

Филиал муниципального общеобразовательного учреждения «Средняя общеобразовательная школа им. Г.И. Марчука р.п. Духовницкое Духовницкого района Саратовской области» в с. Липовка Духовницкого района Саратовской области

Рассмотрено на заседании Педагогического совета Протокол № 7 от 29 марта 2024 г	«Утверждаю» Директор МОУ «СОШ им. Г.И. Марчука р.п. Духовницкое» Фролова Т.А. Приказ № _____ от _____ 2024 г
---	--



Дополнительная общеобразовательная
общеразвивающая программа
«Программирование на языке Python»

Направленность: естественнонаучная
Срок реализации программы: 1 год
Объем программы: 72 часа
Возраст детей: 13-15 лет

Федулов Андрей Юрьевич
педагог дополнительного образования

с. Липовка, 2024 год

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

1.1. Пояснительная записка

Задача построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического 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 состоит в том, что активизация познавательного процесса позволяет учащимся более полно выражать 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 внутрипрофильной 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.).

Практическая значимость в рамках предлагаемого курса «Программирование на языке Python» изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых

общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Особую роль программирование служит для формирования мыслительных и психических процессов учащихся (внимание, память, логика), освоения 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.

Адресат программы

Дополнительная общеразвивающая программа «Программирование на языке Python» рассчитана на детей 13–15 лет, проявляющих интерес к информационно-коммуникационным технологиям.

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.

Особенности организации образовательного процесса:

- Объем программы: 72 часа.
- Срок реализации программы: 1 год.
- Режим обучения: 1 раз в неделю по 2 часа.
- Форма обучения: очная.
- Особенности набора: свободный.
- Вид группы: среднешкольная.
- Состав группы: постоянный.

Методические принципы программы:

- принцип развития: развитие индивидуальных способностей, общей культуры, навыков творческой продуктивной деятельности обучающихся;
- принцип демократизма: право каждого субъекта системы технического творчества на выбор своей траектории развития;
- принцип дифференциации и индивидуализации образования: выявление и развитие способностей обучающихся в области технического творчества, обеспечение их развития в соответствии с потенциалом, индивидуальными возможностями и интересами;
- принцип культуросообразности: ориентация на потребности общества и личности учащихся, единство человека и социокультурной среды, адаптация детей к современным условиям жизни общества;
- принцип единства учебного и воспитательного процесса;
- принцип систематичности и последовательности: вначале используется репродуктивный и интерактивный методы усвоения знаний с постепенным введением проблемного метода обучения, метода проекта.

Формы обучения и занятий

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 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 курсе – обучение 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.

1.2. Цель и задачи программы

Цель: способствовать формированию творческой 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

1. Образовательные:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- 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);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

2. Воспитательные:

- воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;
- сформировать культуру занятий, направленную на воспитание личностных и социальных качеств.

3. Развивающие:

- развить познавательные процессы (внимание, восприятие, логическое мышление, память);
- развить креативность;
- развить способности к самореализации.

1.3. Содержание программы

Учебный план

№ п/п	Наименование тем	Количество часов			Формы аттестации / контроля
		Всего	Теория	Практика	
Раздел 1. Знакомство с языком		6	4	2	Срезовая работа, решение практических задач.
1.1	Общие сведения о языке Python	2	2		
1.2	Установка среды программирования	2	1	1	
1.3	Знакомство с IDE	2	1	1	
Раздел 2. Базовые понятия		10	4	6	Срезовая работа, решение практических задач.
2.1	Имена, переменные	2	1	1	
2.2	Операция присваивания	2	1	1	
2.3	Ввод и вывод	2	1	1	
2.4	Типы данных, конвертация типов	2	1	1	
2.5	Операции с целыми числами	2		2	
Раздел 3. Условный оператор		24	10	14	Срезовая работа, решение практических задач.
3.1	Величины логического типа	4	2	2	
3.2	Логические операции и выражения	4	2	2	
3.3	Условный оператор	6	2	4	
3.4	Множественное ветвление	4	2	2	
3.5	Сложные алгоритмы ветвления	6	2	4	
Раздел 4. Циклы		32	12	20	Срезовая работа, решение практических задач, зачёт.
4.1	Оператор цикла while	8	4	4	
4.2	Оператор цикла for	8	4	4	
4.3	Сочетание цикла и условия	4	1	4	
4.4	Операции break и continue	4	1	4	
4.5	Вложенные циклы	8	2	4	
ВСЕГО		72	30	42	

Содержание учебного плана

Раздел 1. Знакомство с языком (6 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. Базовые понятия (10 часов)

Теория: Типы данных . Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. 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. Условный оператор (24 часа)

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».

Зачётная 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 (32 часа)

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 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 цикла с параметром для решения поставленной задачи.

1.4. Планируемые результаты освоения программы

Планируемые результаты курса

В рамках курса «Программирование на языке Python» учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python.

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 также 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 в ходе обучения 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.

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

2.1. Календарный учебный график

Год обучения (уровень)	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год обучения	1 сентября 2024 г	05 июня 2025 г.	36	36	72	1 раз в неделю по 2 часа

2.2. Условия реализации программы

Материально-техническое обеспечение:

- столы для компьютера;
- стулья.

Информационное обеспечение:

- персональный компьютер на каждого ученика;
- мультимедийный проектор;
- выход в сеть Internet;

Аппаратное обеспечение компьютерной техники:

- процессор не ниже Core 2 Duo;
- объем оперативной памяти не ниже 4 Гб DDR3;
- дисковое пространство на менее 128 Гб;
- монитор диагональю на мене 19 дюймов.

Программное обеспечение:

- операционная система Linux;
- интерпретатор Python версии 3.9 и выше;
- интегрированная среда программирования JatBrains PyCharm Community;
- пакет офисных программ LibreOffice Community;
- растровый графический редактор GIMP.

2.3. 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.

2.4. Оценочные материалы

Входящий 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.

2.5. Методическое обеспечение

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации и отображает следующие традиционные и современные методы и технологии обучения.

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.

Технологии обучения:

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.

2.6. Список литературы

Нормативно-правовые акты и документы

1. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
2. Приказ Министерства образования и науки РФ от 29.08.2013 №1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Письмо Министерства образования Российской Федерации от 18 июня 2003 №28-02-484/16 «Требования к содержанию и оформлению образовательных программ дополнительного образования детей».
4. Постановление Главного государственного санитарного врача Российской Федерации от 4.07.2014 г. «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
5. Стратегия инновационного развития Российской Федерации на период до 2020 года. Распоряжение правительства Российской Федерации от 08.12.2011 № 2227-р.
6. Концепция развития дополнительного образования детей. Распоряжение Правительства Российской Федерации от 24 апреля 2015 года № 729-р.
7. Стратегия развития воспитания в Российской Федерации на период до 2025 года». Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р.

Литература 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.

Литература 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 тренажёр 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. Как с помощью только одной палочки образовать на столе треугольник? Как с помощью только одной палочки образовать на столе треугольник? Как с помощью только одной палочки образовать на столе треугольник? Как с помощью только одной палочки образовать на столе треугольник?

Входящий контроль

Низкий уровень: 0-70% выполненных заданий;

Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

Фамилия _____ Имя _____

1. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

--	--

2. Собака была привязана к десятиметровой верёвке, а прошла двести метров. Как ей это удалось?

--

3. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?

--

4. Как с помощью только одной палочки образовать на столе треугольник?

--

5. В каком месяце болтливая девочка говорит меньше всего?

--

6. Что становится на треть больше, если его поставить вверх ногами?

--

7. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?

--

8. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10. Сколько кусков пиццы получилось у Юры.

--

9. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела. Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

--

10. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

--	--

11. Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?

12. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?

13. Как с помощью только одной палочки образовать на столе треугольник?

14. В каком месяце болтливая девочка говорит меньше всего?

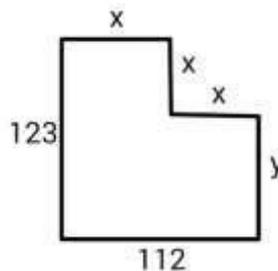
15. Что становится на треть больше, если его поставить вверх ногами?

16. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?

17. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10. **Вопрос:** Сколько кусков пиццы получилось у Юры.

18. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела. Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

19. Из прямоугольника вырезали квадрат со стороной x , получилась фигура как на рисунке. Чему равна сумма цифр y ?



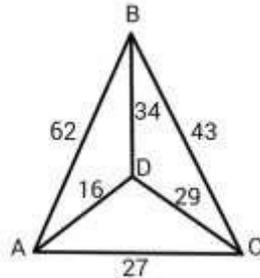
20. В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой. Каково максимальное количество слов в этом языке?

21. Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым —

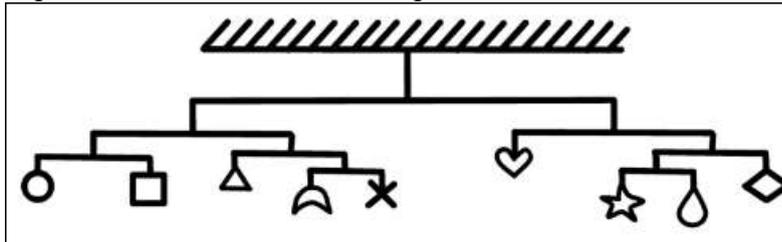
47 кусков, а если по зеленым — 31 кусок. Сколько кусков рулета получится, если

разрезать по кольцам всех трёх цветов? *Примечание:* учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

22. На рисунке показано расположение городов А, В, С и D и расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу. Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



23. Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке. Оказалось, что все её части находятся в равновесии. Сколько весит ромб?



24. Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

25. Сравните пары слов. Сколько среди них полностью идентичных?

O/Sanmarco	O/Samnarco
Ф.Wagonerrte	Ф.Wagonertre
A.S.Schmetterling	A.S.Schnetterling
N.V.Murfreesboroque	N.V.Munfreesboroque
P.S.Splendoursec	P.S.Sqlendoursec

Итоговый контроль

Диагностическая карта

№	Фамилия Имя	Входящая диагностика	Промежуточная диагностика	Итоговая диагностика
1.				
2.				
3.				
	Итого в %	Н С В		Н С В

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).

Низкий уровень

Учащиеся должны знать / понимать:

- технику безопасного поведения во время занятий;
- правила поведения в общественных местах;
- понятие программы Python; общую структуру программы;
- основные типы данных;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- основные циклы с условием;
- основные правила записи циклов условием;
- формат записи цикла с параметром;
- понятие функции;
- основные принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- способ передачи параметров;
- назначение строкового типа данных;
- операторы для работы со строками;
- операции со строками;
- способ описания списка;

- способ описания кортежа;
- способ описания словаря;
- основные 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.

Учащиеся должны уметь:

- понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога;
- работать в паре, малой группе;
- выполнить установку программы под руководством педагога;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе под руководством педагога;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием под руководством педагога;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога;
- создавать и использовать основные функции;
- описывать и соединять строки;
- находить подстроку в строке с помощью педагога;
- находить количество слов в строке;
- вводить и выводить элементы списка под руководством педагога;
- приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога;
- описывать множества под руководством педагога;
- определять принадлежность элемента множеству по наводящим вопросам;
- определять вид ошибок и находить ошибки в программе под руководством педагога;
- составлять элементарные алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

Средний уровень

Учащиеся должны знать / понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности;
- основные приемы взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python;
- 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;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов;
- понятие функции;
- основные способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров;
- 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;
- основные шаги работы над проектом, его презентации.

Учащиеся должны уметь:

- уважительно относиться к преподавателям и сверстникам;
- применять некоторые приемы логического (абстрактное) мышления;
- концентрировать внимание на одном или двух объектах;
- понимать причины успеха/неуспеха с помощью анализа педагога;
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- 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 использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- 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;
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать и сохранять цели и задачи учебной деятельности в ходе занятия;
- планировать свою деятельность с помощью взрослого;
- сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи;
- понимать причины успеха/неуспеха учебной деятельности под руководством взрослого;
- делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога;
- выражать творческие идеи, разработать творческий проект на основе образца;
- конструктивно взаимодействовать в составе группы в ходе работы над проектом.

Высокий уровень

Учащиеся должны знать / понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий;
- приёмы конструктивного взаимодействия в группе сверстников;
- понятие программы;
- структуру 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;

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров;
- 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;
- правила и этапы работы над проектом;
- приёмы успешной презентации проекта.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- 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 использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- 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;
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать учебную задачу, анализировать достижение результата;
- делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений;
- понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога;
- планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи;
- создать проектировочную команду и организовать её деятельность;
- разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;
- самостоятельно подготовить презентацию и защитить проект.

Вопросы для промежуточного контроля по усвоению материала

Промежуточный контроль проводится в последнюю неделю декабря.

1.	Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления: 11111·1111111 — произведение чисел 11111 (5 единиц) и 1111111 (7 единиц).
2.	Запишите число 1.2345e3 в виде десятичной дроби.
3.	Составьте и запишите выражение для вычисления: 2014.0^14 (Возвести в 14 степень). Обратите внимание на запись числа: это вещественное число.
4.	Приведите к целому типу число 2.99
5.	Расставьте скобки в выражении: a and b or not a and not b в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).
6.	Найдите результат выражения для заданных значений <i>a</i> и <i>b</i> . Учитывайте регистр символов при ответе. a = True b = False a and b or not a and not b
7.	Отметьте выражения, значения которых равны True: <ul style="list-style-type: none"> • "239" < "30" and 239 < 30 • "239" < "30" and 239 > 30 • "239" > "30" and 239 < 30 • "239" > "30" and 239 > 30

8.	Укажите результат выражения: "123" + "42"
9.	Какое значение будет у переменной <i>i</i> после выполнения фрагмента программы? <pre>i = 0 while i <= 10: i += 1 if i > 7: i += 2</pre>
10.	Сколько итераций цикла будет выполнено в этом фрагменте программы? <pre>i = 0 while i <= 10: i += 1 if i > 7: i += 2</pre>
11.	Сколько всего знаков * будет выведено после исполнения фрагмента программы: <pre>i = 0 while i < 5: print('*') if i % 2 == 0: print('**') if i > 2: print('***') i += 1</pre>
12.	Определите, какое значение будет иметь переменная <i>i</i> после выполнения следующего фрагмента программы: <pre>i = 0 s = 0 while i < 10: i += 1 s += i if s > 15: break i += 1</pre>

Задачи для промежуточного контроля по усвоению материала

1.	<p>Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.</p> <p>Поддерживаемые операции: +, -, /, *, mod, pow, div, где mod — это взятие остатка от деления, pow — возведение в степень, div — целочисленное деление.</p> <p>Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!".</p> <p>Обратите внимание, что на вход программе приходят вещественные числа.</p>
----	--

2.	<p>Пусть автобусные билеты на маршруте имеют шестизначные номера от 111111 до 999999.</p> <p>Посчитайте количество счастливых билетов. Таким билетом считается тот, у которого сумма первых трёх цифр соответствует сумме последних трёх цифр, например: 764845.</p> <p>Также подсчитайте количество весёлых билетов. Таким билетом считается тот, у которого сумма половинок отличается на 1, например: 764844.</p>
3.	<p>Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу. На вход подаётся тип фигуры комнаты и соответствующие параметры. Программа должна вывести площадь получившейся комнаты. Для числа π в стране Малевии используют значение 3.14.</p> <p>Формат ввода, который используют Малевийцы:</p> <p>Треугольник a b c <i>где a, b и c — длины сторон треугольника</i></p> <p>Прямоугольник a b <i>где a и b — длины сторон прямоугольника</i></p> <p>Круг r <i>где r — радиус окружности</i></p>
4.	<p>Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число. На ввод могут подаваться и повторяющиеся числа.</p>
5.	<p>Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трёх цифр совпадает с суммой последних трёх цифр номера билета.</p> <p>Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.</p> <p>На вход программе подаётся строка из шести цифр. (Пример:123321)</p> <p>Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.</p>
6.	<p>Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и сразу после этого выводит сумму квадратов всех считанных чисел.</p> <p>Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, после этого считывание продолжать не нужно.</p> <p>В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.</p>

7.	<p>Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число n — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.</p> <p>Например, если $n = 7$, то программа должна вывести 1 2 2 3 3 3 4.</p>
8.	<p>Выведите таблицу размером $n \times n$, заполненную числами от 1 до n^2 по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере (здесь $n=5$)</p> <pre> 1 2 3 4 5 16 17 18 19 6 15 24 25 20 7 14 23 22 21 8 13 12 11 10 9 </pre>