
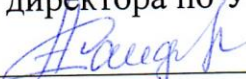



МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЛМЫКИЯ
ЧАСТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«СОВРЕМЕННЫЙ ГУМАНИТАРНЫЙ ЛИЦЕЙ»

РАССМОТРЕНО На заседании МО  Адучиев Е.Б. Приказ № <u>1</u> от « <u>29</u> » <u>08</u> 2024 г.	СОГЛАСОВАНО Зам. директора по УВР  Санджиева С.Н. Приказ № <u>1</u> от « <u>30</u> » <u>08</u> 2024 г.	УТВЕРЖДЕНО Директор  Бухаев Д.Э. Приказ № <u>122</u> от « <u>30</u> » <u>08</u> 2024 г.
--	--	---

РАБОЧАЯ ПРОГРАММА
СТУДИИ «ПРОГРАММИРОВАНИЕ»
ТВОРЧЕСКОЙ ЛАБОРАТОРИИ «ИМПУЛЬС»
для обучающихся 6 – 11 классов

Пояснительная записка

Дополнительная общеобразовательная программа «Основы программирования на языке «Python» разработана согласно требованиям следующих нормативных документов:

- Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ;
- Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 № 1726-р).
- СанПиН к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (утверждено постановлением Главного государственного санитарного врача РФ от 04.07.2014 № 41);
- Условия реализации дополнительных общеобразовательных программ (прописано в Федеральном законе № 273-ФЗ, п.1,2,3,9 ст.13; п. 1,5,6; ст. 14; ст. 15; ст. 16; ст. 33; ст. 34; ст.75);
- Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден приказом Министерства образования и науки РФ от 29.08.2013 № 1008);
- Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена Распоряжением Правительства Российской Федерации от 29 мая 2015 г. N 996-р).
- Концепцией развития дополнительного образования детей (утв. Распоряжением Правительства РФ от 04.09.2014 No1726);
- Концепцией духовно-нравственного развития и воспитания личности гражданина России;
- Государственной программы Российской Федерации «Развитие образования на 2018-2025г.г.» (пост. Правительства РФ от 26.12.2017г. № 1642);

Направленность программы

Программа технической направленности., которая направлена на

- формирование и развитие творческих способностей обучающихся
- развитие и поддержку детей, проявивших интерес и определенные способности к техническому творчеству, техническому моделированию.

Новизна программы

Задача построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического развития Российской Федерации и долгосрочным прогнозом научно-технологического развития Российской Федерации, 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.

Актуальность программы

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.

Адресат программы

Дополнительная общеразвивающая программа «Основы программирования на языке Python» рассчитана на обучающихся 6-11 классов, проявляющих интерес к информационно-коммуникационным технологиям.

Подростковый возраст от 10 до 18 лет. Переход от детства к взрослости составляет главный смысл и специфическое различие этого этапа. Подростковый период считается «кризисным», такая оценка обусловлена многими качественными сдвигами в развитии подростка. Именно в этом возрасте происходят интенсивные и кардинальные изменения в организации ребенка на пути к биологической зрелости и **полового созревания**. Анатомо-физиологические сдвиги в развитии подростка порождают психологические новообразования: чувство взрослости, развитие интереса к противоположному полу, пробуждение определенных романтических чувств. Характерными новообразованиями подросткового возраста есть стремление к самообразованию и самовоспитанию, полная определенность склонностей и профессиональных интересов.

Юношеский возраст от 12 до 18 лет. В юношеском возрасте происходит интенсивное физиологическое и психическое развития. Особое значение в юношеском возрасте приобретает моральное воспитание, основные виды деятельности — учение и посильный труд, увеличивается диапазон социальных ролей и обязательств.

Психическое развитие личности в юношеском возрасте тесно связано с обучением, трудовой деятельностью и усложнением **общения со взрослыми**. В связи с началом трудовой деятельности **отношения между личностью** и обществом значительно углубляются, что приводит к наиболее четкому пониманию своего места в жизни.

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.

Объем и срок освоения программы

Общее количество учебных часов, запланированных на весь период обучения и необходимых для освоения программы 306 часов. Продолжительность реализации программы: 1 учебный год.

Формы обучения - очное, с применением электронного обучения, дистанционных образовательных технологий. Занятия ведутся на русском языке

Особенности организации образовательного процесса

Учащиеся объединения являются разными по возрасту и социальному статусу. Группы являются основным составом объединения, состав группы постоянный.

Возраст детей, участвующих в освоении программы – 10 - 18 лет

Условия набора детей в коллектив: принимаются все желающие, не имеющие медицинских противопоказаний. Если группе первого года обучения появились вакантные места, то допускается дополнительный набор обучающихся на основании результатов собеседования.

Режим занятий, периодичность и продолжительность занятий

Режим занятий в день 2 часа. Периодичность занятий 5 раз в неделю. Продолжительность занятий - 80 минут. Между занятиями предусматривается 10 минутный перерыв на отдых. Учитываются индивидуальные особенности каждого обучающегося, и при необходимости возможна корректировка времени занятия согласно особенностям обучающегося.

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 практикоориентированных 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.

Отличительные особенности программы

Программа «Основы программирования на языке «Python» является модифицированной, составлена на основе собственного опыта и использования элементов авторских программ и курсов: *А.А. Чернов «Программирование на языках высокого уровня», программы курса К.Ю. Полякова; *«Информатика», углубленный уровень, курса «Основы программирования», автор СМ. Окулов, курса М. Лутца «Изучаем Питон», А.А. Чернов «Программирование на языках высокого уровня». Волгоград: «Учитель», 2012;

* «Основы программирования». С.М. Окулов и др., М.: Бином. Лаборатория Знаний, 2012, М. Лутц «Изучаем Питон», Санкт-Петербург: Символ, 2013г.;

* «Программирование на языке Python» программа дополнительного образования, автор: Ефимов Кирилл Николаевич, педагог дополнительного образования, г. Калуга;

* «Программирование аркадных игр на языке Python» в дополнительном образовании, автор- составитель Прядкина Елена Владимировна, учитель информатики, г. Москва.

По уровню освоения программа относится к общекультурной. По уровню сложности – к стартовой.

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

Можно без преувеличения сказать, что программа имеют потенциальные предпосылки для развития коммуникативных и речевых действий в силу их действительно универсального, т. е. максимально обобщенного, характера.

Заложенные в Федеральном государственном образовательном стандарте второго поколения основы формирования универсальных учебных действий подчёркивают ценность современного образования – школа должна побуждать молодёжь принимать активную гражданскую позицию, усиливать личностное развитие и безопасную социальную включённость в жизнь общества.

Цель программы

Цель: 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:

- 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 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 коррективы 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, функции 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.

Учебный план

№ п/п	Наименование разделов и тем	Кол-во часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Инструктаж по технике безопасности в компьютерном классе		2	2		Опрос
1.	Раздел 1. Знакомство с языком Python	26	12	14	Тестирование, решение практических задач
1.1	Занятие1. Общие сведения о языке Python Практическая работа: Установка программы Python	12	6	6	
1.2.	Занятие2. Режимы работы Практическая работа: Режимы работы сPython	12	6	6	
1.3	Тест №1. Знакомство сязыком Python	2		2	
2.	Раздел 2. Переменные и выражения	31	14	17	Тестирование, решение практических задач
2.1	Занятие3. Переменные Практическая работа: Работа со справочной системой	5	2	3	
2.2	Практическая работа: Переменные	4		4	
2.3	Занятие4. Выражения Практическая работа: Выражения	8	4	4	
2.4	Занятие5. Ввод и вывод	4	4		
2.5	Занятие6. Задачи на элементарные действия с числами Практическая работа2.5. Задачи на элементарные действия с числами	8	4	4	
2.6	Тест №2.Выражения и операции.	2		2	
3	Раздел 3. Условные предложения	40	16	24	Тестирование, Решение практических задач
3.1	Занятие7. Логические выражения и операторы. Практическая работа: Логические выражения	5	2	3	
3.2	Занятие8. Условный оператор Практическая работа: «Условный оператор»	7	4	3	
3.3	Занятие9. Множественное ветвление Практическая работа: Множественное ветвление	10	4	6	
3.4	Занятие10. Реализация ветвления в языке Python. Практическая работа: «Условные операторы»	8	4	4	
3.5	Самостоятельная работа №1 по теме «Условныеоператоры».	4	2	2	
3.6	Занятие11. Зачетная работа №1. "Составление программ с ветвлением".	4		4	
3.7	Тест №3."Условныеоператоры".	2		2	
4	Раздел 4. Циклы	76	19	57	Тестирование, решение практических задач, творческая работа
4.1.	Занятие12. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	8	3	5	
4.2.	Занятие13. Оператор цикла for Практическая работа Решение задачи с циклом for.	8	3	5	
4.3.	Занятие14. Вложенные циклы Практическая работа: Реализация циклических алгоритмов	8	3	5	
4.4.	Занятие15. Случайные числа Практическая работа: Случайные числа	8	3	5	
4.5.	Занятие16. Примеры решения задач с циклом Практическая работа: Решение задач сциклом.	13	3	10	
4.6	Самостоятельная работа №2 "Составление программ с циклом"	13	2	11	

4.7	Тест №4. Циклы	2		2	
4.8	Занятие17. Творческая работа№1. "Циклы"	16	2	14	
5	Раздел 5. Функции	34	16	18	
5.1.	Занятие 18. Создание функций Практическая работа Создание функций	6	3	3	Тестирование, решение практических задач
5.2.	Занятие 19. Локальные переменные Практическая работа Локальные переменные	6	3	3	
5.3.	Занятие 20. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	6	3	3	
5.4	Самостоятельная работа №3 по теме "Функции"	8	4	4	
5.5	Занятие 21. Рекурсивные функции Практическая работа5.4. Рекурсивные функции	6	3	3	
5.6	Тест № 5. Функции	2		2	
6	Раздел 6. Строки - последовательности символов	30	13	17	
6.1.	Занятие 22. Строки Практическая работа: Строки	8	4	4	Решение практических задач
6.2.	Занятие23. Срезы строк	8	4	4	
6.3	Занятие24. Примеры решения задач со строками Практическая работа: Решение задач со строками.	14	5	9	
7	Раздел 7.Сложныетипыданных	37	17	20	
7.1.	Занятие25. Списки Практическая работа:Списки	5	2	3	Тестирование, Решение практических задач
7.2	Занятие26.Срезысписков	2	2		
7.3	Занятие27. Списки: примерырешениязадач Практическая работа7.2. Решение задач сосписками	9	2	7	
7.4	Занятие28. Матрицы	5	2	3	
7.5	Тест №7. Списки	4		4	
7.6	Занятие29. Кортежи	3	3		
7.7	Занятие30. Введение в словари	3	3		
7.8	Занятие31. Множества в языкеPython	6	3	3	
8	Раздел 8.Стиль программирования и отладкапрограмм	30	16	14	
8.1	Занятие32. Стиль программирования	12	6	6	Решение практических задач, презентация проекта, рефлексия.
8.2	Занятие33. Отладка программ	12	6	6	
8.3	Занятие34. Зачет по курсу «Программирование на языке Python»	4	2	2	
8.3	Занятие35. Что дальше?	2	2		
	ВСЕГО	306	125	181	

Календарно-тематическое планирование студии «Программирование»

№	Содержание занятия	Раздел, тема	Время на изучение	
			теоретических	практических
1.	Теоретические занятия. Знакомство с планом работы объединения, инструктаж по ТБ. Практические занятия. Общие сведения о языке Python. Установка Python на компьютер.	Общие сведения о языке Установка программы Python	1	2
2.	Теоретические занятия. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Практические занятия программы Python. Режимы работы с Python	Режимы работы Режимы работы с Python	1	2
3.	Практические занятия. Знакомство с языком Python	Знакомство с языком Python	1	2
4.	Теоретические занятия Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Практические занятия Выражения. Операции. Порядок выполнения операций. Композиция.	Переменные Работа со справочной системой.	1	2
5.	Практические занятия Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.	Практическая работа: Переменные	1	2
6.	Теоретические занятия Выражения. Операции. Практические занятия Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.	Выражения Практическая работа: Выражения	1	2
7.	Теоретические занятия Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных	Ввод и вывод	1	2
8.	Теоретические занятия. Решение задач на элементарные действия с числами. Практические занятия Выполнение практической работы	Задачи на элементарные действия с числами	1	2
9.	Теоретические занятия Работа со справочной системой	Выражения и операции.	1	2
10.	Теоретические занятия Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Практические занятия Примеры решения задач с условным оператором. Множественное ветвление.	Логические выражения и операторы. Практическая работа: Логические выражения	1	2
11.	Теоретические занятия Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Практические занятия Работа с программными блоками.	Условный оператор	1	2
12.	Теоретические занятия Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным	Множественное ветвление	2	2

	оператором. Множественное ветвление. Реализация ветвления в языке Python. Практические занятия Примеры решения задач с условным оператором. Множественное ветвление..			
13.	Теоретические занятия Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python	Множественное ветвление	2	2
14.	Практические занятия Работа с программными блоками.	Реализация ветвления	2	2
15.	Практические занятия Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа	Самостоятельная работа № 1 по теме «Условные операторы».	2	2
16.	Практические занятия 3.4. "Условные операторы"		2	2
17.	Практические занятия Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа 3.4. "Условные операторы"	Зачетная работа № 1. "Составление программ с ветвлением".	2	2
18.	Практические занятия Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа 3.4. "Условные операторы"	Зачетная работа № 1. "Составление программ с ветвлением".	2	2
19.	Практические занятия Решение задач по теме "Условные операторы".	Тест № 3. "Условные операторы".	2	2
20.	Теоретические занятия 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 условием	2	2
21.	Практические занятия Числа Фибоначчи" Практическая работа	Практическая работа "Числа Фибоначчи"	2	2
22.	Практические занятия Программирование робота на движение змейкой или по спирали.	Оператор цикла for	2	2
23.	Теоретические занятия Поворот на месте и знакомство с блоком EV3, генерирующим случайные числа.	Практическая работа Решение задачи с циклом for.	2	2
24.	Теоретические занятия Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Практические занятия Реализация циклических алгоритмов Практическая работа	Вложенные циклы	2	3
25.	Практические занятия Реализация циклических алгоритмов Практическая работа		Реализация циклических алгоритмов	2
26.	Теоретические занятия Случайные числа. Функция randrange. Функция random.	Случайные числа	2	3

27.	Практические занятия Решения задач с случайными числами	Случайные числа	2	3
28.	Практические занятия Примеры решения задач с циклом.	Примеры решения задач с циклом	2	3
29.	Практические занятия Решение задач с циклом	Решение задач с циклом.	2	3
30.	Практические занятия Решение задач с циклом	Решение задач с циклом.	2	3
31.	Практические занятия Решение задач с циклом	Решение задач с циклом.	2	3
32.	Практические занятия Выполнение самостоятельной работы	Самостоятельная работа № 2	2	3
33.	Практические занятия Выполнение самостоятельной работы	Самостоятельная работа № 2	2	3
34.	Практические занятия Выполнение самостоятельной работы.	Самостоятельная работа № 2	2	3
35.	Практические занятия Выполнение самостоятельной работы.	Самостоятельная работа № 2	2	3
36.	Практические занятия Выполнение самостоятельной работы.	Самостоятельная работа № 2	2	3
37.	Практические занятия Решение теста	Тест № 4. Циклы	2	3
38.	Практические занятия Выполнение творческой работы.	Творческая работа № 1. "Циклы"	2	3
39.	Практические занятия Выполнение творческой работы.	Творческая работа № 1. "Циклы"	2	3
40.	Практические занятия Выполнение творческой работы..	Творческая работа № 1. "Циклы"	2	3
41.	Практические занятия Выполнение творческой работы..	Творческая работа № 1. "Циклы"	2	3
42.	Практические занятия Выполнение творческой работы.	Творческая работа № 1. "Циклы"	2	3
43.	Практические занятия Выполнение творческой работы.	Творческая работа № 1. "Циклы"	2	3
44.	Теоретические занятия Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Практические занятия Рекурсивные функции. Вычисление факториала.	Создание функций	2	3
45.	Теоретические занятия Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Практические занятия Рекурсивные функции. Вычисление факториала.	Локальные переменные	2	3
46.	Практические занятия Примеры решения задач с использованием функций.	Решения задач	2	3
47.	Теоретические занятия Выполнение самостоятельной работы.	Самостоятельная работа № 3	2	3
48.	Практические занятия Выполнение самостоятельной работы.	Самостоятельная работа № 3	2	3
49.	Теоретические занятия Локальные и глобальные переменные. Поток выполнения. Практические занятия Решение практических задач с рекурсивными функциями	Рекурсивные функции	2	3
50.	Практические занятия Выполнение теста.	Тест № 5. Функции	2	3
51.	Теоретические занятия Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.	Строки	2	3

	<i>Практические занятия</i> Решение практических задач, создание сток			
52.	<i>Теоретические занятия</i> Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов <i>Практические занятия</i> Решение практических задач, создание сток	Срезы строк	2	3
53.	<i>Теоретические занятия</i> Разбор примеров решения задач со строками <i>Практические занятия</i> Написание программы для обнаружения робота черной линии.	Примеры решения задач со строками	2	3
54.	<i>Практические занятия</i> Решение задач.	Решения задач со строками	2	3
55.	<i>Практические занятия</i> Решение задач.	Решения задач со строками	2	3
56.	<i>Теоретические занятия</i> Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. <i>Практические занятия</i> Решение практических задач, создание списков	Списки	2	3
57.	<i>Практические занятия</i> Сборка и программирование робота для движения вдоль черной линии.	Срезы списков	2	3
58.	<i>Практические занятия</i> Сборка и программирование робота для движения вдоль черной линии.	Списки: примеры решения задач	2	3
59.	<i>Теоретические занятия</i> Примеры задач со списками <i>Практические занятия</i> Решение задач.	Решение задач со списками	2	3
60.	<i>Практические занятия</i> Решение задач.	Решение задач со списками.	2	3
61.	<i>Теоретические занятия</i> Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей.	Матрицы.	2	3
62.	<i>Практические занятия</i> Словарные операции.	Списки, Кортежи	2	3
63.	<i>Практические занятия</i> Словарные операции.	Введение в словари.	2	3
64.	<i>Теоретические занятия</i> Множества в языке Python. Множества. <i>Практические занятия</i> Операции, допустимые над множествами	Множества в языке Python	2	3
65.	<i>Теоретические занятия</i> Стиль программирования. <i>Практические занятия</i> Практическая работа по программированию	Стиль программирования	2	3
66.	<i>Теоретические занятия</i> Отладка программ <i>Практические занятия</i> Практическая работа по программированию	Отладка программ	2	3
67.	<i>Теоретические занятия</i> Зачетная работа	Зачет по курсу	2	3
68.	<i>Теоретические занятия</i> Перспективы и планы дальнейшего изучения языка		2	3
Итого:			125	181
Всего:			306	

Содержание учебного плана

Вводное занятие (2 часа).

Теория: Знакомство с планом работы объединения, инструктаж по ТБ.

Практика: Опрос. Тренинг на командообразование.

Раздел 1. Знакомство с языком Python (26 часов)

Теория:

Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. 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 (31 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 (40 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 (76 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 работа 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 с 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 (34 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 (30 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 (37 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. Кортежи 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 (30 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.

Ожидаемые результаты

В рамках курса «Программирование 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.

Условия реализации программы Материально-техническое обеспечение:

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- канцтовары;

Информационное обеспечение:

персональный компьютер (на каждого участника);
мультимедийный проектор;
оргтехника;
выход в сеть Internet;

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 ГбDDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19”;

Программное обеспечение:

Операционная система Windows 7 Профессиональная или выше;
Интерпретатор Python версии 3.7 и выше;
IDE JatBrainsPyCharm;
Foxit Reader или другой просмотрщик PDF файлов;
WinRAR;
Пакет офисных программ;
Adobe Photoshop или другой растровый графический редактор;
Любой браузер для интернет серфинга.

Характеристика помещений для занятий по программе

Теоретические занятия проходят в компьютерном классе, который соответствует нормам СанПиН, оборудованном современными ПК с ЖК-мониторами, мультимедийным проектором, стендами с литературой..

Информационное обеспечение

- видеоматериалы разной тематики по программе;
-цифровые разработки педагога к занятиям (презентации, сайты, тесты и т.д.)

Кадровое обеспечение

Занятия по программе может вести педагог дополнительного образования владеющий ИКТ, учитель информатики, физики, технологии, а также и совсем не педагог, а обычный инженер.

Определение результативности

Формы аттестации (контроля)

Входящий контроль осуществляется при комплектовании группы в начале учебного года. *Цель* - определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися.

Форма контроля: тестирование.

Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

Промежуточный контроль осуществляется в конце I полугодия учебного года.

Форма контроля: тест, решение практических задач средствами языка программирования.

Итоговый контроль осуществляется в конце учебного года.

Форма контроля: защита творческого проекта.

Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

Формы отслеживания и фиксации образовательных результатов

Формы определения результативности обучения Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий. Формы аттестации и оценочные материалы В процессе реализации программы предусмотрены следующие виды контроля:

- входной контроль проводится с целью определения уровня знаний учащихся (Приложение № 1);

- промежуточный контроль проводится регулярно на занятиях с

целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;

- итоговый контроль – защита проекта.

Формы подведения итогов реализации программы Результаты обучения по программе выявляются по итогам проведения олимпиад, соревнований по программированию, защиты проекта

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а так же их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам: - алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;

- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).

- Проверка достигаемых учащимися результатов производится в следующих формах:

- текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- текущая диагностика и оценка педагогом деятельности учащихся;
- итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальными и возрастными особенностями;
- Итоговый контроль проводится в конце всего курса. Он состоит из тестирования и решения практической задачи, защиты творческого проекта.

Формы предъявления и демонстрации образовательных результатов Аналитический материал по итогам проведения диагностики, защита творческих работ, конкурс, научно-практическая конференция, открытое занятие.

Перечень (пакет) диагностических методик

Входной контроль: *Тестирование (Приложение 1)*

Промежуточный контроль: *Тестирование, решение задач (Приложение 2)*

Итоговый контроль: *учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.*

Краткое описание методики работы по программе Особенности организации образовательного процесса – очная форма.

Методы обучения:

1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.

2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.
3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.
4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Формы организации образовательного процесса

Индивидуально-групповая и групповая

Формы организации учебного занятия

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать личностно-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности.

Решение данных задач кроется в организации курса деятельностного подхода к обучению, в проблемном изложении материала педагогом, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном обучении учащихся навыкам самостоятельной творческой деятельности.

Теория преподносится в форме беседы, эвристической беседы, презентации, обзора и т.п.

Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования и защиты проектов, проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.

Педагогические технологии

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.
2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.
3. Игровая технология. Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.
4. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

Алгоритм учебного занятия

Теоретические занятия строятся следующим образом:

- заполняется журнал присутствующих на занятиях обучаемых;
- объявляется тема занятий;
- раздаются материалы для самостоятельной работы и повторения материала или указывается, где можно взять этот материал;
- теоретический материал преподаватель дает обучаемым, помимо вербального, классического метода преподавания, при помощи различных современных технологий в образовании (аудио, видео лекции, экранные видео лекции, презентации, интернет, электронные учебники);
- проверка полученных знаний осуществляется при помощи тестирования обучаемых. Практические занятия проводятся следующим образом:

преподаватель показывает конечный результат занятия, т.е. заранее готовит практическую работу;

- далее преподаватель показывает, используя различные варианты, последовательность работы;
- преподаватель отдает обучаемым, ранее подготовленные самостоятельно мультимедийные материалы по изучаемой теме, либо показывает где они размещены на его сайте посвященном именно этой теме;
- далее обучаемые самостоятельно (и, или) в группах разрабатывают проект;
- весь процесс работы преподаватель снимает на видео, ранее установленную в аудитории; - видеоматериалы выкладываются на сайт в качестве поощрения и повторения материала, материалы так или иначе становятся методическим материалом, который можно в дальнейшем использовать в учебном процессе;
- практические занятия начинаются с правил техники безопасности при работе с различным инструментом и с электричеством и разбора допущенных ошибок во время занятия в обязательном порядке.

Дидактические материалы

- специальная и научно-популярная литература для педагога и учащихся;
- раздаточные материалы, инструкционные, технологические карты, задания, упражнения, образцы изделий и т.п.

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.

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.