

Государственное бюджетное профессиональное образовательное учреждение
Псковской области «Опочецкий индустриально-педагогический колледж»
Центр цифрового образования детей «IT – куб»

СОГЛАСОВАНО
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 для детей
«Программирование на Python»

Уровень: базовый

Направленность: техническая

Возраст обучающихся: 15-17 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.....	15
КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ.....	21
СПИСОК ЛИТЕРАТУРЫ	24
Приложение 1	25
Приложение 2	26

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.

Новизна данной программы заключается в том, что учащиеся продолжают изучение современного универсального языка Python и применяют полученные навыки, решая задачи, связанные с построением программ в рамках объектно-ориентированной парадигмы программирования и взаимодействием с базами данных. В процессе реализации данной программы учащиеся изучат основы объектно-ориентированного программирования и познакомятся с базами данных, визуализацией информации, научатся использовать профессиональные средства разработки программного обеспечения, такие как интегрированные системы разработки.

Педагогическая целесообразность программы определяется образовательным замыслом. В процессе освоения программы обучающиеся закрепят полученные на стартовом уровне знания и получат новые знания необходимые к переходу к прикладному программированию.

Направленность программы: техническая.

3. Адресат программы.

Адресатом программы являются дети в возрасте от 15 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.

4. Срок реализации программы.

Срок реализации программы составляет 1 год (36 часов).

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 12 до 15 человек.

6. Объём программы и режим работы.

Объём программы: 36 часов.

Режим занятий: 1 раза в неделю по 1 академическому часу.

Продолжительность часа – 40 минут.

7. Цель программы:

Целью программы является создание условий для изучения основных парадигм программирования и внедрения библиотек для работы с базами данных и визуализацией информации.

8. Задачи программы.

Программа направлена на решение следующих задач:

- изучение принципов и методов объектно-ориентированного программирования;
- изучение базовых и сложных структур данных;
- изучение способов организаций функций;
- изучение основ реляционных баз данных;
- изучение способов визуализации информации;
- формирование навыков работы с базами данных;
- развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- способствовать формированию познавательных и регулятивных универсальных учебных действий;
- 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;
- 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 объектно-ориентированного программирования;
- базовые и сложные структуры данных;
- способы организации функций в языке программирования Python.

Уметь:

- разрабатывать программы для решения задач в рамках ООП;
- использовать инструменты визуализации;
- разрабатывать приложения для работы с реляционными базами данных;
- 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 познавательных и регулятивных универсальных учебных действий;

- 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 тестирования.

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 (Приложение 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 тестирования осуществляется по следующим уровням:

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

Средний уровень – учащийся набрал от 20 до 31 баллов 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 деятельности. 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 выполнивших работу		
Количество	«Высокий уровень»	
	«Средний уровень»	
	«Низкий уровень»	
Успеваемость (в %)		
Качество обученности (в %)		

УЧЕБНЫЙ ПЛАН

№	Название темы	Количество часов			Итоговые формы контроля (аттестации)
		Всего	Теория	Практика	
1	Функции	4	2	2	Опрос
2	Классы. Наследование. Импортирование	4	2	2	зачётная практическая работа
3	Файлы. Исключения	2	1	1	зачётная практическая работа
4	Тестирование	2	1	1	зачётная практическая работа
5	Планирование проекта. Модуль PyGame. Рефакторинг	4	2	2	зачётная практическая работа
6	Анализ проектов	5	2	3	зачётная практическая работа
7	Проработка интерфейса в PyGame	1	1	0	зачётная практическая работа
8	Визуализация данных в Python. Работа с matplotlib и plotly	1	1	0	зачётная практическая работа
9	Загрузка данных и работа с CSV и JSON	5	3	2	зачётная практическая работа
10	Работа с API веб-приложений	5	1	4	зачётная практическая работа
11	Интеграция взаимодействия с API в собственное приложение	3	1	2	зачётная практическая работа
	ИТОГО:	36	17	19	

СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

1. Функции

Теоретическая часть. Определение функции. Передача аргументов. Передачи информации функции.

Практическая часть. Написание функции «display message ()». Написание функции, которая выводит сообщение вида «One of my favorite books»

2. Массы. Наследование. Импортное

Теоретическая часть. Создание и использование класса. Создание класса Dog. Обращение к атрибутам. Создание нескольких экземпляров. Работа с классами и экземплярами. Метод init () класса-потомка. Определение атрибутов и методов класса-потомка. Экземпляры как атрибуты. Импортное одного класса. Хранение нескольких классов в модуле. Импортное нескольких классов из модуля. Импортное модуля в модуль. Использование псевдонимов.

Практическая часть. Напишите класс с именем Admin, наследующий от класса User. Написать программу, которая использует окончательную версию программы electric car.py. Создайте отдельный файл, импортующий класс Restaurant. Написать цикл, который проверяет, насколько сложно выиграть в смоделированной вами лотерее.

3. Файлы. Исключения.

Теоретическая часть. Чтение из файла. Чтение всего файла. Пути к файлам. Создание списка строк по содержимому файла. Работа с содержимым файла. Запись в пустой файл. Присоединение данных к файлу. Обработка исключения ZeroDivisionError. Блоки try-except. Использование исключений для предотвращения аварийного завершения программы. Ошибки без уведомления пользователя. Сохранение данных.

Практическая часть. Написать программу, которая читает файл и выводит текст три раза. Написать программу, которая запрашивает у пользователя его имя. Введенный ответ сохраняется в файле с именем guest.txt. Напишите программу, которая читает файлы из проекта «Гутенберг» и определяет количество вхождений слова 'the' в каждом тексте.

4. Тестирование

Теоретическая часть. Тестирование функции. Модульные тесты и тестовые сценарии. Прохождение теста. Сбой теста. Тестирование класса. Класс для тестирования. Тестирование класса AnonymousSurvey. Метод setUp().

Практическая часть. Написать функцию, которая получает два параметра: название страны и название города. Написать класс Employee, представляющий работника.

5. Планирование проекта. Модуль PyGame. Рефакторинг

Теоретическая часть. Планирование проекта. Установка Pygame. Создание окна Pygame и обработка ввода. Назначение цвета фона. Создание класса Ship. Вывод корабля на экран. Обработка нажатия клавиши. Непрерывное перемещение. Перемещение влево и вправо.

Практическая часть. Создайте класс, который рисует персонажа в центре экрана, и приведите цвет фона изображения в соответствие с цветом фона экрана (или наоборот). Решение задач.

6. Анализ проектов и перевод в классы

Теоретическая часть. Создание пришельца. Создание экземпляра Alien. Построение флота. Добавление рядов. Перемещение флота. Перемещение вправо.

Практическая часть. Найти функции и методы, которые решают более одной задачи, и проведите рефакторинг, улучшающий структуру и эффективность кода. Разработать приложение с боковой стрельбой, используя все, чему вы научились в этом проекте.

7. Проработка интерфейса в PyGame

Теоретическая часть. Распознавание коллизий. Программирование игрового меню. Создание кнопок. Создание меню.

Практическая часть. Создать встроенное меню с собственной поверхностью, которое рендерит свои подкомпоненты. Решение задач.

8. *Визуализация данных в Python. Работа с matplotlib и plotly. Теоретическая часть.* Matplotlib. Диаграмма рассеяния. Графики.

Гистограммы. Plotly.

Практическая часть. Написать программу, которая распределяет переменные в данных.

9. *Загрузка данных и работа с CSV и JSON*

Теоретическая часть. Формат CSV. Разбор заголовка файлов CSV. Печать заголовков и их позиций. Извлечение и чтение данных. Цветовое выделение части диаграммы. Загрузка собственных данных. Построение карт с глобальными наборами данных: формат JSON. Знакомство с данными. Построение карты мира. Другой способ определения данных для диаграммы.

Практическая часть. Написать программу, добавляющую подсказки, которая будет появляться при наведении указателя мыши на маркер. Решение задач.

10. *Работа с API веб-приложений*

Теоретическая часть. Использование API веб-приложений. Запрос данных с использованием вызовов API. Обработка ответа API. Работа со словарем ответа. Проверка ограничений частоты обращений API. Визуализация репозитория с использованием Plotly. Подробнее о Plotly и GitHub API.

Практическая часть. Доработка диаграмм Plotly. Добавление подсказок. Добавление активных ссылок на диаграмму. Напишите программу `test_ython_repos.py`, которая использует модуль `unittest` для проверки того, что значение `status code` равно 200. Решение задач.

11. *Интеграция взаимодействия с API в собственное приложение*

Теоретическая часть. Определение и принцип работы. Импорт и экспорт данных. Поиск товаров и услуг.

Практическая часть. Работа с Weather API. Решение задач.

№ п/п	Форма занятия	Кол- во часов	Тема занятия	Форма контроля
Функции 4 часа				
1	теория	2	Определение функции. Передача аргументов. Передачи информации функции.	Опрос
2	практика	1	Написание функции «display message ()».	Практическая работа
	практика	1	Написание функции, которая выводит сообщение вида «One of my favorite books»	
Классы. Наследование. Импортрование 4 часа				
3	теория	1	Создание и использование классов	Опрос
4	теория	1	Создание нескольких классов, работа с экземплярами класса.	Опрос
5	практика	1	Создание класса Admin, наследующегося от класса User.	практическая работа
6	практика	1	Написать цикл, который проверяет, насколько сложно выиграть в смоделированной вами лотерее.	практическая работа
Файлы. Исключения 2 часа				
7	теория	1	Чтение из файла, работа с содержимым файла. Обработка исключений.	Опрос
8	практика	1	Вывод содержимого файла	Практическая работа
Тестирование 2 часа				
9	теория	1	Тестирование функции. Модульные тесты и тестовые сценарии.	Опрос
10	практика	1	Написать класс Employee, представляющий работника.	практическая работа
Планирование проекта. Модуль PyGame. Рефакторинг 4 часа				
11	теория	2	Планирование проекта. Установка Pygame. Создание окна Pygame и обработка ввода.	Опрос
12	практика	2	Создание класса, рисующего персонажа в центре экрана	практическая работа
Анализ проектов 5 часов				
13	теория	2	Создание пришельца. Создание экземпляра Alien. Построение флота. Добавление рядов. Перемещение флота. Перемещение вправо.	Опрос

14	практика	2	Найти функции и методы, которые решают более одной задачи, и проведите рефакторинг, улучшающий структуру и эффективность кода.	практическая работа
15	практика	1	Разработать приложение с боковой стрельбой, используя все, чему вы научились в этом проекте.	практическая работа
Проработка интерфейса в PyGame 1 час				
16	теория	1	Распознавание коллизий. Программирование игрового меню. Создание кнопок. Создание меню..	Опрос
Визуализация данных в Python. Работа с matplotlib и plotly 1 час				
17	теория	1	Matplotlib. Диаграмма рассеяния. Графики. Гистограммы. Plotly.	Опрос
Загрузка данных и работа с CSV и JSON 5 часов				
18	теория	2	Формат CSV. Разбор заголовка файлов CSV. Печать заголовков и их позиций.	Опрос
19	теория	1	Извлечение и чтение данных. Цветовое выделение части диаграммы.	Опрос
20	практика	2	Написать программу, добавляющую подсказки, которая будет появляться при наведении указателя мыши на маркер. Решение задач.	практическая работа
Работа с API веб-приложений 5 часов				
21	теория	1	Использование API веб-приложений. Запрос данных с использованием вызовов API. Обработка ответа API. Работа со словарем ответа. Проверка ограничений частоты обращений API. Визуализация репозитория с использованием Plotly. Подробнее о Plotly и GitHub API.	Опрос

22	практика	2	Доработка диаграмм Plotly. Добавление подсказок. Добавление активных ссылок на диаграмму.	практическая работа
24	практика	2	Напишите программу test ython repos.py, которая использует модуль unittest ДЛЯ проверки того, что значение status code равно 200.	практическая работа
Интеграция взаимодействия с API в собственное приложение 3 часа				
25	теория	1	Определение и принцип работы. Импорт и экспорт данных. ПОИСК товаров и услуг.	Опрос
26	практика	2	Работа с Weather API	практическая работа

1. Формы аттестации и оценочные материалы

Предполагаются следующие виды контроля:

Входной контроль: в начале проводится тестирование для определения уровня развития учеников и их готовности к программе курса.

Текущий контроль: в процессе прохождения раздела, путём наблюдения, опроса, решения задач, выполнения творческих заданий, выступлений отслеживаются результаты учащихся.

Итоговый контроль: завершающим этапом является презентация проектов обучающихся.

2. Организационно-педагогические условия реализации программы

а. Материально-техническое обеспечение программы

Помещение должно быть оформлено в соответствии с санитарными нормами для ведения занятий в группах не менее 10 человек. В нем должны быть столы и стулья для преподавателя и учащихся (не менее 10 человек),

проекционный экран для демонстрации материалов, персональные компьютеры с устройствами ввода (либо ноутбуки с мышками), полное программное обеспечение (операционная система, текстовый редактор) для педагога и каждого ученика. Помимо этого, необходим шкаф учебных материалов (книг, журналов и т.п.), должен быть обеспечен доступ к электросети (в виде розеток и сетевых распределителей), к сети интернет (через Ethernet-кабель либо Wi-Fi).

С первых дней занятий учащиеся должны быть ознакомлены с правилами техники безопасности.

в. Перечень дидактических материалов

Для обеспечения наглядности в ходе курсов будут использованы пособия, доступ к которым можно получить через интернет: видео, изображения, таблицы, схемы, инструкции к моделям.

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 с 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 скрипт, который создает директорию с вашим именем (5 баллов).
2. Создайте скрипт, который выводит в CSV-файл информацию о пяти пользователях. За информацию о пользователе следует взять: ФИО, дата рождения, мобильный телефон и email (5 баллов).
3. Создать класс, реализующий кулинарное блюдо. Сделать поля приватными, а работу с ними через геттеры и сеттеры (10 баллов).
4. Создать класс работника, от которого будут наследоваться бухгалтер, разнорабочий и строитель. Инициализировать поля дочерних классов через родительский (15 баллов).
5. Создайте приложение, которое будет содержать информацию о цветах в саду. Информацию вводит пользователь. Информация хранится в списке классов. Информация должна выводиться в CSV-файл (20 баллов).

Примерные задания для итоговой аттестации

1. Визуализация информации о подземных толчках из базы данных.
2. Чат-бота, формирующий характеристику личности пользователя.
3. Учет заявок на поступление в ЦЦОД ИТ -Куб.
4. Файловый менеджер.
5. Калькулятор.
6. Графический редактор.
7. Учет посещаемости группы ЦЦОД ИТ -Куб.
8. Викторина.
9. Чат-бот, оповещающий о занятиях.
10. Визуализация информации об успеваемости группы ЦЦОД ИТ-Куб.