

Государственное бюджетное профессиональное образовательное учреждение
Псковской области «Опочецкий индустриально-педагогический колледж»

Центр цифрового образования детей «IT – куб»

ПРИНЯТО
Педагогическим советом
Протокол № 2 от 09.02.2024 г.

УТВЕРЖДЕНО
приказом
исполняющего обязанности
директора колледжа
И.А. Гайдовской
от 23.04.2024г № 207

Дополнительная общеразвивающая программа
«Прикладное программирование на PYTHON»

Уровень: базовый

Направленность: техническая

Возраст обучающихся: 14-17 лет

Срок реализации: 1 год

Разработчики:

Иванова С.П., зам. директора по
учебно-методической работе;
Иванова С.Т., методист

г. Опочка
2024 г.

Содержание

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
УЧЕБНЫЙ ПЛАН	13
СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА	16
КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ	22
СПИСОК ЛИТЕРАТУРЫ	25
Приложение 1	26
Приложение 2	27

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Нормативно-правовая база разработки и реализации программы.

Программа разработана на основе следующих нормативных документов:

- Федерального закона от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020);
- Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденного приказом Министерства просвещения Российской Федерации от 27.07.2022 г. № 629;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
- Приказа Минобрнауки России №882, Минпросвещения России №391 от 05.08.2020 «Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ» (вместе с «Порядком организации и осуществления образовательной деятельности при сетевой форме реализации образовательных программ»);
- Методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» (письмо Министерства просвещения Российской Федерации от 10 ноября 2021 г. № ТВ-1984/04);
- Стратегии развития воспитания в Российской Федерации на период до 2025 года (Утверждена распоряжением Правительства РФ от 29.05.2015 № 996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»);

– Концепции развития дополнительного образования детей до 2030 года, утвержденной распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р.

2. Новизна, актуальность, педагогическая целесообразность реализации программы.

Актуальность данной программы в том, что она составлена с учётом современных потребностей рынка в специалистах в области информационных технологий, особенно в областях программирования и работы с большими объемами данных. Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных и отраслевых задач. Знания и навыки, приобретенные в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию и алгоритмике, при решении задач по общим дисциплинам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования. Программа является практико-ориентированной, прикладной.

Новизна данной программы заключается в том, что учащиеся продолжают изучение современного универсального языка Python и применяют полученные навыки, решая задачи, связанные с созданием веб-запросов и веб-сайтов с использованием динамического строения блоков сайта, объектно-ориентированной парадигмы программирования и взаимодействием с базами данных. В процессе реализации данной программы учащиеся совершенствуют знания объектно-ориентированной парадигмы программирования и методами ее использования для реализации более сложных приложений, а также совершенствуют навыки работы с базами данных и использования профессиональных средств разработки программного обеспечения, таких как интегрированные системы разработки.

Педагогическая целесообразность программы определяется образовательным замыслом. В процессе освоения программы обучающиеся

закрепят полученные на стартовом уровне знания и получают новые знания необходимые для разработки прикладных программ.

Направленность программы: техническая.

3. Адресат программы.

Адресатом программы являются дети в возрасте от 14 до 17 лет.

Содержание и объем стартовых знаний, необходимых для начального этапа освоения программы: знание синтаксиса языка Python, понимание основных алгоритмических структур, умение работать с базовыми структурами данных, знание объектно-ориентированной парадигмы программирования и принципов строения приложений на языке Python.

4. Срок реализации программы.

Срок реализации программы составляет 1 год.

Уровень программы: базовый.

Программа предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка, гарантированно обеспечивают трансляцию общей и целостной картины в рамках содержательно-тематического направления программы.

5. Форма реализации программы.

Форма обучения – очная.

Образовательные технологии: информационные технологии, проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

Форма организации содержания и процесса педагогической деятельности – комплексная.

Тип организации работы учеников: групповая работа, индивидуальная, коллективная.

Виды занятий: лекции и практические занятия.

Наполняемость группы: от 10 до 12 человек.

6. Объём программы и режим работы.

Объём программы: 144 часа.

Режим занятий: 2 раза в неделю по 2 академических часа.

Продолжительность часа – 40 минут.

7. Цель программы:

Целью программы является создание условий для обучения созданию динамически изменяемых веб-страниц на языке Python с использованием языков HTML и CSS и тестирования приложений.

8. Задачи программы.

Программа направлена на решение следующих задач:

- изучение устройства веб-запросов;
- изучение основ HTML и CSS;
- изучение принципов и методов разработки веб-страниц;
- изучение принципов разработки unit-тестов;
- формирование навыков разработки веб-страниц;
- формирование навыков тестирования приложений;
- развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- формировать коммуникативные навыки (диалогическая и монологическая речь при защите проектов);
- развитие навыков поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- способствовать формированию познавательных и регулятивных универсальных учебных действий;

- планировать и регулировать собственную деятельность по реализации проекта, доводить начатые проекты до конца;
- формировать коммуникативные навыки (диалогическая и монологическая речь при защите проектов);
- воспитание умения работать в коллективе с учетом личностных качеств учащихся, психологических и возрастных особенностей;
- развитие рефлексивную деятельность учащихся;
- воспитание трудолюбия, целеустремленности, уважения к труду;
- воспитание информационной культуры.

9. Планируемые результаты освоения программы.

В результате освоения программы, учащиеся будут:

Знать:

- принципы устройства веб-запросов;
- принципы построения и методы разработки веб-страниц;
- принципы использования unit-тестов.

Уметь:

- разрабатывать веб-страниц для реализации поставленных задач;
- создавать макет сайта с помощью языка HTML;
- использовать таблицы стилей при создании веб-сайта с помощью языка CSS;
- использовать инструменты тестирования ПО;
- составлять веб-запросы и формировать ответы;
- планировать и регулировать собственную деятельность по реализации проекта, доводить начатые проекты до конца;
- самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;
- критически оценивать правильность решения задачи;
- корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;

Владеть:

- навыками решения задач, построения алгоритмов решения;
- навыками разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;

Метапредметные результаты:

- развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- развитие навыков поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- развитие коммуникативных навыков и навыков работы в группе.

Личностные результаты:

- воспитание умения работать в коллективе с учетом личностных качеств учащихся, психологических и возрастных особенностей;
- развитие рефлексивной деятельности учащихся;
- воспитание трудолюбия, целеустремленности, уважения к труду;
- воспитание информационной культуры.

10. Формы представления результатов.

Формы аттестации: беседа, демонстрация решения, наблюдение, защита проектов.

Входной контроль осуществляется в начале реализации программы в форме беседы и наблюдения и имеет диагностические задачи. Цель входной диагностики – зафиксировать начальный уровень подготовки обучающихся, имеющиеся знания, умения и навыки, связанные с предстоящей деятельностью.

Наблюдение осуществляется в течение реализации программы.

Промежуточный контроль осуществляется в целях диагностики теоретических знаний и практических умений и навыков по итогам освоения одного из разделов курса. Проводится в форме демонстрации проекта.

Итоговый контроль проводится с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения – представляет из себя защиту проекта.

11. Оценочные материалы, формирующие систему оценивания.

Промежуточное тестирование состоит из задач, которые требуют развернутого ответа. Примеры задач приведены в приложении 1. Максимальный балл за промежуточное тестирование: 55 баллов.

Оценивание тестирования осуществляется по следующим уровням:
высокий уровень – учащийся набрал не менее 80% от максимально возможного количества баллов (от 44 балла).

средний уровень – учащийся набрал не менее 50% от максимально возможного количества баллов (от 27 баллов).

низкий уровень – учащийся набрал менее 50% от максимально возможного количества баллов (менее 27 баллов).

Итоговый контроль представляет из себя защиту проекта. Проект представляет собой программную реализацию на языке программирования Python, по предложенной теме. В проекте реализуются основные функции автоматизирующие процессы предметной области и направленные на оптимизацию рабочих процессов.

Критерии оценки проекта

№	Название критерия	Максимальный балл
1.	Актуальность и проработанность проблемы	До 5 баллов
2.	Четкость формулировки целей и задач	До 5 баллов
3.	Технологическая сложность проекта. Наличие нестандартных структур данных и алгоритмов, специфика которых направлена на решение конкретной задачи.	До 10 баллов
4.	Новизна и оригинальность решения	До 5 баллов
5.	Качество разработанного продукта. Наличие удобного и понятного интерфейса, как абстракции между реализацией и пользователем.	До 5 баллов
6.	Защита проекта: <ul style="list-style-type: none">• качество презентации;	До 5 баллов

	• четкость и ясность изложения, умение взаимодействовать с аудиторией, отвечать на вопросы	
7.	Наличие самооценки и перспектив дальнейшей разработки проекта	До 5 баллов
Итого		40 баллов

Оценивание проекта осуществляется по следующим уровням:

Высокий уровень – учащийся набрал не менее 32 баллов по итогам защиты проекта.

Средний уровень – учащийся набрал от 20 до 31 балла по итогам защиты проекта.

Низкий уровень – учащийся набрал менее 20 баллов по итогам защиты проекта.

Общими критериями оценки результативности обучения являются:

- оценка уровня теоретических знаний: свобода восприятия теоретической информации, осмысленность и свобода использования специальной терминологии, свобода ориентации в теоретическом материале;

- оценка уровня практической подготовки: соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением;

- оценка уровня достижения личностных результатов: культура организации самостоятельной деятельности, культура работы с информацией, аккуратность и ответственность при работе.

Оценка итоговых результатов освоения программы осуществляется по трем уровням:

Высокий уровень – достижение 80 - 100% показателей освоения программы.

Средний уровень – достижение 50 - 79% показателей освоения программы.

Низкий уровень - достижение менее чем 50% показателей освоения программы.

Достиженные обучающимся знания, умения и навыки заносятся в сводную таблицу результатов обучения.

Оценка уровней освоения программы

Уровни	Параметры	Показатели
Высокий уровень (80-100%)	Теоретические знания	Обучающийся освоил материал в полном объеме. Знает и понимает значение терминов, употребляет их осознанно и в полном соответствии с содержанием. Самостоятельно ориентируется в содержании материала по темам.
	Практические умения и навыки	Обучающийся овладел 80-100% умений и навыков, предусмотренных программой за конкретный период. Умет работать самостоятельно, применяя практические умения и навыки. Правильно и по назначению применяет инструменты. Умеет выполнять основные логические действия (анализ, синтез, установление причинно-следственных связей). Способен планировать и регулировать свою деятельность по реализации проекта. Умеет осуществлять поиск информации, в том числе в сети Интернет; выслушивать собеседника и вести диалог; выражать свои мысли в соответствии с задачами и условиями коммуникации.
	Личностные результаты	Обучающийся обладает внутренней мотивацией. Способен самостоятельно организовать собственную деятельность. Сформирована культура работы с информацией. Работу выполняет аккуратно, доводит до конца. Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.
Средний уровень (50-79%)	Теоретические знания	Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу. Использует специальную терминологию, однако сочетает её с бытовой
	Практические умения и навыки	Владеет базовыми навыками и умениями, но не всегда может выполнить задание самостоятельно, просит помощи педагога. В основном выполняет задания на основе образца. Способен разработать проект с помощью преподавателя. Встречаются отдельные случаи неправильного применения инструментов. Делает ошибки в работе, но может устранить их после наводящих вопросов или самостоятельно Испытывает незначительные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей). Способен планировать и регулировать свою деятельность по реализации проекта с помощью педагога. Испытывает незначительные сложности в осуществлении коммуникации с педагогом и сверстниками.
	Личностные	Внутренняя мотивация к обучению сочетается с внешней.

	результаты	В работе допускает небрежность. Работу не всегда выполняет аккуратно и/или доводит до конца. Оценить результаты своей деятельности может с подсказкой педагога.
Низкий уровень (меньше 50%)	Теоретические знания	Владеет минимальными знаниями, ориентируется в содержании материала по темам только с помощью педагога. Избегает употреблять специальные термины.
	Практические умения и навыки	Владеет минимальными начальными навыками и умениями. Учащийся способен выполнять каждую операцию только с подсказкой педагога или товарищей. Часто неправильно применяет необходимый инструмент или не использует его вовсе. В работе допускает грубые ошибки, не может их найти даже после указания преподавателя. В состоянии выполнять лишь простейшие практические задания педагога. Испытывает существенные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей). Не способен планировать и регулировать свою деятельность по реализации проекта. Испытывает значительные сложности в осуществлении коммуникации с педагогом и сверстниками.
	Личностные результаты	Преобладает внешняя мотивация к обучению. Работу часто выполняет неаккуратно и/или не доводит до конца. Не способен самостоятельно и объективно оценить результаты своей работы.

Сводная таблица результатов обучения по программе

№ п/п	Фамилия, имя обучающегося	Оценка теоретических знаний	Оценка практических умений и навыков (предметных и метапредметных)	Личностные результаты	Итоговая оценка
1.					

Анализ диагностической работы

Количество обучающихся по списку		
Количество выполнявших работу		
Количество	«Высокий уровень»	
	«Средний уровень»	
	«Низкий уровень»	
Успеваемость (в %)		
Качество обученности (в %)		

УЧЕБНЫЙ ПЛАН

№ п/п	Название разделов, тем	Количество часов			Форма аттестации/контроля
		Всего	Теория	Практика	
1	Актуализация знаний языка Python	24	4	20	
1.1	Решение задач с использованием стандартных алгоритмических структур	6	1	5	Демонстрация решения
1.2	Решение задач с использованием списков	6	1	5	Демонстрация решения
1.3	Решение задач с использованием строк	6	1	5	Демонстрация решения
1.4	Решение задач с использованием функций	6	1	5	Демонстрация решения
2	Работа с запросами	14	8	6	
2.1	Веб-запросы. Структура HTTP-запроса. Код HTTP-запроса	4	4		Беседа
2.2	Библиотека requests. Составление запросов к веб-приложениям. Получение информации в JSON-файле	4	2	2	
2.3	Получение медиа-файлов через запрос	4	2	2	
2.4	Самостоятельное решение задач по теме работа с запросами	2		2	Демонстрация решений
3	Регулярные выражения	10	4	6	
3.1	Регулярные выражения. Основные метасимволы языка	2	2		Беседа
3.2	Построение регулярных выражений для тестирования входных строк. Стандартные маски	2		2	
3.3	Использование регулярных выражений в языке Python. Библиотека re	4	2	2	
3.4	Самостоятельное решение задач по теме работа с регулярными выражениями	2		2	Демонстрация решений
4	Тестирование	10	5	5	
4.1	Тестирование приложений. Виды тестов. Unit-тесты	2	2		Беседа
4.2	Unit-тесты в стандартной	4	2	2	

	библиотеке Python. Реализация тестов				
4.3	Библиотека pytest	2	1	1	
4.4	Самостоятельное решение задач по теме тестирование	2		2	Демонстрация решений
5	Создание веб-страницы	26	10	16	
5.1	Обзор библиотеки flask	2	2		Беседа
5.2	Шаблоны страниц. Создание шаблона. Синтаксис HTML и CSS	4	2	2	
5.3	Создание стандартной формы страницы	2	1	1	
5.4	Работа с БД. Подключение базы данных. Вывод информации из БД на страницу	6	2	4	
5.5	Реализация авторизации	2	1	1	
5.6	Реализация CRUD-функций	2	2		Наблюдение
5.7	Добавление информации в БД	2		2	
5.8	Редактирование информации в БД	2		2	
5.9	Удаление информации из БД	2		2	
5.10	Самостоятельное решение задач по теме создание веб-страниц	2		2	Демонстрация решений
6	Дополнительные библиотеки	20	9	11	
6.1	Библиотека PyPy. Ускорение работы приложения	4	2	2	Беседа
6.2	Библиотека time. Подсчет времени работы приложения	2	1	1	
6.3	Библиотека hashlib. Шифрование информации	4	2	2	
6.4	Библиотека zipfile. Архивирование информации	4	2	2	
6.5	Библиотека Ruff. Создание линеров	4	2	2	
6.6	Самостоятельное решение задач по теме работа с дополнительными библиотеками	2		2	Демонстрация решений
7	Создание проекта	40	8	32	
7.1	Определение и согласование темы и цели итогового проекта	4	2	2	Беседа
7.2	Планирование работы над итоговым проектом	2	2		
7.3	Дизайн итогового проекта	4		4	
7.4	Создание макетов приложения	6	2	4	

7.5	Программирование логики программы	8		8	Наблюдение
7.6	Тестирование разработанного приложения	4		4	
7.7	Подготовка презентаций	4	2	2	
7.8	Подготовка речи	2		2	
7.9	Подготовка к защите проектов	2		2	Беседа
7.10	Защита итоговых проектов	2		2	Защита проектов
7.11	Доработка проектов	2		2	
	Итого	144	48	96	

СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

Раздел 1. Актуализация знаний языка Python

1.1. Решение задач с использованием стандартных алгоритмических структур

Теория (1 час). Инструктаж о правилах поведения на занятиях и технике безопасности. Вредные и опасные факторы во время работы в компьютерном классе. Повторение основных базовых алгоритмических структур, конструкций и правил работы с ними.

Практика (5 часов). Решение задач по теме «Алгоритмические структуры»

1.2. Решение задач с использованием списков

Теория (1 час). Повторение списков и правил работы с ними.

Практика (5 часов). Решение задач по теме «Списки»

1.3. Решение задач с использованием строк

Теория (1 час). Повторение строк и правил работы с ними.

Практика (5 час). Решение задач по теме «Строки»

1.4. Решение задач с использованием функций

Теория (1 час). Повторение функций и правил работы с ними.

Практика (5 часов). Решение задач по теме «Функции»

Раздел 2. Работа с запросами

2.1. Веб-запросы. Структура HTTP-запроса. Код HTTP-запроса

Теория (4 часа). Структура веб-запросов. Коды возвращаемых значений

2.2. Библиотека requests. Составление запросов к веб-приложениям.

Получение информации в JSON-файле

Теория (2 часа). Знакомство с библиотекой requests. Методы для составления и отправки запросов.

Практика (2 часа). Отправка запроса на WebAPI, получение JSON-ответа

2.3. Получение медиа-файлов через запрос

Теория (2 часа). Различие между GET и POST запросами

Практика (2 часа). Получение аудио и видеофайлов в ответ на запрос

2.4. Самостоятельное решение задач по теме работа с запросами

Практика (2 часа). Самостоятельная демонстрация полученных навыков по работе с веб-запросами

Раздел 3. Регулярные выражения

3.1. Регулярные выражения. Основные метасимволы языка

Теория (2 часа). Регулярные выражения. Метасимволы и их последовательности

3.2. Построение регулярных выражений для тестирования входных строк. Стандартные маски

Практика (2 часа). Создание маски для проверки корректности почты

3.3. Использование регулярных выражений в языке Python.

Библиотека re

Теория (2 часа). Знакомство с библиотекой re, основные методы

Практика (2 часа). Использование регулярных выражений для проверки корректности вводимой информации в приложении

3.4. Самостоятельное решение задач по теме работа с регулярными выражениями

Практика (2 часа). Самостоятельная демонстрация полученных навыков по работе с регулярными выражениями

Раздел 4. Тестирование

4.1. Тестирование приложений. Виды тестов. Unit-тесты

Теория (2 часа). Использование Unit-тестов, для тестирования приложения

4.2. Unit-тесты в стандартной библиотеке Python. Реализация тестов

Теория (2 часа). Обзор библиотеки unittest. Функция Assert

Практика (2 часа). Создание тестов для приложения

4.3. Библиотека pytest

Теория (1 час). Обзор библиотеки pytest

Практика (1 час). Создание тестов для приложения

4.4. Аттестация по пройденным темам

Практика (2 часа). Тестирование по темам предыдущих разделов

Раздел 5. Создание веб-страницы

5.1. Обзор библиотеки flask

Теория (2 часа). Обзор библиотеки flask. Принцип создания веб-приложения

5.2. Шаблоны страниц. Создание шаблона. Синтаксис HTML и CSS

Теория (2 часа). Шаблоны страниц. HTML-теги

Практика (2 часа). Реализация HTML-разметки страницы

5.3. Создание стандартной формы страницы

Теория (1 час). Формы страниц. Частоиспользуемые формы

Практика (1 час). Реализация формы авторизации, личного кабинета, восстановления пароля

5.4. Работа с БД. Подключение базы данных. Вывод информации из БД на страницу

Теория (2 часа). Пример работы с базой данных

Практика (4 часа). Реализация подключения к БД, вывод информации из БД на страницу сайта

5.5. Реализация авторизации

Теория (1 час). Основные практики реализации авторизации в приложении

Практика (1 час). Реализация авторизации на сайте

5.6. Реализация CRUD-функций

Теория (2 часа). Разбор примера создания CRUD-функций

5.7. Добавление информации в БД

Практика (2 часа). Написание INSERT-запросов для добавления информации в базу данных

5.8. Редактирование информации в БД

Практика (2 часа). Написание UPDATE-запросов для редактирования информации в базе данных

5.9. Удаление информации в БД

Практика (2 часа). Написание DELETE-запросов для удаления информации из базы данных

5.10. Самостоятельное решение задач по теме создание веб-страниц

Практика (2 часа). Самостоятельная демонстрация полученных навыков по созданию веб-страниц

Раздел 6. Дополнительные библиотеки

6.1. Библиотека PyPy. Ускорение работы приложения

Теория (2 часа). Обзор библиотеки PyPy. Ускорение выполнения кода

Практика (2 часа). Использование библиотеки PyPy в коде для ускорения приложения

6.2. Библиотека time. Подсчет времени работы приложения

Теория (1 час). Обзор библиотеки time. Проверка времени выполнения

Практика (1 час). Использование библиотеки time в коде для замера скорости выполнения

6.3. Библиотека hashlib. Шифрование информации

Теория (2 часа). Обзор библиотеки hashlib. Шифрование информации

Практика (2 часа). Использование библиотеки hashlib в коде приложения для шифрования информации

6.4. Библиотека zipfile. Архивирование информации

Теория (2 часа). Обзор библиотеки zipfile. Архивирование информации

Практика (2 часа). Использование библиотеки zipfile в коде приложения для архивирования информации

6.5. Библиотека Ruff. Создание линеров

Теория (2 часа). Обзор библиотеки Ruff. Создание линеров

Практика (2 часа). Использование библиотеки Ruff в коде приложения для создания линеров и вывода информации из БД в линер

6.6. Самостоятельное решение задач по теме работа с дополнительными библиотеками

Практика (2 часа). Самостоятельная демонстрация полученных навыков по работе с дополнительными библиотеками

Раздел 7. Создание проекта

7.1. Определение и согласование темы и цели итогового проекта

Теория (2 час). Введение в проектную деятельность. Цели, задачи проекта.

Практика (2 часа). Выбор темы, определение целей и задач проекта.

7.2. Планирование работы над итоговым проектом

Теория (2 час). Этапы работы над проектом. Правила эффективного планирования.

7.3. Дизайн итогового проекта

Практика (4 часа). Создание интуитивно понятного интерфейса приложения.

7.4. Создание макетов приложения

Теория (2 час). Правила создания дружелюбного интерфейса. Правила гармоничного расположения элементов.

Практика (4 часа). Создание макетов приложения.

7.5. Программирование логики программ.

Практика (8 часов). Программирование функционала программ для итоговых проектов.

7.6. Тестирование приложения

Практика (4 часа). Тестирование приложения на наличие ошибок

7.7. Подготовка презентаций

Теория (2 часа). Правила создания презентаций

Практика (2 часа). Подготовка презентации для защиты итогового проекта.

7.8. Подготовка речи

Практика (2 часа). Подготовка речи и контрольного примера для защиты проекта

7.9. Подготовка к защите программы

Практика (2 часа). Проверка работоспособности программного продукта, финальные правки

7.10. Защита итоговых проектов

Практика (2 часа). Защита проектов.

7.11. Доработка проектов

Практика (2 часа). Доработка проектов в соответствии с рекомендациями, полученными обучающимися на защите проектов.

КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

Материально-техническое обеспечение

Комплекс условий реализации программы:

Аппаратное и техническое обеспечение:

а) Рабочее место учащегося

- ноутбук с выходом в сеть Интернет;

б) Рабочее место наставника

- ноутбук с выходом в сеть Интернет;

- технические средства обучения (мультимедийное устройство).

Методическое обеспечение программы

Методы обучения:

- объяснительно-иллюстративный (беседы, объяснения);

- репродуктивный (деятельность обучаемых носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях);

- метод проблемного изложения;

- эвристический (метод обучения заключается в организации активного поиска решения выдвинутых в обучении (или самостоятельно сформулированных) познавательных задач в ходе подготовки и реализации творческих проектов);

- исследовательский.

Педагогические технологии: информационные технологии, проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

В процессе обучения учащиеся работают со средами программирования, поддерживающими язык Python, текстовыми редакторами, тренажерами быстрого набора текста, табличными редакторами.

Проектная технология дает возможность самостоятельно конструировать свои знания, ориентироваться в информационном пространстве, развивает критическое и творческое мышление, создаёт условия для формирования и

развития внутренней мотивации учащихся к более качественному овладению знаниями, повышения мыслительной активности и приобретения навыков логического мышления.

Здоровьесберегающие технологии позволяют создать максимально возможные условия для сохранения, укрепления и развития эмоционального, интеллектуального и физического здоровья, в том числе в условиях работы с компьютерной техникой.

Проблемное обучение — это тип развивающего обучения, содержание которого представлено системой проблемных задач различного уровня сложности, в процессе решения которых учащиеся овладевают новыми знаниями и способами действия, а через это происходит формирование творческих способностей: продуктивного мышления и познавательной мотивации.

Учебно-методические средства обучения

Для реализации программы используется:

- специализированная литература по направлению, подборка журналов,
- образцы программ и систем, выполненные учащимися и педагогом,
- учебно-методические пособия для педагога и учащихся, включающие дидактический, информационный, справочный материалы на различных носителях, компьютерное и видео оборудование.

Применяемое на занятиях дидактическое и учебно-методическое обеспечение включает в себя электронные учебники, справочные материалы и системы используемых Программ, Интернет, рабочие тетради учащихся.

Кадровое обеспечение

Программу реализуют педагоги дополнительного образования «IT-куб».

Информационное обеспечение

Для реализации программы планируется использование следующих информационных ресурсов:

- курс «Программирование на Python» (<https://stepik.org/course/67>);
- курс «Алгоритмы: теория и практика. Методы» (<https://stepik.org/course/217>).

СПИСОК ЛИТЕРАТУРЫ

Список литературы для педагогов

1. Бейдер Д. Чистый Python. Тонкости программирования для профи. — Питер, 2019. — 228 с.
2. Мэтиз Э. Изучаем Python. 3-издание. — СПб.: Питер, 2021 – 511 с.
3. Ромальо Л. Python. К вершинам мастерства. Лаконичное и эффективное программирование. Второе издание. — ДМК Пресс, 2022. — 898 с.
4. Свейгарт Эл. Автоматизация рутинных задач с помощью Python. Практическое руководство для начинающих. – Букнистика, 2021 – 672 с.
5. Стивенс Р. Алгоритмы. Теория и практическое применение. – М.: Эксмо, 2022 – 547 с.
6. Фёдоров Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата. — М.: Издательство Юрайт, 2019. — 161 с.

Список литературы для учащихся

1. Гэддис Т. Начинаем программировать на Python / пер. с англ. 4-е изд. — СПб.: БХВ-Петербург, 2019. — 768 с.
2. Доусон М. Програмируем на Python. – СПб.: Питер, 2022. – 416 с.
3. Имран А. 40 алгоритмов, которые должен знать каждый программист на Pythonю. - СПб.: Питер, 2023 – 368 с.
4. Луридас П. Алгоритмы для начинающих: теория и практика для разработчика. — М. : Эксмо, 2018. — 608 с.
5. Мюллер Дж. Python для чайников. — СПб.: Диалектика, 2019. — 416 с.

Примерные задания для промежуточной аттестации

1. Создайте запрос для получения погоды на завтрашний день с сайта Яндекс.Погода (5 баллов).
2. Создайте запрос для получения медиа-файла с сайта Python.org (20 баллов).
3. Создайте регулярное выражение для проверки целостности электронной почты (10 баллов).
4. Создайте регулярное выражение для проверки маркировки ноутбука (15 баллов).
5. Создайте тест, который будет проверять регулярное выражение (5 баллов).

Примерные задания для итоговой аттестации

1. Веб-страница с расписанием занятий и темами ЦЦОД It-Куб.
2. Сайт-агрегатор.
3. Система шифрования и архивирования файлов.
4. Система рейтинга обучающихся ЦЦОД It-Куб.
5. Модуль проверки личной информации абитуриентов ЦЦОД It-Куб.
6. Чат-бот, собирающий информацию о интересующих темах для обучения ЦЦОД It-Куб.
7. Интернет-магазин с возможностью заказа из чат-бота.
8. Парсер.
9. Веб-сайт по ведению проекта.
10. Сайт-портфолио обучающегося ЦЦОД «IT-куб».