

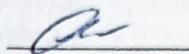
Муниципальное общеобразовательное учреждение – средняя общеобразовательная школа №3 города Аткарска Саратовской области имени Героя Советского Союза Антонова Владимира Семеновича

Центр образования естественно-научной и технологической направленностей «Точка роста»

Принята
Педагогическим советом
МОУ-СОШ №3 г. Аткарска
Протокол №2 от 26.09.2024 г.

СОГЛАСОВАНА

Заместителем директора по УВ



/И.В. Осауленко

27.09.2024 г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON»
с использованием оборудования центра «Точка роста»**

Направленность: техническая

Возраст обучающихся: 15 - 18 лет

Срок реализации программы: 2024 - 2025 и 2025-2026 учебные года

Автор-составитель: педагог дополнительного образования, учитель информатики и математики высшей квалификационной категории

Бражникова Ольга Валентиновна

г. Аткарска, 2024 г.

1. Комплекс основных характеристик программы

Пояснительная записка

Дополнительное образование является составной частью образовательного процесса и одной из форм организации свободного времени обучающихся. Дополнительная общеобразовательная общеразвивающая программа «Основы программирования на языке Python» имеет техническую направленность и разработана на основе положения о дополнительной общеобразовательной общеразвивающей программе МОУ-СОШ №3 г. Аткарска Саратовской области.

Программа «Основы программирования на языке Python» направлена на подготовку творческой, технически грамотной, гармонично развитой личности, обладающей логическим мышлением, способной анализировать и решать задачи в команде в области информационных технологий, решать ситуационные кейсовые задания, основанные на групповых проектах. Занятия по данному курсу рассчитаны на общенаучную подготовку обучающихся, развитие их мышления, логики, математических способностей, исследовательских навыков. Курс направлен на изучение основ программирования на языке Python. В рамках курса обучающиеся смогут познакомиться с физическими, техническими и математическими понятиями. Приобретенные знания будут применимы в творческих проектах.

Направленность программы

Направленность программы: техническая. Программа направлена на привлечение учащихся к современным технологиям программирования и развитию способностей к техническому творчеству, техническому моделированию.

Актуальность программы

Программа направлена на привлечение учащихся к современным технологиям программирования. С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать. Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями. Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

Потребностью общества в специалистах, владеющих профессиональными навыками и умением. Определение и выбор учащихся (ещё на стадии школьного обучения) дальнейшего профессионального развития, обучения и освоения конкретных специальностей; - более лёгкой адаптацией «во взрослой» жизни.

Актуальность программы проявляется в том, что на современном этапе развития общества она отвечает запросам детей и родителей: формирует социально значимые знания, умения и навыки оказывает комплексное обучающее, развивающее, воспитательное и здоровьесберегающее воздействие, способствует формированию эстетических и нравственных качеств личности, приобщает детей к творчеству.

Педагогическая целесообразность

Педагогическая целесообразность состоит в том, что активизация познавательного процесса позволяет учащимся более полно выразить свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутри профильной специализации в области новых

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

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Целесообразность программы выражена в подборе интерактивных и практико-ориентированных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития и др.).

В рамках изучения курса программы «Основы программирования на языке Python» – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков. Особую роль в программировании отводится формированию мыслительных и психических процессов учащихся (внимание, память, логика), освоение приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования предметных и метапредметных результатов учащихся особенно важно, т.к. именно они активизируют процесс индивидуально-личностного становления учащихся.

Новизна программы

Задача построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического развития Российской Федерации до 2020 года и долгосрочным прогнозом научно-технологического развития Российской Федерации до 2025 года, не может быть решена без существенных изменений системы дополнительного образования детей, создания новых общеразвивающих программ технической направленности.

Изменение взглядов на программирование как науку, его место в системе научного знания требует существенных изменений в содержании образовательного процесса. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей образовательной среды.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения в отношении к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции – одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных учащихся. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей.

Содержание обучения, представленное в программе «Программирование на языке «Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов – блок-схем, алгоритмов, программ – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний.

Отличительные особенности

Особенность данной программы заключается в том, что основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

Исключительно велика роль программирования для формирования мышления школьников, приемов умственных действий, умения строить модели, самостоятельного нахождения и составления алгоритмов решения задач, умения четко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования общеинтеллектуальных и общеучебных умений школьников активизирует процесс индивидуально-личностного становления учащихся. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции - одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования - необходимое условие подготовки современных школьников.

Адресат программы

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке «Python» ориентирована на детей в возрасте 15-18 лет, и составлена с учётом современных потребностей рынка в специалистах в области информационных технологий. К зачисленным учащимся не предъявляются требования относительно наличия базовых знаний, специальных способностей. Принцип набора в группы – свободный.

Наполняемость групп– 12 -15 человек.

Расширение связей с окружающим миром, широкое всепоглощающее общение со сверстниками, личные интересы и увлечения часто снижают непосредственный интерес к учению. Сознательно – положительное отношение ребят к учению возникает тогда, когда учение удовлетворяет их познавательные потребности, благодаря чему знания приобретают для них определенный смысл как необходимое и важное условие подготовки к будущей самостоятельной жизни.

В учебной деятельности у школьника формируется интерес к самому процессу учебной деятельности без осознания её значения. Только после возникновения интереса к результатам своего учебного труда формируется интерес к содержанию учебной деятельности, к приобретению знаний. Вот эта основа и является благоприятной почвой для формирования устойчивых мотивов учения, связанных соответственным отношением к учебным занятиям. Указанные особенности учитываются при организации обучения. Набор в объединение построен на свободной основе (по желанию ребёнка и с согласия родителей).

Программа составлена с учётом индивидуальных и возрастных особенностей учащихся. Психолого-педагогические особенности учащихся определяют и методы индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута.

Объём и сроки реализации программы.

Объём программы – 136 часов

Программа рассчитана на 2 год обучения, т.е. 68 недель

Уровень обучения:

Уровень обучения - базовый: направлен на освоение определенного вида деятельности, углубление и развитие их интересов и навыков, расширение спектра специализированных занятий по дисциплине; формирование устойчивой мотивации к выбранному виду деятельности; формирование специальных знаний и практических навыков, развитие творческих способностей ребенка. В процессе обучения накапливаются базовые знания, умения и навыки, что способствует не только успешности обучения, но и создает возможности освоения творческо-продуктивной, проектной и учебно-исследовательской деятельности.

Форма занятия – групповая (12-15 человек).

Режим занятий – занятия проводятся 2 раза в неделю, всего 68 часов за весь период обучения. Занятия объединения проводятся согласно расписанию. Занятия по данной программе будут проходить с использованием оборудования и кабинета Центра образования естественно-научной и технологической направленностей «Точка роста».

Форма обучения: очная.

Формы организации занятий следующие:

1. фронтальная работа с демонстрационным материалом;
2. практическая, творческая работы;
3. самостоятельная работа детей с раздаточным материалом;
4. совместная деятельность детей;
5. совместная деятельность взрослого и детей;
6. самостоятельная деятельность.

Цель и задачи дополнительной общеобразовательной программы

Цель программы:

Способствовать развитию инженерного мышления, воспитание конкурентно способной личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи:

Образовательные (предметные)

- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- обучить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.
- обучить языку программирования Python и созданию программ на его основе;
- научить создавать прикладное программное обеспечение;
- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

Личностные:

- критическое отношение к информации и избирательность её восприятия;
- осмысление мотивов своих действий при выполнении заданий;
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремленности, умения преодолевать трудности – качеств весьма важных в практической деятельности любого человека;
- развитие самостоятельности суждений, независимости и нестандартности мышления;
- воспитание чувства справедливости, ответственности;
- начало профессионального самоопределения, ознакомление с миром профессий, связанных с программированием.

Метапредметные:

Регулятивные универсальные учебные действия:

- принимать и сохранять учебную задачу;
- планировать последовательность шагов алгоритма для достижения цели;
- формировать умения ставить цель – создание творческой работы, планировать достижение этой цели;

- осуществлять итоговый и пошаговый контроль по результату;
- адекватно воспринимать оценку учителя;
- различать способ и результат действия;
- вносить коррективы в действия в случае расхождения результата решения задачи на основе ее оценки и учета характера сделанных ошибок;
- в сотрудничестве с учителем ставить новые учебные задачи;
- проявлять познавательную инициативу в учебном сотрудничестве;
- осваивать способы решения проблем творческого характера в жизненных ситуациях;
- оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

Познавательные универсальные учебные действия:

- осуществлять поиск информации в индивидуальных информационных архивах учащегося, информационной среде образовательного учреждения, в федеральных хранилищах информационных образовательных ресурсов;
- использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач;
- ориентироваться на разнообразие способов решения задач;
- осуществлять анализ объектов с выделением существенных и несущественных признаков;
- проводить сравнение, классификацию по заданным критериям;
- строить логические рассуждения в форме связи простых суждений об объекте;
- устанавливать аналогии, причинно-следственные связи;
- моделировать, преобразовывать объект из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая);
- синтезировать, составлять целое из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
- выбирать основания и критерии для сравнения и классификации объектов;

Коммуникативные универсальные учебные действия:

- аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
- выслушивать собеседника и вести диалог;
- признавать возможность существования различных точек зрения и права каждого иметь свою;
- планировать учебное сотрудничество с учителем и сверстниками — определять цели, функций участников, способов взаимодействия;
- осуществлять постановку вопросов — инициативное сотрудничество в поиске и сборе информации;
- разрешать конфликты – выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация;
- управлять поведением партнера — контроль, коррекция, оценка его действий;
- уметь с достаточной полнотой и точностью выразить свои мысли в соответствии с задачами и условиями коммуникации;
- владеть монологической и диалогической формами речи.

**Содержание программы
Перечень разделов**

№п/п	Наименование разделов / тем	Количество часов на изучение
------	-----------------------------	------------------------------

		Всего	Теория	Практика
1.	Раздел 1. Знакомство с языком Python	6	3	3
2.	Раздел 2. Переменные и выражения	10	4	6
3	Раздел 3. Условные предложения	18	5	13
4	Раздел 4. Циклы	34	8	26
5	Раздел 5. Функции	21	6	16
6	Раздел 6. Строки - последовательности символов	14	3	11
7	Раздел 7. Сложные типы данных	21	7	14
8	Раздел 8. Стиль программирования и отладка программ	12	7	5
	ВСЕГО	136	43	93

СОДЕРЖАНИЕ КУРСА

Вводное занятие (1 час).

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. Практика: Опрос. Тренинг на командообразование.

Раздел 1. Знакомство с языком Python (5 часов)

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа а. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python Практическая работа 1.2. Режимы работы с Python Тест №1. Знакомство с языком Python

Учащиеся должны знать/понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Раздел 2. Переменные и выражения (10 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами

Тест №2. Выражения и операции.

Учащиеся должны знать/понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;

–оператор присваивания;

Раздел 3. Условные предложения (18 часов)

Теория: 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 3.1. Логические выражения

Практическая работа 3.2. Условный оператор

Практическая работа 3.3. Множественное ветвление

Практическая работа 3.4. "Условные операторы"

Самостоятельная работа №1. Решение задач по теме "Условные операторы".

Зачетная работа №1. "Составление программ с ветвлением".

Тест №3. "Условные операторы".

Учащиеся должны знать/понимать:

–назначение условного оператора;

–способ записи условного оператора;

–логический тип данных;

–логические операторы or, and, not;

Учащиеся должны уметь:

–использовать условный оператор;

–создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы (34 часов)

Теория: 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 4.1. Числа Фибоначчи

Практическая работа 4.2. Решение задачи с циклом for.

Практическая работ 4.3. Реализация циклических алгоритмов

Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом.

Самостоятельная работа №2 "Составление программ с циклом"

Тест №4. Циклы

Творческая работа № 1. "Циклы"

Учащиеся должны знать/понимать:

–циклы с условием и их виды;

–правила записи циклов условием;

–назначение и особенности использования цикла с параметром;

–формат записи цикла с параметром;

–примеры использования циклов различных типов.

Учащиеся должны уметь:

–определять вид цикла, наиболее удобный для решения поставленной задачи;

–использовать цикл с условием;

–определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

Раздел 5. Функции (21 часов)

Теория: 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. Числа Фибоначчи.

Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций

Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа №3 по теме "Функции"

Тест №5. Функции

Учащиеся должны знать/понимать:

–понятие функции;

–способы описания функции;

–принципы структурного программирования;

–понятие локальных переменных подпрограмм;

–понятие формальных и фактических параметров подпрограмм;

–способ передачи параметров.

Учащиеся должны уметь:

–создавать и использовать функции;

–использовать механизм параметров для передачи значений.

Раздел 6. Строки - последовательности символов (14 часов)

Теория: 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 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Учащиеся должны знать/понимать:

–назначение строкового типа данных;

–операторы для работы со строками;

–процедуры и функции для работы со строками;

–операции со строками.

Учащиеся должны уметь:

–описывать строки;

–соединять строки;

–находить длину строки;

–вырезать часть строки;

–находить подстроку в строке;

–находить количество слов в строке.

Раздел 7. Сложные типы данных (21 часов)

Теория: 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.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи.

Присваивание кортежей. Кортеж и как возвращаемые значения

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

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками.

Тест №7. Списки

Учащиеся должны знать/ понимать:

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

Учащиеся должны уметь:

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

Раздел 8. Стиль программирования и отладка программ (12 часов)

Стиль программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

Учащиеся должны знать/ понимать:

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

Учебно-тематическое планирование

Перечень разделов

№п/п	Наименование разделов / тем	Количество часов на изучение			Формы контроля/ аттестации
		Всего	Теория	Практика	
	Первый год обучения				
1.	Раздел 1. Знакомство с языком Python	6	3	3	
	Инструктаж по технике безопасности в компьютерном классе	1	1		Тестирование, решение практических задач
1.1	Общие сведения о языке Практическая работа 1.1: «Установка программы Python»	2	1	1	Тестирование, решение практических задач

1.2.	Режимы работы Практическая работа 1.2: «Режимы работы с Python»	2	1	1	Тестирование, решение практических задач
1.3	Тест №1. Знакомство с языком Python	1		1	Тестирование, решение практических задач
2.	Раздел 2. Переменные и выражения	10	4	6	
2.1	Переменные. Практическая работа 2.1: «Работа со справочной системой»	2	1	1	Тестирование, решение практических задач
2.2	Практическая работа 2.2: «Переменные»	2		2	Тестирование, решение практических задач
2.3	Выражения Практическая работа 2.3: «Выражения»	2	1	1	Тестирование, решение практических задач
2.4	Ввод и вывод	1	1		Тестирование, решение практических задач
2.5	Задачи на элементарные действия с числами Практическая работа 2.5: «Задачи на элементарные действия с числами»	2	1	1	Тестирование, решение практических задач
2.6	Тест №2. Выражения и операции.	1		1	Тестирование, решение практических задач
3	Раздел 3. Условные предложения	18	5	13	
3.1	Логические выражения и операторы. Практическая работа 3.1: «Логические выражения»	2	1	1	Тестирование, решение практических задач
3.2	Условный оператор Практическая работа 3.2: «Условный оператор»	2	1	1	Тестирование, решение практических задач
3.3	Множественное ветвление Практическая работа 3.3: «Множественное ветвление»	2	1	1	Тестирование, решение практических задач
3.4	Реализация ветвления в языке Python. Практическая работа 3.4: «Условные операторы»	2	1	1	Тестирование, решение практических задач
3.5	Самостоятельная работа №1 по теме «Условные операторы».	4	1	3	Тестирование, решение практических задач
3.6	Зачетная работа №1. "Составление программ с ветвлением".	4		4	Тестирование, решение практических задач
3.7	Тест №3. "Условные операторы".	2		2	Тестирование, решение практических задач
4	Раздел 4. Циклы	34	8	26	
4.1.	Оператор цикла с условием Практическая работа 4.1: "Числа Фибоначчи"	2	1	1	Тестирование, решение практических задач

4.2.	Оператор цикла for Практическая работа 4.2: «Решение задачи с циклом for»	2	1	1	Тестирование, решение практических задач
4.3.	Вложенные циклы Практическая работа 4.3: Реализация циклических алгоритмов	4	1	3	Тестирование, решение практических задач
4.4.	Случайные числа Практическая работа 4.4: «Случайные числа»	4	1	3	Тестирование, решение практических задач
4.5.	Примеры решения задач с циклом Практическая работа 4.5: «Решение задач с циклом»	6	1	5	Тестирование, решение практических задач
4.6	Самостоятельная работа №2 "Составление программ с циклом"	6	2	4	Тестирование, решение практических задач
4.7	Тест №4. Циклы	2		2	Тестирование, решение практических задач
4.8	Творческая работа №1. "Циклы"	8	1	7	Тестирование, решение практических задач
Второй год обучения					
5	Раздел 5. Функции	21	6	16	
5.1.	Создание функций Практическая работа 5.1: «Создание функций»	2	1	1	Тестирование, решение практических задач
5.2.	Локальные переменные Практическая работа 5.2: «Локальные переменные»	2	1	1	Тестирование, решение практических задач
5.3.	Примеры решения задач с использованием функций Практическая работа 5.3: «Решение задач с использованием функций»	6	1	5	Тестирование, решение практических задач
5.4	Самостоятельная работа №3 по теме "Функции"	4	2	3	Тестирование, решение практических задач
5.5	Рекурсивные функции Практическая работа 5.4. «Рекурсивные функции»	5	1	4	Тестирование, решение практических задач
5.6	Тест № 5. Функции	2		2	Тестирование, решение практических задач
6	Раздел 6. Строки - последовательности символов	14	3	11	
6.1.	Строки. Практическая работа: 6.1 «Строки»	4	1	3	Тестирование, решение практических задач
6.2.	Срезы строк	2	1	1	Тестирование, решение практических задач
6.3	Примеры решения задач со строками Практическая работа: 6.2 «Решение задач со строками»	8	1	7	Тестирование, решение практических задач
7	Раздел 7.Сложные типы данных	21	7	14	

7.1.	Списки. Практическая работа: Списки	2	1	1	Тестирование, решение практических задач
7.2	Срезы списков	1	1		Тестирование, решение практических задач
7.3	Списки: примеры решения задач Практическая работа 7.2: «Решение задач со списками»	8	1	7	Тестирование, решение практических задач
7.4	Матрицы	2	1	1	Тестирование, решение практических задач
7.5	Списки	4		4	Тестирование, решение практических задач
7.6	Кортежи	1	1		Тестирование, решение практических задач
7.7	Введение в словари	1	1		Тестирование, решение практических задач
7.8	Множества в языке Python	2	1	1	Тестирование, решение практических задач
8	Раздел 8. Стиль программирования и отладка программ	12	7	5	Тестирование, решение практических задач
8.1	Стиль программирования	2	1	1	Тестирование, решение практических задач
8.2	Отладка программ	4	2	2	Тестирование, решение практических задач
8.3	Зачет по курсу «Программирование на языке Python»	4	2	2	Тестирование, решение практических задач
8.3	Итоговое занятие	2	2		Тестирование, решение практических задач
	ВСЕГО	136	43	93	

Формы аттестации / контроля.

Дополнительная общеобразовательная общеразвивающая программа «Робототехника» не предполагает каких-либо специальных зачётных или экзаменационных часов. Текущий контроль осуществляется в течение всего курса обучения в различных формах. Основные формы подведения итогов и оценка результатов обучения: метод проектов; семинары; экспериментальная и практическая работа; участие в олимпиадах и интеллектуальных марафонах; смотр знаний и т.д.

Промежуточная аттестация проводится как оценка результатов обучения за год и включает в себя проверку теоретических знаний, практических умений и навыков. Итоговая аттестация воспитанников проводится по окончании обучения по дополнительной общеразвивающей программе.

Результаты итоговой аттестации учащихся должны оцениваться таким образом, чтобы можно было определить:

- насколько достигнуты прогнозируемые результаты дополнительной общеразвивающей программой каждым учащимся;
- полноту выполнения дополнительной общеразвивающей программы каждым учащимся;
- результативность самостоятельной деятельности каждым учащимся в течение всего периода обучения.

Содержание программы

Данная рабочая программа преследует цель формирования начальных знаний и понятий в области конструирования роботов, необходимых для дальнейшего освоения базового и углублённого модулей. Для наиболее эффективного освоения обучающимися изучаемого материала основные лекции курса сопровождаются практиками, в том числе с использованием технологического оборудования.

Виды контроля:

- входной – проводится перед началом работы и предназначен для определения стартового уровня возможностей учащихся;
- текущий, проводимый в течение учебного года в процессе освоения учащимися программы;
- промежуточный – предназначен для оценки уровня и качества освоения учащимися программы, либо по итогам изучения раздела/темы, либо в конце определённого периода обучения – полугодия;
- итоговый – осуществляется по завершению всего периода обучения по программе.

Формы проверки промежуточных результатов: тестирование, создание программы.

Планируемые результаты

В рамках курса Программирования на языке Python» обучающиеся

На предметном уровне

должны знать/ понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.
- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;

- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

Личностные результаты

- критическое отношение к информации и избирательность её восприятия;
- осмысление мотивов своих действий при выполнении заданий;

- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремленности, умения преодолевать трудности – качеств весьма важных в практической деятельности любого человека;
- развитие самостоятельности суждений, независимости и нестандартности мышления;
- воспитание чувства справедливости, ответственности;
- начало профессионального самоопределения, ознакомление с миром профессий, связанных с робототехникой.

Метапредметные результаты

Регулятивные универсальные учебные действия:

- принимать и сохранять учебную задачу;
- планировать последовательность шагов алгоритма для достижения цели;
- формировать умения ставить цель – создание творческой работы, планировать достижение этой цели;
- осуществлять итоговый и пошаговый контроль по результату;
- адекватно воспринимать оценку учителя;
- различать способ и результат действия;
- вносить коррективы в действия в случае расхождения результата решения задачи на основе ее оценки и учета характера сделанных ошибок;
- в сотрудничестве с учителем ставить новые учебные задачи;
- проявлять познавательную инициативу в учебном сотрудничестве;
- осваивать способы решения проблем творческого характера в жизненных ситуациях;
- оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

Познавательные универсальные учебные действия:

- осуществлять поиск информации в индивидуальных информационных архивах учащегося, информационной среде образовательного учреждения, в федеральных хранилищах информационных образовательных ресурсов;
- использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач;
- ориентироваться на разнообразие способов решения задач;
- осуществлять анализ объектов с выделением существенных и несущественных признаков;
- проводить сравнение, классификацию по заданным критериям;
- строить логические рассуждения в форме связи простых суждений об объекте;
- устанавливать аналогии, причинно-следственные связи;
- моделировать, преобразовывать объект из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая);
- синтезировать, составлять целое из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
- выбирать основания и критерии для сравнения и классификации объектов;

Коммуникативные универсальные учебные действия:

- аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
- выслушивать собеседника и вести диалог;
- признавать возможность существования различных точек зрения и права каждого иметь свою;
- планировать учебное сотрудничество с учителем и сверстниками — определять цели, функций участников, способов взаимодействия;
- осуществлять постановку вопросов — инициативное сотрудничество в поиске и сборе информации;
- разрешать конфликты – выявление, идентификация проблемы, поиск и оценка

- альтернативных способов разрешения конфликта, принятие решения и его реализация;
- управлять поведением партнера — контроль, коррекция, оценка его действий;
- уметь с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации;
- владеть монологической и диалогической формами речи.

Комплекс организационно-педагогических условий Условия реализации программы.

Материально-техническое обеспечение

В перечень оборудования ЦО «Точка роста» МОУ-СОШ №3 г. Аткарска Саратовской области, в котором будет реализована данная программа, входят:

- столы для практических работ с конструктором, полки для хранения собранных моделей, компьютеры с установленным необходимым программным обеспечением;
- робототехнические образовательные конструкторы КЛИК
- источники питания,
- Ноутбуки;
- Телевизор;
- МФУ (принтер, сканер, копир);

Программное обеспечение:

Операционная система Linux

Интерпретатор Python версии 3.7 и выше;

IDE Jupyter PyCharm;

Foxit Reader или другой просмотрщик PDF файлов; WinRAR;

Пакет офисных программ:

Текстовый редактор (Блокнот)

Текстовый процессор (OpenOfficeWriter);

Табличный процессор (OpenOfficeCalc);

Adobe Photoshop или другой растровый графический редактор;

Любой браузер для интернет серфинга.

Информационное обеспечение программы

Архив (набор) презентаций по темам, видеоуроки, методические и дидактические пособия для проведения занятий, проверки и закрепления знаний по программе.

Формы аттестации и их периодичность

В объединении «Основы алгоритмики и логики» педагогом осуществляется мониторинг эффективности образовательного процесса:

1. входной контроль (форма: анкетирование, диагностика, наблюдение, опрос);
2. текущий контроль (форма: наблюдение, опрос, ведение таблицы результатов, тестирование);
3. итоговый контроль (форма: тестирование, опрос, создание и защита проектов, соревнования).

Целью мониторинга является диагностика предметных, метапредметных, личностных результатов учащихся. Основная задача мониторинга – непрерывное отслеживание состояния образовательного процесса. Выясняются следующие вопросы: достигается ли цель образовательного процесса, существует ли положительная динамика в развитии учащегося по сравнению с результатами предыдущих диагностических исследований, существуют ли предпосылки для совершенствования работы преподавателя и коррекции программы.

По окончании изучения каждого раздела проводится промежуточный контроль, позволяющий определить качество усвоенного материала раздела и изучать учебный материал дальше на том же уровне, а также позволяет перейти (при выполнении тестовых заданий повышенной сложности) на следующий уровень. Также проводится итоговый контроль (формы: тест, опрос).

Эффективность реализации программы определяется согласно разработанным критериям количества и качества.

Формы аттестации

- Защита исследовательской работы, проекта, сопровождающееся показом презентации, проведением опытов и экспериментов (в зависимости отвыбранной темы).
- Выступление на конференции муниципального уровня.
- Итоговая аттестация (государственная итоговая аттестация) не предусмотрена.

Оценочные материалы

- Критерии оценивания итогового проекта, исследовательской работы
 - самостоятельность планирования и выполнения;
 - качество проведения исследования, эксперимента;
 - качество оформления результатов;
 - оригинальность и качество решения;
 - связь с жизнью;
 - наличие и качество презентации;
 - умение свободно представить результат

Методическое обеспечение

В зависимости от поставленных задач на занятиях используются различные формы и методы обучения.

Педагогические технологии - ИКТ, разноуровневое обучение, проблемное и поисковое обучение, технология личностно ориентированного обучения И.С. Якиманской (ситуация успеха, возможность выбора, атмосфера сотрудничества, рефлексия) и межпредметных связей. Занятия предполагают не только приобретение дополнительных знаний по информатике, но и развитие способности у них самостоятельно приобретать знания, умений проводить опыты, вести наблюдения. На занятиях используются интересные факты, привлекающие внимание связью с жизнью, объясняющие загадки привычных с детства явлений.

Формы организации деятельности детей на занятии: индивидуальная и групповая.

Формы проведения занятий:

- Беседа
- Практикум
- Практическая работа
- Исследовательская работа
- Круглый стол
- Проектная работа
- Защита проекта

Список литературы для педагога, учащихся и их родителей

Список литературы для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
5. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>.
6. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>.
7. Python. Подробный справочник Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.

Список литературы для родителей и учащихся

1. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
3. Сайт / Адаптивный тренажер Python[Электронный ресурс] – Режим доступа: <https://stepik.org/course/431> , свободный.
4. Сайт / среда разработки для языка Python[Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu> , свободный.
5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net , свободный.