Теория: Работа с историей. Откат изменений.

Практическая работа: Откат изменений и работа с предыдущими версиями проекта.

Тема 2.4: Ветки в git (5 часов)

Теория: Ветки. Теги.

Практическая работа: Работа по слиянию веток репозитория, откат изменений и создание новых веток.

Тема 2.5: Git и GitHub (4 часа)

Теория: Регистрация и знакомство с GitHub. Процесс загрузки проекта.

Практическая работа: Загрузка проекта с локальной машины на сервер GitHub.

Раздел 3. Веб-технологии и базовые сведения об HTML (12 часов)

Тема 3.1: Базовые сведения о веб-технологиях (2 часа)

Теория: Основные понятия HTML.

Практическая работа: Ознакомление с веб-браузером и его возможностями для управления отображением контента.

Тема 3.2: Введение в HTML (2 часа)

Теория: Правила разметки страниц в сети Интернет. Базовые теги HTML.

Практическая работа: Создание простой веб-страницы.

Тема 3.3: Введение в CSS (2 часа)

Теория: Введение в стилизацию веб-страниц. Знакомство с основными принципами и возможностями каскадных таблиц стилей.

Практическая работа: Стилизация созданной веб-страницы.

Тема 3.4: Введение в JavaScript (2 часа)

Теория: Базовые сведения о JavaScript.

Практическая работа: Добавление интерактивных элементов к странице.

Тема 3.5: Создание простой веб-страницы (4 часа)

Теория: Повторение HTML/CSS/JavaScript.

Практическая работа: Создание веб-страницы с использованием технологий HTML/CSS/JavaScript.

Раздел 4. Макетирование веб-страниц при помощи Bootstrap 5 (13 часов)

Тема 4.1: Введение в фреймворк BS 5 (2 часа)

Теория: История появления frontend-фреймворка.

Практическая работа: Подключение BS 5 к созданной странице.

Тема 4.2: Контейнеры и сетка Bootstrap (2 часа)

Теория: Основные элементы верстки веб-страниц.

Практическая работа: Добавление сайту адаптивности.

Тема 4.3: Верстка макета (2 часа)

Теория: Пример создания веб-сайта с использованием bootstrap 5.

Практическая работа: Верстка по макету при помощи BS 5.

Тема 4.4: Задание цвета, отступов и выравнивания содержимого страницы (3 часа)

Теория: Возможности BS 5 для гибкой настройки внешнего вида веб-сайта.

Практическая работа: Добавление к элементам сайта дополнительных классов.

Тема 4.5: Таблицы BS (2 часа)

Теория: Таблицы и их стили в BS 5.

Практическая работа: Создание таблиц и их оформление.

Тема 4.6: Пример использования адаптивных контейнеров (2 часа)

Теория: Процесс создания веб-сайта из готовых модулей фреймворка.

Практическая работа: Создание сайта с использованием готовых шаблонов верстки.

Раздел 5. Фреймворк Django (52 часа)

Тема 5.1: Введение в Django (2 часа).

Теория: Веб-фреймворки. Front-end и Back-end разработка.

Практическая работа: Установка библиотеки Django.

Тема 5.2: Структура приложений на Django. Паттерн MVC (2 часа).

Теория: MVC паттерн проектирования. Структура Django-проекта.

Практическая работа: Основные команды фреймворка Django.

Тема 5.3: Создание приложения. Последовательность создания проекта (2 часа).

Теория: Основные этапы создания клиент-серверного приложения. Этапы создания и запуска проекта на Django.

Практическая работа: Создание заготовки под проект. Первичная настройка.

Тема 5.4 Обработка запросов пользователя (8 часов)

Теория: Маршрутизация запросов пользователя. Понятие представления.

Практическая работа: Создание списка страниц сайта. Описание view-функций.

Тема 5.5: Шаблоны (6 часов).

Теория: Механизм работы шаблонов.

Практическая работа: Создание шаблонов страниц проектируемого сайта. Определение вида базового шаблона сайта.

Тема 5.6: Формы (8 часов).

Теория: Формы. Виджеты. Типы полей форм и описание процедуры отправки и обработки данных сервером Django.

Практическая работа: Определение форм для ввода данных на сайте.

Тема 5.7: Модели данных (10 часов).

Теория: Базы данных. Типы БД. Механизм ORM для манипулирования данными в Django. Проектирование структуры БД. Типы связей таблиц в реляционных БД.

Практическая работа: Проектирование структуры базы данных, проведение миграций.

Тема 5.8: Работа с админ-панелью (6 часов).

Теория: Регистрация созданных моделей данных для отображения и манипулирования в интерфейсе админ-панели. Процесс заполнения таблиц.

Практическая работа: Заполнение БД реальными данными.

Тема 5.9: Права и группы прав пользователей (8 часов).

Теория: Разграничение прав доступа к сайту на основе разрешений.

Практическая работа: Добавление ролей пользователя в систему.

Раздел 6. Публикация веб-приложения в сети Интернет (10 часов)

Тема 6.1: Подготовка инфраструктуры сайта перед публикацией (4 часа).

Теория: Выбор хостинга. Настройка Django проекта перед публикацией.

Практическая работа: Настройка проекта Django перед публикацией.

Тема 6.2: Размещение на хостинге (6 часов).

Теория: Выбор хостинг-провайдера для размещения веб-приложения Django в сети Интернет.

Практическая работа: Размещение файлов проекта на хостинге. Проверка работоспособности сайта.

Раздел 7. Итоговый проект (34 часа)

Тема 7.1: Выполнение проекта (32 часа).

Теория: Повторение основных этапов построения веб-приложений. Постановка задачи.

Практическая работа: Создание своего веб-приложения по выбранной учеником теме. Подготовка к защите.

Тема 7.2: Защита проекта (2 часа).

Теория: Рекомендации к оформлению презентации.

Практическая работа: Защита проекта.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Материально – техническое обеспечение:

1. стол преподавателя;
2. стул преподавателя;
3. стол обучающегося;
4. стул обучающегося;
5. рабочая станция преподавателя;
6. ноутбук обучающегося;
7. маркерная доска;
8. проектор;
9. МФУ;
10. точки подключения к электрической сети.

Программное обеспечение:

1. Операционная система Windows 10;
2. JetBrains PyCharm Community IDE;
3. веб-браузер Edge/Chrome.

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. Меле Антонио. Django 4 в примерах.: ДМК Пресс, 2023
2. Джон Дакетт. HTML и CSS Разработка и дизайн веб-сайтов.: Эксмо, 2019;
3. Анатолий Постолиит. Python, Django и Bootstrap для начинающих.: БХВ, 2023;
4. Морето Сильвио. Bootstrap в примерах.: ДМК Пресс, 2017.

Дополнительная литература:

1. Джей Джей Гивакс. Паттерны проектирования API.: Питер, 2023.

Электронные ресурсы:

1. Сайт python.org – Документация интерпретатора языка Python 3;
2. Сайт pythonworld.ru – «Python 3 для начинающих»;
3. Сайт pythontutor.ru – «Питонтьютор»;
4. Сайт getbootstrap.com/docs/5.3 – официальная документация фреймворка Bootstrap 5;
5. Сайт bootstrap5.ru – русскоязычная документация Bootstrap 5;
6. Сайт djangoproject.com – официальная документация фреймворка Django.

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

Теников Л. Е., педагог дополнительного образования, студент ЗГУ им. Федоровского.

7. МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ПО РАЗДЕЛАМ ПРОГРАММЫ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Отметка за тест формируется следующим образом:

«Удовлетворительно» – За более 50% и менее 65% правильно выполненных заданий;

«Хорошо» – за более 65% и менее 75% правильно выполненных заданий;

«Отлично» – за правильное выполнение более 75% заданий.

Примеры заданий

Раздел 1. Повторение пройденного материала:

1. В Python какой оператор используется для условного выполнения кода?

- a) **if**
- b) for
- c) while
- d) switch

2. Какой тип данных в Python используется для хранения последовательности элементов?

- a) int
- b) str
- c) **list**
- d) bool

3. Какой метод в Python используется для объединения двух строк?

- a) **join()**
- b) split()
- c) append()
- d) upper()

4. Какой цикл в Python позволяет выполнять код определенное количество раз?

- a) **for**
- b) while
- c) do-while
- d) switch

5. Какой принцип ООП позволяет создавать новые классы на основе существующих?

- a) **наследование**
- b) инкапсуляция
- c) абстракция
- d) полиморфизм

Раздел 2. Система контроля версий git:

1. Какая команда используется для просмотра истории коммитов в Git?

- a) git log
- b) git status
- c) git branch
- d) **git checkout**

2. Какой ключ используется с командой git clone для клонирования репозитория с определенной веткой?

- a) **-b**
- b) -c
- c) -r
- d) -m

3. Какая команда используется для отмены последнего коммита в Git?
- a) git commit
 - b) git push
 - c) git revert
 - d) **git reset**
4. Какой тип слияния используется в Git, когда создается новая ветка на основе существующей?
- a) **merge**
 - b) rebase
 - c) cherry-pick
 - d) squash
5. Какая команда используется для отправки изменений в удаленный репозиторий в Git?
- a) **git push**
 - b) git pull
 - c) git commit
 - d) git clone

Раздел 3. Веб-технологии и базовые сведения об HTML:

1. Какой тег используется для создания заголовка первого уровня в HTML?
- a) **<h1>**
 - b) <p>
 - c) <div>
 - d)
2. Какой атрибут используется для задания ссылки на внешний CSS-файл в HTML?
- a) **href**
 - b) src
 - c) class
 - d) style
3. Какой тег используется для создания ссылки в HTML?
- a) **<a>**
 - b)
 - c) <div>
 - d)
4. Какой тег используется для создания неупорядоченного списка в HTML?
- a) ****
 - b)
 - c)
 - d) <dl>
5. Какой атрибут используется для задания цвета фона элемента в HTML?
- a) **background-color**
 - b) color
 - c) font-size
 - d) text-align

Раздел 4. Макетирование веб-страниц при помощи Bootstrap 5:

1. Какой класс используется в Bootstrap для создания навигационного меню?
- a) **.navbar**
 - b) .container
 - c) .row
 - d) .col

2. Какой класс используется в Bootstrap для создания кнопки?
- a) **.btn**
 - b) `.button`
 - c) `.link`
 - d) `.clickable`
3. Какой класс используется в Bootstrap для создания карточки с содержанием?
- a) **.card**
 - b) `.container`
 - c) `.row`
 - d) `.col`
4. Какой класс используется в Bootstrap для создания модального окна?
- a) **.modal**
 - b) `.popup`
 - c) `.window`
 - d) `.dialog`
5. Какой класс используется в Bootstrap для создания адаптивной таблицы?
- a) **.table**
 - b) `.grid`
 - c) `.container`
 - d) `.row`

Раздел 5. Фреймворк Django:

1. Какой файл в Django отвечает за маршрутизацию URL-адресов?
- a) **urls.py**
 - b) `views.py`
 - c) `models.py`
 - d) `settings.py`
2. Какой метод в Django используется для обработки GET-запросов?
- a) **get()**
 - b) `post()`
 - c) `put()`
 - d) `delete()`
3. Какой тег используется в Django шаблонах для вывода переменной?
- a) **{{ }}**
 - b) `{% % }`
 - c) `[]`
 - d) `()`
4. Какой метод в Django используется для создания нового объекта модели?
- a) **create()**
 - b) `save()`
 - c) `update()`
 - d) `delete()`
5. Какой класс в Django используется для создания форм?
- a) **forms.Form**
 - b) `forms.ModelForm`
 - c) `forms.FormSet`
 - d) `forms.ModelFormSet`

Раздел 6. Публикация веб-приложения в сети Интернет:

1. Какая команда используется для размещения файлов проекта на хостинге?
- a) `git push`
 - b) `git commit`

c) git clone **d) Никакая из вышеперечисленных**

2. Какой файл в Django отвечает за настройку проекта перед публикацией?

- a) **settings.py** b) urls.py
c) views.py d) models.py

3. Какой протокол используется для передачи файлов на хостинг?

- a) **FTP** b) HTTP
c) SSH d) SMTP

4. Какой сервис облачного хостинга позволяет быстро и относительно просто разместить веб-приложение Django?

- a) **Heroku** b) AWS
c) Google Cloud d) Azure

5. Какая команда используется для проверки работоспособности сайта после размещения на хостинге?

- a) **ping** b) test
c) check d) Никакая из вышеперечисленных

Раздел 7. Итоговый проект:

1. Какой этап является первым при создании клиент-серверного приложения на Django?

- a) **Создание заготовки под проект**
b) Определение форм для ввода данных на сайте
c) Проектирование структуры базы данных
d) Заполнение БД реальными данными

2. Какой метод в Django используется для обработки POST-запросов?

- a) **post()** b) get()
c) put() d) delete()

3. Какой тег используется в Django шаблонах для создания ссылки?

- a) **<a>** b) <link>
c) <url> d) <href>

4. Какой класс в Django используется для создания пользовательских моделей данных?

- a) **models.Model** b) models.User
c) models.Form d) models.View

5. Какой метод в Django используется для сохранения данных в базе данных?

- a) **save()** b) create() c) update() d) delete()