

СОДЕРЖАНИЕ

1. Пояснительная записка.....	3
2. Новизна данной образовательной программы.....	4
3. Общая характеристика программы	8
4. Результаты освоения программы	10
5. Тематический план.....	10
6. Руководитель и составитель программы	11
7. Список литературы	11

1. Пояснительная записка

Актуальность и направленность программы. В настоящее время прослеживается острый кризис квалифицированных кадров в различных сферах рынка труда. Работодатели, хоть и обращают внимание на область предметных знаний, но все же отдают предпочтения универсальным навыкам. Подобная позиция связана с постоянно изменяющимися условиями труда, технологическим прогрессом. Подобные универсальные навыки (способность работать в команде, многозадачность, творческий подход, критическое мышление, продуктивное мышление, нацеленность на результат) вырабатываются в процессе профессионального становления.

Характерной чертой развития общества на протяжении последних десятилетий является его все более расширяющаяся информатизация. Отражением и следствием этой тенденции явилась потребность в подготовке подрастающего поколения к вступлению в информационное общество, любая профессиональная деятельность в котором будет связана с информатикой и информационными технологиями. Изучение данного курса имеет важное значение для развития мышления подростков. В современной психологии отмечается значительное влияние изучения информатики и использования компьютеров в обучении на развитие у школьников теоретического, творческого мышления, а также формирование нового типа мышления, так называемого операционного мышления, направленного на выбор оптимальных решений; открывает новые возможности для овладения такими современными методами научного познания, как формализация, моделирование, компьютерный эксперимент и т.д. Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и некоторые общие мыслительные навыки, полезные и в более широком контексте.

Направленность дополнительной образовательной программы — техническая. Она заключается в развитии технического и творческого мышления у обучающихся среднего школьного возраста через разработку алгоритмов в цифровой среде, взаимодействие с приложениями, сервисами и инструментами вне зависимости от платформы или интерфейса, формирование навыка определения различных видов ошибок (логическая, синтаксическая, ошибка среды и ошибка взаимодействия) и принципах их исправления в процессе разработки с помощью процедур отладки. Также развитие у детей умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, формирование навыков сотрудничества и коллективной работы в цифровой среде.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного мышления и навыков проектной деятельности.

Python – это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. ...

Программа «Основы программирование на Python» представляет собой начальный курс по программированию, дающий представление о понятиях структурного программирования (данных, операциях, переменных, ветвлениях в программе, циклах и функциях, массивах, файлах). Программа направлена на развитие логического и пространственного мышления обучающегося, способствует раскрытию творческого потенциала личности, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Курс служит средством внутрипрофильной специализации в области новых

информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

По форме организации содержания и процесса педагогической деятельности программа «Основы программирования на Python» интегрированная, так как объединяет в целое отдельные образовательные области на основе единства математики и информатики. Главной целью является приобретение обучающимся теоретических знаний и практических навыков и компетенций, достаточных для дальнейшего успешного обучения в системе непрерывного образования: школа – допобразование – ВУЗ – предприятие. Сущностью программы можно определить как развитие мышления и формирование мировоззрения в условиях преподавания интегрированного курса информатики, математики, программирования обучающихся среднего и старшего звена. Изучение всех этих предметов должно быть не целью, а средством познания мира, давать возможность обучающимся проникать в сущность изучаемых проблем. В теории интеграции в данном случае вкладывается понятие взаимосвязи, взаимопроникновения математики и информатики, что предполагает качественное изменение в параметрах нового объекта. В школьном курсе информатики вопросы программирования рассматриваются лишь в ознакомительном плане и на это выделяется недостаточное количество часов, как следствие – формальное восприятие обучающимися основ современного программирования. Образовательная программа «Основы программирования на Python» направлена на устранение данного пробела.

Программа рассчитана на 24 учебных (практических) часа и предназначен для обучающихся, получающих среднее (полное) общее образование

2. Новизна данной образовательной программы

Отличительной особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной практической задачи, представленную в виде адаптированного кейса из реальной жизни, составления алгоритма решения и его реализации с помощью средств программирования. Для этого, учащиеся проходят через следующие этапы: определяют и составляют последовательность действий, ведущую к цели, и записывают ее с помощью формального языка. Полученные знания учащиеся применяют при создании собственных проектов, которые защищаются перед другими учениками, педагогами и родителями.

Использование компьютерных технологий в работе с детьми среднего школьного возраста является стремительно развивающейся методикой в образовании во всем мире. С ее помощью можно более эффективно решать образовательные задачи, которые будут способствовать качественному улучшению обучения ребенка в школе.

Адресат программы. Программа ориентирована на дополнительное образование обучающихся 13-16 лет, интересующихся программированием.

Срок и этапы реализации программы. Программа рассчитана на 2 месяца обучения - 24 академических часа.

Основной формой обучения являются групповые занятия. В основе образовательного процесса лежит проектный подход.

Режим занятий: 2 раза по 2 часа в неделю.

Продолжительность 1 занятия: 2 академических часа.

Структура двухчасового занятия:

- 40 минут – рабочая часть;
- 10 минут – перерыв (отдых);
- 40 минут – рабочая часть.

Основная форма работы теоретической части – лекционные занятия в группах до 15 человек. Практические задания планируется выполнять индивидуально, в парах и в малых группах. Занятия проводятся в виде бесед, семинаров, лекций: для наглядности изучаемого материала используется различный мультимедийный материал – презентации, видеоролики.

Цель программы

Целью курса – способствовать формированию алгоритмического и критического мышления, навыков проектной деятельности через использование языка программирования Python как инструмента для реализации идей в области своих интересов.

Задачи программы

Достижение поставленной цели складывается из выполнения следующих задач:

Обучающие:

1. Познакомить с базовыми понятиями основ программирования.
2. Познакомить с особенностями синтаксиса языка программирования Python.
3. Познакомить с принципами объектно-ориентированного программирования.
4. Сформировать навык правильного оформления кода.
5. Сформировать навык использования итеративного подхода при решении различных задач.
6. Формировать навык решения задач и разработки проектов с помощью Python.
7. Формировать навык разработки игр с помощью языка программирования Python.
8. Формировать навык тестирования и оптимизации созданных проектов.

Развивающие:

1. Формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление.
2. Формировать навык публичного выступления и презентации.
3. Формировать навык планирования деятельности и основ тайм-менеджмента.
4. Формировать и расширять словарный запас на английском языке.
5. Формировать функциональную грамотность.
6. Расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.
7. Совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.

Воспитательные:

1. Воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развивать эмоциональный интеллект.
2. Формировать умение давать качественную обратную связь и реагировать на нее.
3. Формировать информационную культуру.

В результате освоения дисциплины обучающийся должен:

уметь:

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- настраивать рабочую среду Python;

знать:

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;

- принципы объектно-ориентированного программирования;
- основы создания приложений в Python;
- инструментальные средства Python.

Методы: кейс-метод, проектная деятельность.

Формы работы:

- практическое занятие;
- занятие – соревнование;
- консультация;
- самостоятельная работа.

Практические занятия составляют важную часть теоретической и профессиональной подготовки. Они направлены на формирование практических навыков и умений. Практические занятия создают оптимальные дидактические условия для деятельностного освоения обучающимися содержания и методологии изучаемой дисциплины, использование специального оборудования, технических средств. Ведущей дидактической целью практических занятий является формирование практических умений – профессиональных (выполнять определённые действия, операции, необходимые в последующем в профессиональной деятельности).

Целью занятия-соревнования является создание условий для развития интеллекта и проявления способности к творчеству. Задачами такого занятия могут быть - повышение интереса к данному курсу в частности и познанию в общем, формирование навыка принимать необычные решения. Занятие-соревнование отличается от «традиционного» тем, что он учит выполнению работы по предмету, использованию теории на практике, коллективной деятельности, делать выводы.

Консультации являются одной из форм руководства самостоятельной работой обучающихся и оказания им помощи в освоении учебного материала. Они могут проводиться регулярно или внепланово по мере необходимости. Консультации помогают обучающемуся избрать правильные методы работы, ни в коей мере не снижая его ответственности за принятые решения. Часто применяются при работе обучающегося над итоговым проектом.

Самостоятельная работа обучающихся – это разнообразные виды деятельности обучающихся, осуществляемые под руководством, но без непосредственного участия педагога в специально отведенное для этого аудиторное или внеаудиторное время. Это особая форма обучения по заданиям педагога, выполнение которых требует активной мыслительной, поисково – исследовательской и аналитической деятельности. Методологическую основу самостоятельной работы обучающихся составляет деятельностный подход, когда цели обучения ориентированы на формирование умений решать типовые и нетиповые задачи, т. е. на реальные ситуации, где обучающимся надо проявить знание конкретной дисциплины, использовать внутрпредметные и межпредметные связи.

Виды учебной деятельности:

- анализ проблемных учебных ситуаций;
- построение гипотезы на основе анализа имеющихся данных;
- поиск необходимой информации;
- выполнение практических работ;
- подготовка выступлений и докладов с использованием разнообразных источников информации;
- публичное выступление.

Формы подведения итогов реализации программы

Основной формой подведения итогов дополнительной общеразвивающей программы «Основы программирования на Python» является решение задач, проектная деятельность (разработка и отладка программ на языке программирования Python).

Критерии оценки защиты проекта:

Критерии Ф.И.О. обучающегося, тема проекта		
<p>Постановка цели, проблематизация:</p> <p>1. Проектная работа соответствует цели и отвечает на проблемные вопросы – 3 балла</p> <p>2. Проектная работа соответствует цели и отвечает на некоторые проблемные вопросы – 2 балла</p> <p>3. Проектная работа не совсем точно отражает цель проекта и его проблемные вопросы – 1 балл</p>		
<p>Формулировка задач проекта:</p> <p>1. Поставленные задачи ведут к достижению цели проекта – 3 балла</p> <p>2. Не все задачи ведут к достижению цели проекта – 2 балла</p> <p>3. Представленные задачи не ведут к достижению цели проекта – 1 балл</p>		
<p>Результаты работы:</p> <p>1. Результаты работы, представленные при помощи компьютерных средств, оформлены в соответствие с правилами– 3 балла</p> <p>2. Результаты работы, представленные при помощи компьютерных средств, содержат незначительные ошибки в оформлении – 2 балла</p> <p>3. Результаты работы, представленные при помощи компьютерных средств, содержат значительные ошибки в оформлении – 1 балл</p>		
<p>Выступление:</p> <p>1. Устное выступление участника логично, отсутствуют грамматические и лексические ошибки – 3 балла</p> <p>2. Устное выступление участника логично, присутствуют незначительные грамматические и лексические ошибки, не мешающие пониманию материала – 2 балла</p> <p>3. Устное выступление участника не всегда логично, присутствуют грамматические и лексические ошибки, которые затрудняют понимание – 1 балл</p>		
<p>Соответствие выступления и презентации:</p> <p>1. Выступление не повторяет текст презентации или публикации – 3 балла</p> <p>2. Выступление частично повторяет текст презентации или публикации – 2 балла</p> <p>3. Выступление полностью повторяет текст презентации или публикации – 1 балл</p>		
<p>Ответы на вопросы:</p> <p>1. В ходе устного выступления даны ответы на все вопросы – 3 балла</p>		

2. В ходе устного выступления даны ответы на некоторые вопросы – 2 балла		
3. Обучающийся затруднялся давать правильные ответы на вопросы – 1 балл		

3. Общая характеристика программы

3.1 Основные разделы программы

Инструкции и структура программы.

Техника безопасности. История создания. Области применения и перспективы. Инструкции и структура программы. Основы Алгоритмизации. Достоинства языка.

Установка Python. Ввод и вывод данных.

Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы Алгоритмизации.

Практическая работа: Решение задач для развития логики и понимания работы алгоритма. Установка программы Python. Написание первой программы вывод на экран.

Типы данных и операции. Переменные. Операторы.

Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран.

Практическая работа: Решение задач на элементарные действия с числами. Создание программы простейший калькулятор.

Типы данных. Базовые функции для работы с различными типами данных.

Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)

Практическая работа: Решение математических задач с использованием функции import math.

Инструкции и синтаксис Условные операторы if, elif, else.

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа: Практическое закрепление знаний по условным операторам. Создание программ.

Циклы while, for. Обработка исключений. Случайные числа.

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа: Числа Фибоначчи. Решение задачи с циклом for. Создание игры угадай число. Повторение пройденного.

Функции и модули в программировании Встроенные и пользовательские функции.

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала.

Практическая работа: Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций.

Строки - последовательности символов.

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа: Написание программ. Работа со строками.

Сложные типы данных. Списки. Срезы списков.

Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Практическая работа: Написание программ. Работа со списками.

Кортежи

Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

Словари

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

Написание и отладка программ. Самостоятельная работа

Практическая работа: Стиль программирования. Отладка программ. Проектная работа по курсу «Основы программирования на языке Python».

3.2 Формы организации учебных занятий

Основной тип занятий - комбинированный, сочетающий в себе элементы теории и практики.

Каждая тема программы сопровождается наглядной демонстрацией приемов работы в среде трехмерного моделирования. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися.

Выполнение практических заданий способствует активизации учебно-познавательной деятельности и ведет к закреплению знаний, а также служит индикатором успешности образовательного процесса.

Методы организации учебного процесса

При организации занятий по программе для достижения поставленных целей и решения поставленных задач используются формы проведения занятий с активными методами обучения:

- занятие в форме демонстрации;
- занятие с использованием информационно-коммуникационных технологий;
- занятие с использованием межпредметных связей;
- занятие в форме частично регламентированного упражнения.

Формы и методы контроля:

- устный опрос;
- практические работы;
- участие в проектной деятельности.

Общая характеристика учебного процесса:

- при изучении программы используются практические работы;
- программа не предусматривает сдачу итогового экзамена.

4. Результаты освоения программы

В результате освоения данной программы обучающийся должен:

- уметь Установка Python использовать документацию;
- знать основные инструкции и синтаксис языка
- уметь создавать функции, словари, кортежи, строки
- выполнять построение и редактирование сборочных чертежей;
- уметь писать и отлаживать код программ.

5. Тематический план

Тип	Тема	Часы
Комбинированное занятие	Инструкции и структура программы.	2
Комбинированное занятие	Установка Python. Ввод и вывод данных.	2
Комбинированное занятие	Типы данных и операции. Переменные. Операторы.	2
Комбинированное занятие	Типы данных. Базовые функции для работы с различными типами данных.	2
Комбинированное занятие	Инструкции и синтаксис Условные операторы if, elif, else.	2
Комбинированное занятие	Циклы while, for. Обработка исключений. Случайные числа.	2
Комбинированное занятие	Функции и модули в программировании Встроенные и пользовательские функции.	2
Комбинированное занятие	Строки - последовательности символов.	2
Комбинированное занятие	Кортежи	2
Комбинированное занятие	Словари	2
Комбинированное занятие	Написание и отладка программ. Самостоятельная работа	4

6. Руководитель и составитель программы

Составитель программы:

Руководитель программы: А.В.Ежова, преподаватель.

Составитель программы: А.В.Ежова, преподаватель.

7. Список литературы

Основная:

1. Клейнберг Дж. Алгоритмы: разработка и применение. СПб: Питер, 2016. - 800 с.
2. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.: ил.
3. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. - 288 с.

Дополнительная:

1. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.
2. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. М: Ямб, 2012. – 520 с.
4. Златопольский Д.М. Сборник задач по программированию. СПб: БХВ-Петербург, 2011. - 295 с.
5. ООП на Python: концепции, принципы и примеры реализации [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/python-oop/>