Министерство образования и науки Мурманской области Государственное автономное учреждение дополнительного образования Мурманской области «Мурманский областной центр дополнительного образования «Лапландия»

АТКНИЧП

методическим советом

Протокол

OTO1.06.2021 № 44

А.Ю. Решетова

УТВЕРЖДЕНА приказом ГАУДО МО «МОЦДО «Лапландия»

#10.06 2021 № 677

иректор **Пессия** С.В. Кулаков

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ «Основы программирования на языке Python»

Возраст учащихся: **12 – 17 лет** Срок реализации программы: **4 месяца**

Автор-составитель: Пантюхова Нина Александровна, педагог дополнительного образования

Пояснительная записка

Программа «Основы программирования на языке Python» имеет *техническую направленность*, в её основу заложены принципы модульности и практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на детальное изучение алгоритмизации, реализацию межпредметных связей, организацию проектной и исследовательской деятельности обучающихся.

Программа разработана в соответствии с:

- Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства просвещения Российской Федерации от 09.11.2018 г. № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» с изменениями, утверждёнными приказом Министерства просвещения РФ от 30.09.2020 №533 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Письмом Министерства образования и науки РФ от 25.07.2016 № 09-1790 «Рекомендации по совершенствованию дополнительных образовательных программ, созданию детских технопарков, центров молодежного инновационного творчества и внедрению иных форм подготовки детей и молодежи по программам инженерной направленности»;
- Письмом Министерства образования и науки Российской Федерации от 18 ноября 2015 года № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ»;
- Распоряжением Правительства Российской Федерации от 24 апреля 2015 года № 729-р «Концепция развития дополнительного образования детей»;
- Распоряжением Правительства Российской Федерации от 29 мая 2015 года N 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;
- Постановлением Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей, и молодежи»;
- Постановлением Главного государственного санитарного врача РФ от 28.01.2021 №2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

Актуальность программы обусловлена потребностью общества в технически грамотных специалистах и полностью отвечает социальному заказу по подготовке квалифицированных кадров в области программирования, а также высоким интересом подростков к ІТ-сфере. Важнейшей характеристикой подрастающего поколения является активность в информационном пространстве, интернет-коммуникации.

Педагогическая целесообразность обусловлена тем, что научившись программировать на языке Python, обучающиеся получат мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит обучающимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Цель программы: личностное и интеллектуальное развитие обучающихся в процессе обучения основам программирования на языке Python.

Задачи программы:

1. Обучающие

– изучение конструкций языка программирования Python;

- знакомство с принципами и методами функционального и объектно-ориентированного программирования; основными структурами данных и типовыми методами обработки этих структур;
- приобретение навыков работы в интегрированной среде разработки на языке Python.

2. Развивающие

- формирование навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;
- развитие у обучающихся интереса к программированию, самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- формирование и развитие навыков работы с различными источниками информации, необходимой для решения учебных задач; умения планировать свои действия с учётом фактора времени, в обстановке с элементами конкуренции, предвидеть результат и достигать его, при необходимости вносить коррективы в первоначальный замысел.

3. Воспитательные

- воспитание этики групповой работы, отношений делового сотрудничества, взаимоуважения;
- развитие основ коммуникативных отношений внутри проектных групп и в коллективе в целом;
- воспитание упорства в достижении результата;
- создание условий для самоопределения обучающихся в профессиональном выборе.

Ответительная особенность программы «Основы программирования на языке Python» в том, что она является практико-ориентированной. Освоение подростками IT-навыков происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и уверенно овладевать IT-технологиями, что поможет им самоопределиться и выстроить траекторию личностного роста в современном информационном обществе.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки вебприложений до машинного обучения.

Уровень программы: вводный

Возраст обучающихся, участвующих в реализации программы: 13-17 лет

Форма реализации программы – очная.

Срок реализации программы (модуля): 4 месяца

Объем программы – 36 часов.

Количество обучающихся в группе: 10-12 человек.

Форма организации занятий – групповая, при работе над проектами – групповая, парная.

Режим занятий: 1 раз в неделю по 2 академических часа.

Виды учебных занятий и работ: практические работы, беседы, лекции, конкурсы.

Ожидаемые результаты.

Предметные результаты:

- знание необходимой терминологии («информация», «алгоритм», «исполнитель», «модель»), смысла этих понятий и умение применять полученные знания на практике;
- развитие у обучающихся алгоритмического (знакомство и навыки работы с простейшими алгоритмическими структурами – линейной, условной и циклической), а также логического

мышления, что необходимо для грамотного составления алгоритмов, рассчитанных для конкретного исполнителя;

- знание основных понятий и этапов проектной деятельности;
- навыки пошагового выполнения алгоритмов управления исполнителями и анализа числовых и текстовых данных, умение осуществлять данные операции как вручную, так и с использованием компьютера;
- знания основ программирования и областей применения полученных навыков.

Личностные результаты:

- развитие у обучающихся внимания, сосредоточенности, терпения;
- использование принципов здоровьесбережения, а также отработка на практике принципов индивидуального и коллективного безопасного поведения при работе с компьютерной техникой;
- формирование осознанного уважительного отношения к другому человеку, его мнению, своему и чужому труду, бережное отношение к используемому оборудованию;
- развитие коммуникативных навыков, умения работать в команде сверстников в процессе образовательной, учебно-исследовательской и проектной деятельности.

Метапредметные результаты:

- умение самостоятельно планировать последовательность своих действий для достижения поставленных целей, а также грамотно распределять свое время и ресурсы для получения максимально эффективного результата;
- умение работать в паре и в коллективе, выстраивать совместную деятельность как с педагогом, так и со сверстниками;
- умение осуществлять самостоятельный поиск информации, анализировать и обобщать её;
- способность к принятию решений, а также умение формулировать, аргументировать и отстаивать своё мнение.

Учебный план

No	Раздел программы	Теория	Практика	Всего	Формы
п/п				часов	аттестации/контроля
	Классификация языков	1	1	2	
	программирования. Компиляция,				практическое задание
	интерпретация. Среда разработки				
	Знакомство с языком Python.	2	2	4	программный
	Синтаксис. Документация.				продукт
	Создание и запуск скриптов				
	Встроенные типы данных.	3	5	8	программный
	Переменные. Основные операции				продукт
	с данными				
	Ввод/вывод данных в программу	2	2	4	программный
					продукт
	Понятие алгоритм. Типы	2	2	4	программный
	алгоритмов. Линейный алгоритм и				продукт
	с условием. Условный оператор				
	Типы ошибок в	2	2	4	программный
	программировании. Поиск				продукт
	ошибок. Отладка программы.				
	Обработка исключений в Python				
	Управляющие инструкции –	3	7	10	программный
	циклы				продукт
	Итого	19	27	36	

Содержание программы

1. Классификация языков программирования. Компиляция, интерпретация. Среда разработки (2 ч.).

Теория:

Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим. Официальный сайт языка Python. Дистрибутивы языка Python.

Практика:

- 1. Дистрибутивы интерпретатора языка Python.
- 2. Установка официального дистрибутива интерпретатора языка Python 3.x.
- 3. Запуск интерактивного режима Python.
- 4. Работа в официальном IDLE Python.
- 5. Работа в среде PyScripter, настройка, запуск/остановка скриптов на языке Python.

2. Знакомство с языком Python. Синтаксис. Документация. Создание и запуск скриптов (4 ч.). *Теория:*

Документация на язык программирования. Стиль написания кода: отступы, базовый синтаксис. Имена переменных, длинна строки. Рекомендации по созданию имен переменных. Ключевые (служебные/зарезервированные слова). Первая программа. Сообщения интерпретатора.

Практика:

- 1. Первая программа.
- 2. Подсветка синтаксиса в средах IDLE.
- 3. Сообщения интерпретатора.
- 4. Знакомство с функциями print() и input().
- 5. Выполнение команд в интерактивном режиме.
- 6. Создание скриптов. Сохранение скриптов.
- 7. Программа приветствие.

3. Встроенные типы данных. Переменные. Основные операции с данными (8 ч.).

Теория:

Основные типы данных в программировании. Особенность оперирования данными в языке Python. Переменные, соотношение имени переменной со значением в памяти компьютера. Числовые типы данных. Преобразования числовых типов. Ограничение точности вычислений при работе с вещественными (float) типом данных – IEEE754. Основы строкового типа данных. Склеивание строк. Логический тип данных. Принцип высказываний. Базовые операции И, ИЛИ, НЕ.

Практика:

- 1. Основные операции с числовыми данными.
- 2. Базовые функции языка Python для работы с числовыми данными.
- 3. Ввод/вывод числовых данных.
- 4. Ограничение точности вычислений при работе с вещественными (float) типом данных IEEE754.
- 5. Склеивание строк.
- 6. Использование логических операций.
- 7. Взаимодействие с числовыми и строковыми типами данных.
- 8. Битовые операции.

4. Ввод/вывод данных в программу (4 ч.).

Теория:

Область видимости переменных. Присваивание значения переменных в коде программы. Ввод значения переменных пользователем с помощью функции input(). Функция print() для вывода данных на экран и в файл. Формирование строки. Вывод псевдографики.

Практика:

- 1. Использование функции input().
- 2. Преобразование типов вводимых данных.
- 3. Использование функции print().

- 4. Формирование строки.
- 5. Вывод псевдографики.

5. Понятие алгоритм. Типы алгоритмов. Линейный алгоритм и с условием. Условный оператор (4 ч.).

Теория:

Понятие алгоритма их типы. Описание задачи в виде алгоритма. Алгоритмы с ветвлением. Условный оператор. Синтаксис условного оператора. Вложенные условия. Множественные условия. Ленивая оценка условий. Сложные условия. Формулировка условий.

Практика:

- 1. Применение условного оператора if...else для решения прикладных задач.
- 2. Вложенные условия.
- 3. Множественные условия.
- 4. Ленивая оценка условий.
- 5. Сложные условия.
- 6. Формулировка условий.
- 7. Создание текстовой игры «Волк, коза и капуста» с использованием только условного оператора.

6. Типы ошибок в программировании. Поиск ошибок. Отладка программы. Обработка исключений в Python (4 ч.).

Теория:

Отладка программ. Типы ошибок: синтаксические, ошибки выполнения, семантические. Сообщения интерпретатора об ошибках. Поиск документации по ошибкам. Использование исключений для обработки ошибок выполнения без аварийного завершения программы. Разработка алгоритма программ с учетом возможных ошибок выполнения. Отладка программы с семантическими ошибками.

Практика:

- 1. Сообщения интерпретатора об ошибках.
- 2. Поиск документации по ошибкам.
- 3. Разработка алгоритма программ с учетом возможных ошибок выполнения.
- 4. Отладка программы с семантическими ошибками: экспериментальная отладка, создание контрольных меток.

7. Управляющие инструкции – циклы (10 ч.).

Теория:

Циклические алгоритмы. Циклы с предусловием и постусловием. Счетчик итераций. Цикл for, синтаксис. Функция range(). Команды break и continue. Вложенные циклы for. Цикл while, синтаксис. Счетчик итераций. Команды break и continue. Вложенные циклы. Бесконечный цикл. Формулировка условия. Составление оптимального алгоритма программы.

Практика:

- 1. Решение задач с помощью цикла for.
- 2. Бесконечный цикл.
- 3. Формулировка условия.
- 4. Решение задач с использованием циклов while.
- 5. Решение задач различного типа с помощью циклических алгоритмов.

Комплекс организационно-педагогических условий Календарный учебный график (см. Приложение 1)

Материально-техническое обеспечение

Для реализации программы имеется компьютерный класс, мультимедийный проектор, необходимое программное обеспечение (среда программирования языка Python, операционная система Windows, пакет Microsoft Office), компьютеры, принтер, интернет.

Информационное обеспечение

Официальный сайт языка Python - https://www.python.org/

Интерактивный учебник языка Питон - http://pythontutor.ru/

Python 3 для начинающих - https://pythonworld.ru/

База знаний, примеры по программированию на языке Python - https://pythonru.com

Сайт «Школа программиста» — https://acmp.ru/

Подборка материалов по языку Python - https://proglib.io/p/learning-python/

Выполнение программы онлайн - https://rextester.com/l/python3

Методическое обеспечение

Учебно-методические средства обучения:

- специализированная литература по направлению, подборка журналов,
- наборы технической документации к программному обеспечению,
- плакаты, фото и видеоматериалы,
- учебно-методические пособия для педагога и учащихся, включающие дидактический, информационный, справочный материалы на различных носителях, компьютерное и видео оборудование.

Применяемое на занятиях дидактическое и учебно-методическое обеспечение включает в себя электронные пособия, справочные материалы, программное обеспечение, используемое для обеспечения учебной и проектной деятельности, ресурсы сети Интернет.

Педагогические технологии, которые применяются при работе с учащимися

Название	цель				
Технология личностно- ориентированного обучения.	Развитие индивидуальных технических способностей на пути профессионального самоопределения учащихся.				
Технология развивающего обучения.	Развитие личности и ее способностей через вовлечение в различные виды деятельности.				
Технология проблемного обучения.	Развитие познавательной активности, самостоятельности учащихся.				
Технология дифференцированного обучения.	Создание оптимальных условий для выявления задатков, развития интересов и способностей, используя методы индивидуального обучения.				
Здоровьесберегающие технологии	Создание оптимальных условий для сохранения здоровья учащихся.				

Диагностика результативности образовательного процесса

- В течение всего периода реализации программы по определению уровня ее усвоения учащимися, осуществляются диагностические срезы:
- 1. Входной контроль посредством бесед, анкетирования, тестов, где выясняется начальный уровень знаний, умений и навыков учащихся, а также выявляются их творческие способности.

Входной контроль может проводиться в следующих формах: творческие работы, самостоятельные работы, вопросники, тестирование и пр.

- 2. Промежуточный контроль позволяет выявить достигнутый на данном этапе уровень ЗУН учащихся, в соответствии с пройденным материалом программы. Проводятся контрольные тесты, опросы, беседы, выполнение практических заданий.
- 3. *Итоговый контроль* проводится по окончании программы и предполагает комплексную проверку образовательных результатов по всем ключевым направлениям. Данный контроль позволяет проанализировать степень усвоения программы учащимися. Результаты контроля фиксируются в диагностической карте.

Критерии оценки результатов аттестации обучающихся

Общими критериями оценки результативности обучения являются:

- оценка уровня теоретических знаний: широта кругозора, свобода восприятия теоретической информации, развитость практических навыков работы со специальной литературой, осмысленность и свобода использования специальной терминологии;
- оценка уровня практической подготовки учащихся: соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением, качество выполнения практического задания, технологичность практической деятельности;
- оценка уровня развития и воспитанности обучающихся: культура организации самостоятельной деятельности, аккуратность и ответственность при работе, развитость специальных способностей, умение взаимодействовать с членами коллектива.

Возможные уровни теоретической подготовки обучающихся:

<u>Высокий уровень</u> – учащийся освоил практически весь объем знаний (80- 100%), предусмотренных программой за конкретный период; специальные термины употребляет осознанно и в полном соответствии с их содержанием.

<u>Средний уровень</u> – у учащегося объем освоенных знаний составляет 50-79%; сочетает специальную терминологию с бытовой.

<u>Низкий уровень</u> – учащийся овладел менее чем 50% объема знаний, предусмотренных программой; учащийся, как правило, избегает употреблять специальные термины.

Возможные уровни практической подготовки обучающихся:

<u>Высокий уровень</u> — учащийся овладел 80-100% умениями и навыками, предусмотренными программой за конкретный период; работает с оборудованием самостоятельно, не испытывает особых трудностей; выполняет практические задания с элементами творчества.

<u>Средний уровень</u> – у учащегося объем усвоенных умений и навыков составляет 50-79%; работает с оборудованием с помощью педагога; в основном выполняет задания на основе образца.

<u>Низкий уровень</u> – учащийся овладел менее чем 50% умений и навыков, предусмотренных программой; испытывает затруднения при работе с оборудованием; обучающийся в состоянии выполнять лишь простейшие практические задания педагога.

- В целях определения уровня усвоения программы учащимися осуществляются диагностические срезы:
- входная диагностика на основе анализа выбранной обучающимися роли в диагностической игре и степени их участия в реализации отдельных ее этапов, где выясняется начальный уровень знаний, умений и навыков учащихся, а также выявляются их творческие способности.
- промежуточная диагностика позволяет выявить достигнутый на данном этапе уровень знаний, умений и навыков учащихся, в соответствии с реализованной проектной деятельностью. Предлагаются выполнение практических заданий, контрольные тесты.
- итоговая диагностика проводится в конце учебного курса (выставка и защита творческих проектов) и предполагает комплексную проверку образовательных результатов по всем ключевым направлениям. Данный контроль позволяет проанализировать степень усвоения программы

учащимися.

Достигнутые учащимся знания, умения и навыки заносятся в сводную таблицу результатов обучения.

Сводная таблица результатов обучения по модулю

по образовательной программе дополнительного образования детей

Педагог д/о	
группа №	

№ п/п	ФИ учащегося	Теоретичес кие знания	Практические умения и навыки	Творческие способности	Воспитатель ные результаты	Итого
1.						
2.						
3.						

Формы подведения итогов реализации дополнительной программы: участие во внутренних мероприятиях мини-технопарка, муниципальных и областных мероприятиях, защита проекта и создание прототипа или групповые соревнования.

Достигнутые учащимся знания, умения и навыки заносятся в сводную таблицу результатов обучения.

Оценка уровней освоения модуля

Уровни	Параметры	Показатели			
Высокий уровень (80-100%)	Практические умения и навыки.	Обучающийся освоил материал в полном объеме. Знает и понимает значение терминов, самостоятельно ориентируется в содержании материала по темам. учащийся заинтересован, проявляет устойчивое внимание к выполнению заданий. Учащийся способен применять практические умения и навыки во время выполнения самостоятельных заданий.			
	Vauathyustanasua	Правильно и по назначению применяет инструменты. Работу аккуратно доводит до конца. Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.			
	Конструкторские способности.	Учащийся способен узнать и выделить объект (конструкцию, устройство). Учащийся способен собрать объект из готовых частей или построить с помощью инструментов. Учащийся способен выделять составные части объекта. Учащийся способен видоизменить или преобразовать объект по заданным параметрам. Учащийся способен из преобразованного или Видоизмененного объекта, или его отдельных частей собрать новый.			
Средний уровень (50-79%)	Теоретические знания.	Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу. Учащийся заинтересован, но не всегда проявляет устойчивое внимание к выполнению задания.			

	Практические	Учащийся владеет базовыми навыками и умениями, но не
	умения и навыки.	всегда может выполнить самостоятельное задание,
		затрудняется и просит помощи педагога. В работе допускает
		небрежность, делает ошибки, но может устранить их после
		наводящих вопросов или самостоятельно. Оценить
		результаты своей деятельности может с подсказкой педагога.
	Конструкторские	Учащийся может узнать и выделить объект (конструкцию,
	способности.	устройство).
		Учащийся не всегда способен самостоятельно разобрать,
		выделить составные части конструкции.
		Учащийся не способен видоизменить или преобразовать
		объект по заданным параметрам без подсказки педагога.
Низкий	Теоретические	Учащийся владеет минимальными знаниями, ориентируется
уровень	знания.	в содержании материала по темам только с помощью
(меньше		педагога.
50%)	Практические	Учащийся владеет минимальными начальными навыками и
	умения и навыки.	умениями. Учащийся способен выполнять каждую
		операцию только с подсказкой педагога или товарищей. Не
		всегда правильно применяет необходимый инструмент или
		на использует вовсе. В работе допускает грубые ошибки,
		не может их найти их даже после указания. Не способен
		самостоятельно оценить результаты своей работы.
	Конструкторские	Учащийся с подсказкой педагога может узнать и выделить
	способности.	объект (конструкцию, устройство).
		Учащийся с подсказкой педагога способен выделять
		составные части объекта.
		Разобрать, выделить составные части конструкции,
		видоизменить или преобразовать объект по заданным
		видоизменить или преобразовать объект по заданным параметрам может только в совместной работе с педагогом.

Литература для обучающихся

Allen Downey. Думать на языке Python. Green Tea Press. 2012. Перевод на русский язык Николай Орехов 2017. https://bitbucket.org/thinkpython_ru/ book/src

Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.

Васильев А.Н. Python на примерах. Практический курс по программированию. - СПб.: Наука и Техника, 2016. - 432 с.: ил.

Доусен М. Программируем на Python. - СПб.: Питер, 2014. - 416 с.: ил.

Пейн, Брайсон. Python для детей и родителей / Брайсон Пейн. – Москва: Издательство «Э». 2017. – 352 с.: ил.

Хайнеман, Джордж, Пояяис, Гэри, Сеяков, Стэнли. Алгоритмы. Справочник с примерами на С, С++, Java и Python, 2-е изд.: Пер. с англ. — СпБ.: ООО —Альфа-книга , 2017. — 432 с .: ил. — Парал. тит. англ.

Седжвик, Роберт, Уэйн, Кевин, Дондеро, Роберт. Программирование на языке Pyt hon: учебный курс. : Пер. с англ. - СПб. : ООО "Альфа-книга": 201 7. - 736 с. : ил. - Парал. тит. англ.

У. Сэнд, К. Сэнд. Hello World! Занимательное программирование. — СПб.: Питер, 2016. — 400 с.: ил. — (Серия «Вы и ваш ребенок»).

Литература для педагога

- 1. Бизли Д. М. Язык программирования Python : справочник :пер. с англ. / Д. М. Бизли. Киев : ДиаСофт, 2000
- 2. Гифт Н. Руthon в системном администрировании UNIX и Linux : пер. с англ. / Н. Гифт, Д. Джонс. СПб. : Символ-Плюс, 2009
- 3. Лейнингем И. Освой самостоятельно Python за 24 часа : пер. с англ. / И. Лейнингем. М. : Издательский дом «Вильямс», 2001
- 4. Лесса А. Руthon. Руководство разработчика: пер. с англ. / А. Лесса. СПб.: ДиасофтЮП, 2001
- 5. Лутц М. Изучаем Python: пер. с англ. / М. Лутц. СПб.: Символ-Плюс, 2009
- 6. Лутц М. Программирование на Python: пер.с англ. / М. Лутц. СПб.: Символ-Плюс, 2002
- 7. Саммерфельд М. Программирование на Python 3 Подробное руководство : пер. с англ. / М. Саммерфельд. СПб. : Символ-Плюс, 2009
- 8. Сузи Р. А. Руthon / Р. А. Сузи. СПб. : БХВ-Петербург, 2002
- 9. Сузи Р. А. Язык Python и его применения : учеб. пособие / Р.А. Сузи. М. : Интернет-Университет информационных технологий: БИНОМ. Лаборатория знаний, 2006
- 10. Язык программирования Python / Г. Россум [и др.]. СПб. : АНО «Институт логики» Невский диалект, 2001
- 11. Сэнд У., Сэнд К. Hello World! Занимательное программирование. СПб.: Питер, 2016
- 12. Шоу, Зед. Легкий способ выучить Python / Зед Шоу ; [пер. с англ. М. А. Райтмана]. М: Издательство «Э», 2017.
- 13. Мэтиз Эрик. Изучаем Python. Программирование игр, визуализация данных, вебприложения. СПб.: Питер, 2017.
- 14. Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений / Н. А. Прохоренок, В. А. Дронов. СПб.: БХВ-Петербург, 2016
- 15. Саммерфилд М.Программирование на Python 3. Подробное руководство. СПб.:Символ-Плюс 2009

- 16. Златопольский Д. М. Основы программирования на языке Python. М.: ДМК Пресс, 2017
- 17. Пэйн, Брайсон Python для детей и родителей. М.: «Э», 2017
- 18. Лучано Рамальо Python. К вершинам мастерства / Пер. с англ. Слинкин А. А. М.: ДМК Пресс, 2016
- 19. Гифт Н., Джонс Д. Python в системном администрировании UNIX и Linux. СПб.: Символ-Плюс, 2009
- 20. М.Э. Абрамян. 1000 задач по программированию. Часть І. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.
- 21. М.Э. Абрамян. 1000 задач по программированию. Часть II. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.
- 22. М.Э. Абрамян. 1000 задач по программированию. Часть III. Методические указания для студентов механико-математического, физического и экономического факультетов, ГОУ ПО РФ Ростовский государственный университет, 2004.
- 23. Златопольский Д.М. Сборник задач по программированию. 3-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2011. 304 с.: ил. (ИиИКТ)

Приложение 1 к программе «Основы программирования на языке Python»

Календарный учебный график

Педагог:

Количество учебных недель: 18

Режим проведения занятий: 2 раза в неделю по 2 часа

Праздничные и выходные дни (согласно государственному календарю)

04.11.2020, 01.01.2021-08.01.2021, 23.02.2021, 08.03.2021, 01.05.2021, 09.05.2021

Каникулярный период:

- осенние каникулы с 29 октября 2020 по 04 ноября 2020;
- зимние каникулы с 28 декабря 2020 по 08 января 2021;
- весенние каникулы с 25 марта 2021 по 31марта 2021;
- дополнительные каникулы с 19 февраля 2021 по 22 февраля 2021;
- летние каникулы с 01 июня 2021 по 31 августа 2021.
- летние каникулы.

Во время каникул занятия в объединениях проводятся в соответствии с учебным планом, допускается изменение расписания.

№ п/п	Дата	Время проведения занятия	Форма занятия	Кол- во часов	Тема занятия	Место проведен ия	Форма контроля
1.			Очная	2	Классификация языков программирования. Компилируемые и интерпретируемые языки. Области применения. Язык программирования Руthon. Области применения. Дистрибутивы интерпретатора языка Руthon. Установка официального дистрибутива интерпретатора языка Руthon 3.х. Запуск интерактивного режима Руthon. Работа в официальном IDLE Python. Работа в среде РуScripter, запуск/остановка скриптов на языке Руthon.		
2.			Очная	2	Документация на язык программирования – PEP8. Стиль написания кода: отступы. Имена переменных, длинна строки. Рекомендации по созданию имен переменных. Комментарии. Ключевые (служебные/зарезервированные слова). Вызов справки, поиск информации на ресурсах.		
3.			Очная	2	Первая программа. Подсветка синтаксиса в средах IDLE. Сообщения интерпретатора. Знакомство с функциями print() и input().		

	1	I	I	T ==	1	
				Выполнение команд в		
				интерактивном режиме.		
				Создание скриптов.		
				Сохранение скриптов.		
				Программа приветствие.		
4.		Очная	2	Основные типы данных в		
		0 1114	_	программировании.		
				Особенность оперирования		
				данными в языке Python.		
				Переменные, соотношение		
				имени переменной со		
				значением в памяти		
				компьютера. Числовые типы		
				данных. Преобразования		
				числовых типов. Основные		
				операции с числовыми		
				данными.		
5.		Очная	2	Базовые функции языка		
٥.		О-шал		Python для работы с		
				*		
				числовыми данными.		
				Ввод/вывод числовых		
				данных. Ограничение		
				точности вычислений при		
				работе с вещественными		
				(float) типом данных –		
				IEEE754. Основы строкового		
				типа данных.		
6.		Очная	2	Склеивание строк.		
0.		O III	_	Логический тип данных.		
				Принцип высказываний.		
				Базовые операции И, ИЛИ,		
				НЕ. Примеры использования		
		_	_	логических операций.		
7.		Очная	2	Использование логических		
				операций. Взаимодействие с		
				числовыми и строковыми		
				типами данных. Битовые		
				операции.		
8.		Очная	2	Область видимости		
J.			_	переменных. Присваивание		
				значения переменных в коде		
				программы. Ввод значения		
				переменных пользователем с		
				помощью функции input().		
				Использование функции		
				input(). Преобразование		
				типов вводимых данных.		
9.		Очная	2	Функция print() для вывода		
				данных на экран и в файл.		
				Использование функции		
				print(). Формирование		
				строки. Вывод		
				псевдографики.		
10.		Очная	2			
10.		Канко		Понятие алгоритма их типы.		
				Описание задачи в виде		
				алгоритма. Алгоритмы с		
				ветвлением. Условный		
				оператор. Синтаксис		
				условного оператора.		
11.		 Очная	2	Применение условного		
		<u> </u>	l		ı	

		1	T	.0 1	1	
				оператора ifelse для		
				решения прикладных задач.		
				Вложенные условия.		
				Множественные условия.		
				Ленивая оценка условий.		
				Сложные условия.		
				Формулировка условий.		
				Создание текстовой игры		
				«Волк, коза и капуста» с		
				использованием только		
				условного оператора.		
12.		Очная	2	Отладка программ. Типы		
				ошибок: синтаксические,		
				ошибки выполнения,		
				семантические.		
				Использование исключений		
				для обработки ошибок		
				выполнения без аварийного		
				1 -		
12		Orres -	2	завершения программы.		
13.		Очная	2	Разработка алгоритма		
				программ с учетом		
				возможных ошибок		
				выполнения. Сообщения		
				интерпретатора об ошибках.		
				Поиск документации по		
				ошибкам.		
				Отладка программы с		
				семантическими ошибками:		
				экспериментальная отладка,		
				создание контрольных		
				меток.		
14.		Очная	2	Циклические алгоритмы.		
				Циклы с предусловием и		
				постусловием. Счетчик		
				итераций.		
				Цикл for, синтаксис.		
				Функция range(). Команды		
				break и continue.		
15.		Очная	2	Цикл while, синтаксис.		
13.		Опил		Счетчик итераций. Команды		
				break и continue. Вложенные		
1.6		Ovvva -	2	ЦИКЛЫ.		
16.		Очная	2	Бесконечный цикл.		
				Формулировка условия.		
				Решение задач с		
				использованием циклов		
				while. Решение задач		
				различного типа с помощью		
				циклических алгоритмов.		
17.		Очная	2	Решение задач различного		
				типа с помощью		
				циклических алгоритмов.		
18.		Очная	2	Решение задач различного		
10.			_	типа с помощью		
				циклических алгоритмов.		
<u> </u>		<u> </u>	<u> </u>	дими южих алгоритмов.		