

ВОЛХОВСКИЙ МУНИЦИПАЛЬНЫЙ РАЙОН ЛЕНИНГРАДСКОЙ ОБЛАСТИ
МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ «ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ -
ЦЕНТР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

ПРИНЯТА
на заседании педагогического совета
протокол от 30.08.2024 № 1

УТВЕРЖДЕНА
приказом от 30.08.2024 №66 ОД

**Дополнительная общеразвивающая программа
технической направленности
«Мастерская программирования на языке Python»**

Возраст обучающихся: 13-18 лет Срок реализации: 2 года

Составитель программы
педагог дополнительного
образования Пятаков Андрей
Александрович

Волхов
2024 г.

Оглавление

Пояснительная записка	3
Содержание программы «Мастерская программирования на языке Python»	8
Тематическое планирование курса	17
Организационно-педагогические условия реализации программы.....	24
Программа воспитания	24
Основные направления самоанализа воспитательной работы.....	26
План воспитательной работы 2024-2025 учебный год.	27
Список литературы.....	29
Оценочные материалы.....	30

Пояснительная записка

Нормативно-правовые документы:

- Закон Российской Федерации от 29.12.2012г. №273-ФЗ «Об образовании в Российской Федерации»;
- Концепции развития дополнительного образования детей до 2030 года (утверждена Распоряжением Правительства Российской Федерации от 31.03.2022года № 678-р),
- Приказа Министерства Просвещения Российской Федерации от 27 июля 2022 года № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»
- Положения «О дополнительной развивающей программе» МБУДО «Центр информационных технологий»

Общая характеристика программы

Направленность программы: техническая.

Программа «Мастерская программирования на Python» - технологический курс дополнительного образования для обучающихся 7-11 классов. Программа дополнительного образования «Мастерская программирования на Python» нацелена на то, чтобы каждый обучающийся смог эффективно использовать современные компьютерные технологии в учебной, творческой, самостоятельной, досуговой деятельности. Программа способствует развитию познавательных интересов и творческих способностей обучающихся, удовлетворению их индивидуальных потребностей в интеллектуальном, нравственном, физическом совершенствовании. Рабочая учебная программа курса имеет практическую направленность по развитию ИКТ-компетентности обучающихся.

Содержание курса разработано с учетом особенностей обучающихся, основано на распределении учебного времени объемом в 68 часов в год на лекционные и практические занятия.

Большое значение имеет формирование у обучающихся на занятиях адекватной самооценки и осознание перспектив будущей жизни, что особенно важно в определении дальнейшей социализации ребенка с ограниченными возможностями здоровья. Самооценка лежит в основе наиболее адекватного мотива учебной деятельности – мотива достижения. Формирование знаний и умений осуществляется для обучающихся на доступном уровне.

Курс связан с удовлетворением индивидуальных образовательных интересов обучающегося, приобретением знаний, востребованных на рынке труда. Этот курс – важное средство построения индивидуальной образовательной траектории ученика в зависимости от его интересов и последующих жизненных планов.

Целесообразность изучения данного курса определяется быстрым внедрением компьютерной техники в повседневную жизнь и переходом к новым технологиям обработки информации.

Изучая программирование, обучающиеся лучше понимают сущность работы

компьютеров, их возможности и границы применения. К обучающимся приходит понимание того, что компьютер является инструментом, которым должны управлять люди. Далеко не все станут профессиональными программистами, но все выиграют от того, что постигли природу программирования и поняли, что значит создавать собственные программы.

Программирование даёт необычайно сильный толчок для развития интеллекта в целом и одновременно придаёт благоприятную эмоциональную окраску работе. Всем, без исключения, это помогает развивать навыки логического мышления, а также помогает выработать привычку аккуратной и систематической работы, а одарённым детям (прежде всего в области точных наук) ещё даёт так необходимую им новую пищу для размышлений, поисков, раскрытия скрытых способностей. Формируемое при этом алгоритмическое мышление ценно само по себе. Программирование помогает лучше формулировать логику решения практически любой задачи (совсем не обязательно чисто вычислительной).

Дополнительные занятия по информационным технологиям, реализующие межпредметные связи, неразрывно связаны с дальнейшей социализацией ребенка.

Возможно прохождение отдельных тем без преподавателя, для чего в курсе составлены задания для самостоятельного выполнения.

Цель и задачи программы:

Цель программы: дать обучающимся любого уровня подготовленности достаточный объём теоретических и практических знаний в области алгоритмизации и программирования.

Задачи программы

Образовательные:

1. Приобретение знаний и навыков алгоритмизации учащимися в ее структурном варианте.
2. Освоение всевозможных методов решения задач, реализуемых на языке Python

Развивающие:

1. Развитие алгоритмического мышления обучающихся
2. Формирование алгоритмической культуры.

Воспитательные:

1. Формирование интереса к изучению профессии, связанной с программированием.
2. Воспитание целеустремленности, активной жизненной позиции.
3. Программа рассчитана на учеников 7-11 классов.

Режим занятий

Срок реализации программы – 2 года.

Количество часов в неделю:

В неделю – 2 часа, всего – 140 часов (часть 1 - 68 учебных часов + часть 2 - 72 учебных часов). Количество часов на изучение того или иного раздела может

варьироваться в зависимости от потребностей учеников.

Сроки начала и окончания учебного года.

Начало занятий – первый год обучения с 15 сентября; 2 год обучения с 1 сентября.

Рабочая программа предполагает дополнительную практическую работу (по возможности) - самостоятельно или под руководством учителя. На практическую деятельность учащихся отводится более 60% всего времени.

Формы проведения занятий – мастер-класс, выставка, творческий отчет, практическое занятие.

Формы организации деятельности - групповая, индивидуально-групповая, индивидуальная, по подгруппам.

При необходимости для организации образовательного процесса используются дистанционные формы обучения, как лекции, тесты, семинары, зачетные работы, практические работы, компьютерные проекты.

Основные методические идеи программы:

- ✓ обучение в активной деятельности. Все темы программы обучающиеся изучают на практике. Разрабатывают алгоритмы, изображают их графически, вводят и отлаживают программы в среде программирования, редактируют, анализируют результаты работы;
- ✓ программа обучения построена так, что каждая новая тема логически связана с предыдущей, то есть при изучении новой темы используются все знания и навыки, полученные на предыдущих этапах обучения;
- ✓ наличие блока «Проектная деятельность», в котором учащиеся занимаются поисковой, исследовательской и другими видами работ;
- ✓ активное использование Инфографики как одного из основных средств достижения основной цели обучения по курсу.

Методы и формы обучения:

В основу курса заложен системно-деятельностный подход, который обеспечивает:

- формирование готовности к саморазвитию и непрерывному образованию;
- проектирование и конструирование социальной среды развития обучающихся в системе образования;
- активную учебно - познавательную деятельность обучающихся;
- построение образовательного процесса с учётом индивидуальных возрастных, психологических и физиологических особенностей обучающихся.

В учебном процессе ученики используют преимущественно следующие виды деятельности: аналитическую, поисковую, практическую.

Планируемые результаты обучения

В результате освоения обучающиеся получат представление:

- о методах представления и алгоритмах обработки данных, дискретизации, о программной реализации алгоритмов;
- о математических и компьютерных моделях, их использовании,
- о различных видах программного обеспечения и задачах, решаемых с его помощью; о существовании вредоносного программного обеспечения и средствах защиты от него, о необходимости стандартизации в сфере информационно-коммуникационных технологий;

У учащихся будут сформированы:

- основы алгоритмической культуры;
- умение составлять несложные программы;
- навыки и умения, необходимые для работы с основными видами программных системы интернет-сервисов (с опорой на их применение на протяжении всего учебного процесса по различным предметам);
- навыки коммуникации с использованием современных средств ИКТ, включая дистанционное общение (с опорой на предшествующее использование в различных предметах),
- представления о необходимости учёта юридических аспектов использования ИКТ, о нормах информационной этики.

Программа нацелена на достижение личностных, метапредметных и предметных результатов в области информатики и информационно-коммуникационных технологий.

Личностные результаты:

1. формирование ответственного отношения к учению, готовности и способности учащегося к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
2. формирование целостного мировоззрения, соответствующего современному уровню общества;
3. развитие осознанного и ответственного отношения к собственным поступкам в сфере использования информации;
4. формирование коммуникативной компетентности в различных сферах деятельности.

Метапредметные результаты:

1. умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
2. владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

3. умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
4. умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
5. умение осознанно использовать речевые средства в соответствии с задачей коммуникации; владение устной и письменной речью.

Предметные результаты:

1. умение использовать термины «данные», «кодирование», «алгоритм», «программа» понимание различий между употреблением этих терминов в быденной речи, умение составлять алгоритмы управления исполнителями и записывать их на выбранном алгоритмическом языке (языке программирования);
2. умение использовать логические значения, операции и выражения с ними;
3. умение создавать и выполнять программы для решения несложных алгоритмических задач в выбранной среде программирования;
4. умение использовать готовые прикладные компьютерные программы и сервисы в выбранной специализации, умение работать с описаниями программ и сервисами;
5. навыки выбора способа представления данных в зависимости от поставленной задачи.

Предполагается участие обучающихся в конкурсах, выставках, олимпиадах и т.д.

Формы контроля и подведения итогов реализации программы

Стартовый уровень учащихся:

- начальные умения и навыки работы в текстовых редакторах, использование встроенной панели рисования для отображения блок-схем,
- регистрация в Skype и Google,
- умение работать с браузером,
- умение работать с почтовой программой.

Критерии оценивания обучающихся по курсу:

На курсе дополнительного образования «Мастерская программирования на Python» действует безоценочная система. Содержание и объем материала, подлежащего проверке, определяется программой. При проверке усвоения материала необходимо выявлять полноту, прочность усвоения учащимися теории и умение применять ее на практике в знакомых и незнакомых ситуациях.

Основными формами проверки знаний, умений и навыков учащихся являются

завершенные практические работы, тестирование, самостоятельная работа, устный опрос.

Результативность обучения обеспечивается применением различных форм, методов и приемов, которые тесно связаны между собой и дополняют друг друга. В ходе практической деятельности педагог тактично контролирует, советует, направляет учащихся. Большая часть занятий отводится практической работе, по окончании которой проходит обсуждение и анализ.

Текущий контроль уровня усвоения материала осуществляется по результатам выполнения учащимися заданий по каждому разделу курса. Задания составлены с учетом возраста учащихся. При безоценочной системе курса **итоговый контроль** самого обучающегося может быть реализован в форме итогового задания или теста.

Методы определения результата:

- педагогическое наблюдение;
- оценка продуктов творческой деятельности учащихся;
- беседы, тесты, опросы.

Формы определения результата:

выполнение зачетных заданий по пройденным темам; публикация программ на сайте среды программирования; "участие в проектной деятельности.

Формы подведения итогов:

- выполнение зачетных заданий по пройденным темам;
- публикация программ на сайте среды программирования; " участие в проектной деятельности.

Содержание программы «Мастерская программирования на языке Python»

Часть 1 включает 34 темы (68 часов)

Структура программы с описанием разделов

Раздел 1. Введение в язык программирования Python.

Тема 1. Понятие алгоритм и программа. Начальное знакомство с языком

Теоретическая часть: Понятие алгоритма и программы.

Практическая часть: Установка и работа в среде программирования Python.

Тема 2. Элементы языка. Структура программы. Операции и переменные. Типы данных

Теоретическая часть: Знакомство с основными типами переменных, синтаксисом языка программирования, основными процедурами ввода исходных данных и вывода результатов Практическая часть: Использование основных процедур ввода исходных данных и вывода результатов.

Тема 3. Ввод и вывод данных на языке Python. Ввод и отладка программ в среде

Теоретическая часть: Анализ возможных синтаксических ошибок.

Практическая часть: Создание элементарных программ ввода-вывода данных, работа со средой, отладка программ.

Раздел 2. Линейные алгоритмы и их реализация на языке Python

Тема 4. Линейные алгоритмы целочисленных данных и их реализация на Python

Теоретическая часть: Знакомство со структурой линейного алгоритма, правилами записи арифметических выражений. Различные типы данных, допустимые операции над ними и ресурсы оперативной памяти. Выполнение операции присваивания в ОП компьютера. Практическая часть: создание блок-схемы линейного алгоритма, практикум "Запиши арифметическое выражение на языке программирования"

Тема 5. Решение задач по теме «Обработка целочисленных данных»

Теоретическая часть: Анализ готовых линейных программ.

Практическая часть: Разработка линейных алгоритмов. Ввод и отладка программ, реализующих линейный алгоритм обработки целых чисел.

Тема 6. Линейные алгоритмы вещественных чисел и их реализация на Python

Теоретическая часть: Различные типы данных, допустимые операции над вещественными числами и ресурсы оперативной памяти.

Практическая часть: создание блок-схем алгоритма, практикум "Запиши арифметическое выражение на языке программирования"

Тема 7. Решение задач по теме «Обработка вещественных чисел»

Теоретическая часть: Анализ готовых линейных программ.

Практическая часть: Разработка линейных алгоритмов. Ввод и отладка программ, реализующих линейный алгоритм обработки вещественных чисел.

Тема 8. Самостоятельная работа по теме «Линейные алгоритмы и их реализация в среде Python».

Теоретическая часть: Анализ выполненной работы.

Практическая часть: составление алгоритма, написание программы, ввод и отладка программного кода, анализ результатов.

Раздел 3. Разветвляющиеся алгоритмы и их реализация на языке Python

Тема 9. Алгоритм «Выбор», графическое изображение, полное и неполное ветвление

Теоретическая часть: Понятие алгоритма «Выбор», графическое изображение.

Практическая часть: Составление алгоритма «Полное ветвление».

Тема 10. Реализация алгоритма выбор на Python. Примеры решения задач. Решение задач по теме «Полное и неполное ветвление»

Теоретическая часть: Типовые задачи, использующие алгоритм «Выбор», изображение алгоритма в виде блок-схемы. Составление и анализ алгоритмов.

Практическая часть: Ввод и отладка программ в среде Python.

Тема 11. Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления

Теоретическая часть: Составление и анализ алгоритмов. Практическая часть: Ввод и отладка программ в среде Python.

Тема 12. Множественный выбор и его реализация с помощью вложенных ветвлений.

Теоретическая часть: Понятие множественного выбора, изображение на блок-схеме.

Практическая часть: Ввод и отладка программ в Python.

Тема 13. Решение задач по теме. Сложные условия. Каскадные ветвления.

Теоретическая часть: Составление алгоритмов. Анализ синтаксических и логических ошибок. Практическая часть: Ввод и отладка программ в среде Python.

Тема 14. Разветвляющиеся алгоритмы. Зачетная работа

Теоретическая часть: Составление алгоритмов. Анализ синтаксических и логических ошибок.

Практическая часть: Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Раздел 4. Циклические алгоритмы и их реализация на языке Python.

Тема 15. Циклические алгоритмы. Цикл с предусловием.

Теоретическая часть: Понятие цикла с предусловием, графическое изображение.

Практическая часть: Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 16. Цикл с предусловием. Решение задач.

Теоретическая часть: Типовые задачи, решаемые с помощью цикла с предусловием.

Практическая часть: Составление блок-схем, написание программного кода, ввод и отладка программ, реализующих цикл с предусловием в среде Python.

Тема 17. Самостоятельная работа по теме «Циклы с предусловием».

Теоретическая часть: Анализ разработанного алгоритма.

Практическая часть: Ввод и отладка программ, реализующих цикл с предусловием в среде Python.

Тема 18. Циклические алгоритмы. Цикл с постусловием. Инструкции управления циклом. Решение задач

Теоретическая часть: Типовые задачи, решаемые с помощью цикла с постпроверкой.

Практическая часть: Составление блок-схем, написание программного кода, ввод и отладка программ, реализующих цикл с постпроверкой в Python.

Тема 19. Циклические алгоритмы. Цикл с параметром

Теоретическая часть: Понятие цикла с параметром, графическое изображение.

Практическая часть: Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 20. Циклы со параметром, решение задач.

Теоретическая часть: Разработка алгоритмов цикла с параметром, анализ логических ошибок.

Практическая часть: Ввод и отладка программ, реализующих цикл с параметром в среде Python.

Тема 21. Самостоятельная работа по теме «Циклы со счетчиком».

Теоретическая часть: Анализ разработанного алгоритма. Оценка эффективности разработанного алгоритма.

Практическая часть: Ввод и отладка программ, реализующих цикл с параметром в среде Python. Поиск синтаксических и логических ошибок в программах.

Тема 22. Сложные циклические алгоритмы. Вложенные циклы.

Теоретическая часть: Понятие вложенного цикла, графическое изображение.

Практическая часть: Ввод и отладка программ, реализующих вложенные циклы в среде Python.

Тема 23. Самостоятельная работа по циклам. Анализ самостоятельной работы.

Теоретическая часть: Разработка блок схем.

Практическая часть: Ввод и отладка программ, реализующих вложенные циклы в среде Python.

Раздел 5. Этапы решения задач на языке Python.

Тема 24. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма. Часть 1

Теоретическая часть: Разработка алгоритмов.

Практическая часть: Написание программного кода, ввод и отладка программ в среде Python.

Тема 25. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма. Часть 2

Теоретическая часть: Разработка алгоритмов.

Практическая часть: Написание программного кода, ввод и отладка программ в среде Python.

Тема 26. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма. Часть 3

Теоретическая часть: Разработка алгоритмов.

Практическая часть: Написание программного кода, ввод и отладка программ в среде Python.

Раздел 6. Вспомогательные алгоритмы и их реализация на языке Python.

Тема 27. Вспомогательные алгоритмы и подпрограммы. Функции

Теоретическая часть: Понятие вспомогательного алгоритма. Формат записи вспомогательного алгоритма в виде функции. Типовые задачи.

Практическая часть: Ввод и отладка программ с использованием функции в среде Python.

Тема 28. Вспомогательные алгоритмы и подпрограммы. Применение функций при решении задач. Часть 1

Теоретическая часть: Анализ разработанных алгоритмов.

Практическая часть: Ввод и отладка программ с функцией пользователя в среде программирования Python.

Тема 29. Вспомогательные алгоритмы и подпрограммы. Применение функций при решении задач. Часть 2

Теоретическая часть: Анализ разработанных алгоритмов.

Практическая часть: Ввод и отладка программ с функцией пользователя в среде программирования Python.

Тема 30. Вспомогательные алгоритмы и подпрограммы. Рекурсия.

Теоретическая часть: Понятие рекурсии. Типовые задачи.

Практическая часть: Ввод и отладка программ с использованием рекурсии в среде Python. *Тема 31. Вспомогательные алгоритмы и подпрограммы. Рекурсия. Решение задач*

Теоретическая часть: Разработка и анализ алгоритмов.

Практическая часть: Ввод и отладка программ с использованием рекурсии в среде Python.

Раздел 7. Итоги обучения по курсу.

Тема 32. Итоговая самостоятельная работа по теме «Реализация основных типов алгоритма»

Теоретическая часть: Анализ результатов работы.

Практическая часть: Составление алгоритма, написание программы, ввод и отладка программного кода.

Тема 33-34. Обзор пройденного материала. Подведение итогов обучения.

Теоретическая часть: Обзор пройденного материала. Практическая часть: Анализ выполненных проектов.

Часть 2 включает 38 тем (76 часов)

Раздел 1. Повторение

Тема 1. Основные типы алгоритмов. Реализация линейного алгоритма и выбора на языке Python.

Теоретическая часть: Типы алгоритмов и их графическое изображение.

Практическая часть: Решение задач по реализации линейного алгоритма и выбора на языке Python.

Тема 2. Повторение. Основные типы алгоритмов. Реализация циклического алгоритма на языке Python.

Теоретическая часть: Реализация циклического алгоритма на Python.

Практическая часть: Решение задач на циклы, ввод и отладка программ с циклами.

Тема 3. Повторение. Основные типы алгоритмов. Реализация вложенных циклов на языке Python.

Теоретическая часть: Вложенные циклы, графическое изображение. Практическая часть: Ввод и отладка программ с вложенными циклами.

Тема 4. Повторение. Основные типы алгоритмов. Реализация вспомогательных алгоритмов на языке Python.

Теоретическая часть: Реализация вспомогательных алгоритмов на Python.

Практическая часть: Ввод и отладка программ с функциями.

Раздел 2. Структурные типы данных. Списки

Тема 5. Структурные типы данных. Списки.

Теоретическая часть: Понятие списка. Заполнение списка значениями. Вывод списка на экран. Практическая часть: Ввод и отладка программ со списками.

Тема 6. Структурные типы данных. Обработка списков. Линейный поиск. Добавление и удаление элементов. Копирование списков

Теоретическая часть: Обработка списков. Типовые задачи.

Практическая часть: Ввод и отладка программ по линейному поиску, добавлению и удалению элементов, копированию списков.

Тема 7. Структурные типы данных. Обработка списков. Двоичный поиск и сортировка

Теоретическая часть: Методы поиска и сортировки в списках.

Практическая часть: Создание блок-схем алгоритма, ввод и отладка программ по поиску и сортировке в списках

Тема 8. Структурные типы данных. Решение задач по теме.

Теоретическая часть: Анализ проектов по теме.

Практическая часть: Разработка алгоритмов. Ввод и отладка программ по теме «Структурные типы данных»

Тема 9. Самостоятельная работа по теме "Списки". Анализ самостоятельной работы.

Теоретическая часть: Краткая характеристика заданий.

Практическая часть: составление алгоритма, написание программы, ввод и отладка программного кода, анализ результатов.

Раздел 3. Структурные типы данных. Символьные строки

Тема 10. Символьные строки. Операции обработки строк.

Теоретическая часть: Понятие символьных строк. Основные операции обработки.

Практическая часть: Решение задач по обработке строк, ввод и отладка программ.

Тема 11. Символьные строки. Типовые задачи обработки строк.

Теоретическая часть: Характеристика и классификация типовых задач по обработке строк. Практическая часть: Ввод и отладка программ в среде Python.

Тема 12. Решение задач по теме "Символьные строки".

Теоретическая часть: Составление и анализ алгоритмов. Практическая часть: Ввод и отладка программ в среде Python.

Тема 13. Самостоятельная работа по теме "Символьные строки". Анализ самостоятельной работы

Теоретическая часть: Составление алгоритмов. Анализ созданных проектов.

Практическая часть: Ввод и отладка программ в среде Python.

Раздел 4. Структурные типы данных. Матрицы

Тема 14. Матрицы. Основные понятия.

Теоретическая часть: Понятие матрицы. Методы ввода элементов. Практическая

часть: Разработка алгоритмов. Ввод и отладка программ в Python.

Тема 15. Матрицы. Типовые задачи обработки элементов.

Теоретическая часть: Анализ типовых задач обработки элементов матрицы.

Практическая часть: Ввод и отладка программ в среде Python.

Тема 16. Матрицы. Решение задач обработки элементов. Обработка исключений.

Теоретическая часть: Составление алгоритмов. Анализ синтаксических и логических ошибок.

Практическая часть: Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 17. Матрицы. Самостоятельная работа по теме. Анализ самостоятельной работы.

Теоретическая часть: Составление алгоритмов. Анализ синтаксических и логических ошибок. Практическая часть: Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Раздел 5. Структурные типы данных. Работа с файлами

Тема 18. Типы файлов с точки зрения программиста. Запись информации в файл, чтение из файла.

Теоретическая часть: Типы файлов, этапы обработки, режимы открытия.

Программирование файлового ввода/вывода. Текстовые и двоичные файлы. Этапы обработки файла.

Программирование обработки файлов. Методы считывания данных из файла.

Практическая часть: Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 19. Изменение файлов. Решение задач

Теоретическая часть: Типовые задачи по изменению файлов.

Практическая часть: Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 20. Самостоятельная работа по теме "Работа с файлами". Анализ самостоятельной работы

Теоретическая часть: Анализ составленного алгоритма.

Практическая часть: Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Раздел 6. Объектно – ориентированное программирование.

Тема 21. Объектно - ориентированное программирование. Концепция ООП. Объекты и классы.

Теоретическая часть: Последовательное конструирование алгоритма. Объектная модель решения задачи. Объектно-ориентированный анализ (ООА). Этапы ООА. Концепция ООП. Практическая часть: Разработка проектов по теме.

Тема 22. Объектно - ориентированное программирование. Программирование объектной модели

Теоретическая часть: Абстракция. Поля и методы класса. Программирование объектов и классов.

Практическая часть: Программирование объектной модели.

Тема 23. Объектно - ориентированное программирование. Принципы ООП: инкапсуляция.

Теоретическая часть: Определение инкапсуляции. Применение инкапсуляции в программировании.

Практическая часть: Программирование объектной модели, реализация класса в конкретной задаче.

Тема 24. Объектно - ориентированное программирование. Принципы ООП: наследование и полиморфизм.

Теоретическая часть: Принципы ООП: наследование и полиморфизм, применение в программировании.

Практическая часть: Программирование объектной модели, реализация класса в конкретной задаче с использованием наследования и полиморфизма.

Тема 25. Объектно - ориентированное программирование. Решение задач по теме

Теоретическая часть: Анализ разработанных проектов.

Практическая часть: Разработка проектов с использованием метода ООП.

Тема 26. Объектно - ориентированное программирование. Самостоятельная работа по теме, анализ работы

Теоретическая часть: Анализ разработанных проектов.

Практическая часть: Разработка проектов с использованием метода ООП.

Раздел 7. Приложения с графическим пользовательским интерфейсом.

Тема 27. Основы графического интерфейса.

Теоретическая часть: Современные прикладные программы. RAD-среды. Простейшая программа с графическим интерфейсом.

Практическая часть: Графические библиотеки для языка Python.

Тема 28. Создание приложения с использованием виджетов.

Теоретическая часть: Элементы управления: кнопки, метки, поля ввода. Создание и настройка окон сообщений.

Практическая часть: Создание приложения с графическим интерфейсом.

Тема 29. Создание графического интерфейса без использования программы визуализатора

Теоретическая часть: Создание окна приложения и запуск цикла обработки событий. Размещение на форме и настройка виджетов.

Практическая часть: Создание приложения с графическим интерфейсом.

Тема 30. Управление макетом графического интерфейса: блочный макет

Теоретическая часть: Макет графического интерфейса. Типы макетов. Блочный макет.

Практическая часть: Управление макетом графического интерфейса с блочным макетом.

Тема 31. Управление макетом графического интерфейса: сеточный макет

Теоретическая часть: Принцип работы сеточного макета. Программирование сеточного макета. Работа программы при изменении размеров ее окна.

Практическая часть: Управление макетом графического интерфейса с сеточным макетом. *Тема*

32. Рисование на форме

Теоретическая часть: Рисование при создании графического интерфейса. Инструменты для рисования. Программирование рисования. Практическая часть: Рисование на форме. Проект.

Тема 33. Модель обработки данных в приложении с графическим интерфейсом

Теоретическая часть: Декомпозиция приложения с графическим интерфейсом. Модель обработки данных приложения «Калькулятор».

Практическая часть: Создание приложения «Калькулятор».

Тема 34. Представление в приложении с графическим интерфейсом

Теоретическая часть: Представление приложения «Калькулятор». Контроллер в приложении с графическим интерфейсом.

Практическая часть: Создание приложения «Калькулятор».

Тема 35. Создание собственного виджета

Теоретическая часть: Программирование собственного виджета.

Практическая часть: Создание виджета для побитового отображения целых чисел.

Тема 36. Анализ проектов по теме "Приложения с графическим интерфейсом"

Теоретическая часть: Обзор проектов курса.

Практическая часть: Анализ выполненных проектов.

Раздел 8. Итоги обучения по курсу.

Тема 37. Итоговая самостоятельная работа курсу

Теоретическая часть: Анализ результатов работы.

Практическая часть: составление алгоритма, написание программы, ввод и отладка программного кода.

Тема 38. Обзор пройденного материала. Подведение итогов обучения.

Теоретическая часть: Обзор пройденного материала.

Практическая часть: Анализ выполненных проектов.

Тематическое планирование курса

«Мастерская программирования на языке Python»

1 год обучения (68 часов)

№ п / п	Название разделов и тем	Количество часов			Формы аттестации и (контроля)*
		Все го	Теор ия	П р а к т	
	Раздел 1. Введение в язык программирования Python.				
1	Тема 1. Понятие алгоритм и программа. Начальное знакомство с языком	2	1	1	Тест + Практическая работа
2	Тема 2. Элементы языка. Структура программы. Операции переменные. Типы данных	2	1	1	Практическая работа
3	Тема 3. Ввод и вывод данных на языке Python. Ввод и отладка программ в среде	2		2	Практическая работа
	Итого:	6	2	4	
	Раздел 2. Линейные алгоритмы и их реализация на языке Python				
4	Тема 4. Линейные алгоритмы целочисленных данных и их реализация на Python	2	1	1	Тест + Практическая работа
5	Тема 5. Решение задач по теме «Обработка целочисленных данных»	2		2	Практическая работа
6	Тема 6. Линейные алгоритмы вещественных чисел и их реализация на Python	2	1	1	Тест + Практическая работа

7	Тема 7. Решение задач по теме «Обработка вещественных чисел»	2		2	Практическая работа
8	Тема 8. Самостоятельная работа по теме «Линейные алгоритмы и их реализация в среде Python».	2		2	Практическая работа
	Итого:	10	2	8	
	Раздел 3. Разветвляющиеся алгоритмы				
9	Тема 9. Алгоритм «Выбор», графическое изображение, полное и неполное ветвление	2	1	1	Тест + Практическая работа
10	Тема 10. Реализация алгоритма выбор на Python. Примеры решения задач.	2			
			1	2	
11	Тема 11. Решение задач по теме «Полное и неполное ветвление»	2			работа
12	Тема 13. Множественный выбор и его реализация с помощью вложенных ветвлений.	2	1	1	Практическая работа
13	Тема 14. Решение задач по теме. Сложные условия. Каскадные ветвления.	10		2	Практическая работа
14	Тема 15. Разветвляющиеся алгоритмы. Зачетная работа. Анализ работы		1	1	Практическая работа
	Итого:	2	3	2	Практическая работа
	Раздел 4. Циклические алгоритмы и их реализация на Python.				
16	Тема 16. Циклические алгоритмы. Цикл с предусловием.	2	2	7	
17	Тема 17. Цикл с предусловием. Решение задач.	2			
18	Тема 20. Самостоятельная работа по теме «Циклы с предусловием».	2			Тесты + Практическая работа
19	Тема 21. Циклические алгоритмы. Цикл с постусловием. Инструкции управления циклом		1	3	Практическая работа
20	Тема 22. Циклические алгоритмы. Цикл с постусловием. Решение задач			2	Практическая работа
	Итого:	11	3	1	Тесты + Практическая работа

№ п / п	Название разделов и тем	Всего	Теория	Практика	Формы аттестации (контроля)*
21	Тема 23. Циклические алгоритмы. Цикл с параметром	2		8	Практические работы
22	Тема 24. Циклы со параметром, решение задач.	2			1
23	Тема 26. Самостоятельная работа по теме «Циклы со счетчиком».	2		1	Тесты + Практические работы
24	Тема 27. Сложные циклические алгоритмы. Вложенные циклы.		1	2	Практические работы
25	Тема 28. Самостоятельная работа по циклам. Анализ самостоятельной работы.			2	Тесты + Практические работы
	Итого:	10	2	1	Практические работы
	Раздел 5. Этапы решения задач на Python.	2		2	Тесты + Практические работы
26	Тема 29. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма. Часть 1,2	2	1	8	
27	Тема 30. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма. Часть 3,4	2			
28	Тема 31. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма. Часть 5,6			1	Практическая работа
	Итого:	6	1	2	Практическая работа

	Раздел 6. Вспомогательные алгоритмы и их реализация на Python.	2		2	Практическая работа
29	Тема 32. Вспомогательные алгоритмы и подпрограммы. Функции	2	1	5	
30	Тема 33. Вспомогательные алгоритмы и подпрограммы. Применение функций при решении задач. Часть 1,2	2	1		
31	Тема 34. Вспомогательные алгоритмы и подпрограммы. Применение функций при решении задач. Часть 3,4			1	Тесты + Практические работы
32	Тема 35. Вспомогательные алгоритмы и подпрограммы. Рекурсия.		1	1	Практические работы
33	Тема 36. Вспомогательные алгоритмы и подпрограммы. Рекурсия. Решение задач	2		2	Практическая работа
	Итого:	10	3	7	
	Раздел 7. Итоги обучения по курсу.				
34	Тема 37. Итоговая самостоятельная работа по теме «Реализация основных типов алгоритма»	3		3	Практическая работа
35	Тема 38. Обзор пройденного материала. Подведение итогов обучения.	2	1	1	Тесты + Практические работы
	Итого:	5	1	4	
	Всего:	68	17	51	

2 год обучения (68 часов)

Л п / п	Название разделов и тем	Количество часов			Формы аттестации и (контроля)*
		Всего	Теория	Практика	
	Раздел 1. Повторение				

1	Тема 1. Основные типы алгоритмов. Реализация линейного алгоритма и выбора на языке Python.	2	1	1	Тест + Практическая работа
2	Тема 2. Основные типы алгоритмов. Реализация циклического алгоритма на языке Python.	2	1	1	Практическая работа
3	Тема 3. Основные типы алгоритмов. Реализация вложенных циклов на языке Python.	2		2	Практическая работа
4	Тема 4. Основные типы алгоритмов. Реализация вспомогательных алгоритмов на языке Python.	2		1	Практическая работа
	Итого:	8	2	5	
	Раздел 2. Структурные типы данных. Списки				
5	Тема 5. Структурные типы данных. Списки.	2	1	1	Тест + Практическая работа
6	Тема 6. Структурные типы данных. Обработка списков. Линейный поиск. Добавление и удаление элементов. Копирование списков	2	1	1	Тест + Практическая работа
7	Тема 7. Структурные типы данных. Обработка списков. Двоичный поиск и сортировка	2	1	1	Тест + Практическая работа
8	Тема 8. Структурные типы данных. Решение задач по теме	2		2	Практическая работа
9	Тема 9. Самостоятельная работа по теме "Списки". Анализ самостоятельной работы	2		2	Практическая работа
	Итого:	10	3	7	
	Раздел 3. Структурные типы данных. Символьные строки				
10	Тема 10. Символьные строки. Операции обработки строк.	2	1	1	Практическая работа
11	Тема 11. Символьные строки. Типовые задачи обработки строк.	2	1	1	Практическая работа
12	Тема 12. Решение задач по теме "Символьные строки"	2		2	Тест + Практическая работа

1 3	Тема 13. Самостоятельная работа по теме "Символьные строки". Анализ самостоятельной работы	2		2	Практическая работа
	Итого:	8		2	6
	Раздел 4. Структурные типы данных. Матрицы				
1 4	Тема 14. Матрицы. Основные понятия.	2	1	1	Практические работы
1 5	Тема 15. Матрицы. Типовые задачи обработки элементов.	2	1	1	Тесты + Практические работы
1 6	Тема 16. Матрицы. Решение задач обработки элементов. Обработка исключений.	2		2	Тесты + Практические работы
1 7	Тема 17. Матрицы. Самостоятельная работа по теме. Анализ самостоятельной работы.	2		2	Практические работы
	Итого:	8	2	6	
	Раздел 5. Структурные типы данных. Работа с файлами				
1 8	Тема 18. Типы файлов с точки зрения программиста. Запись информации в файл, чтение из файла.	2	1	1	Практическая работа
1 9	Тема 19. Изменение файлов. Решение задач	2	1	1	Практическая работа
2 0	Тема 20. Самостоятельная работа по теме "Работа с файлами". Анализ самостоятельной работы	2		2	Практическая работа
	Итого:	6	2	4	
	Раздел 6. Объектно – ориентированное программирование.				
2 1	Тема 21. Объектно - ориентированное программирование. Концепция ООП. Объекты и классы.	2	2		Тесты + Практические работы
2 2	Тема 22. Объектно - ориентированное программирование. Программирование объектной модели	2	1	1	Тесты + Практические работы
2 3	Тема 23. Объектно - ориентированное программирование. Принципы ООП:	2	2		Тесты + Практические работы

	инкапсуляция.				
2 4	Тема 24. Объектно - ориентированное программирование. Принципы ООП: наследование и полиморфизм.	2	1	1	Тесты + Практическ иеработы
2 5	Тема 25. Объектно - ориентированное программирование. Решение задач по теме	2		2	Практическ ая работа
2 6	Тема 26. Объектно - ориентированное программирование. Самостоятельная работа по теме, анализ работы	2		2	Практическ ая работа
	Итого:	12	6	6	
	Раздел 7. Приложения с графическимпользовательским интерфейсом.				
2 7	Тема 27. Основы графического интерфейса.	2	2		Тесты + Практическ иеработы
2 8	Тема 28. Создание приложения с использованием виджетов.	2	1	1	Тесты + Практическ иеработы
2 9	Тема 29. Создание графического интерфейса без использования программы-визуализатора	2	1	1	Тесты + Практическ иеработы
3 0	Тема 30. Управление макетом графического интерфейса: блочный макет	2	1	1	Тесты + Практическ иеработы
3 1	Тема 31. Управление макетом графического интерфейса: сеточный макет	2	1	1	Тесты + Практическ иеработы
3 2	Тема 32. Рисование на форме	2	1	1	Тесты + Практическ иеработы
3 3	Тема 33. Модель обработки данных в приложении с графическим интерфейсом	2	1	1	Тесты + Практическ иеработы
3 4	Тема 34. Представление в приложении с графическиминтерфейсом	2	1	1	Тесты + Практическ иеработы
3 5	Тема 35. Создание собственного виджета	2		2	Практическ ая работа
3 6	Тема 36. Анализ проектов по теме "Приложения с графическим интерфейсом"	2		2	Практическ ая работа

	Итого:	20	9	11	
	Раздел 8. Итоги обучения по курсу.				
3 7	Тема 37. Итоговая самостоятельная работа по курсу	2		2	Практическая работа
3 8	Тема 38. Обзор пройденного материала. Подведение итогов обучения.	2	1	1	Тесты + Практические работы
	Итого:	4	1	3	
	Всего:	68	20	48	

Организационно-педагогические условия реализации программы

Учебно-методическое сопровождение программы

Методика преподавания курса предусматривает:

- ✓ проведение по каждой новой теме лекции;
- ✓ выполнение учащимися самостоятельного практического задания на каждом занятии;
- ✓ уровень задания определяется для каждого ученика индивидуально, с учетом возможностей, интересов и склонностей ребенка; при самостоятельном изучении курса, ученик сам выбирает уровень практической работы;
- ✓ для промежуточного и итогового контроля используются тестовые задания;
- ✓ прохождение курса сопровождается созданием учащимися проектов по предлагаемым темам.

Подобная организация обучения способствует решению основных задач современного образования - развитие познавательной активности и творческих способностей учащихся.

Программа воспитания

Цели и задачи воспитания

Цель воспитания в МБУДО «Центр информационных технологий» - воспитание социально-активной, творческой, нравственно и физически здоровой личности, способной на сознательный выбор жизненной позиции, а также к духовному и физическому самосовершенствованию, саморазвитию в социуме. Данная цель ориентирована на обеспечение положительной динамики личностного развития обучающихся:

- ✓ освоение социально значимых знаний и норм и приобретении опыта социального взаимодействия;
- ✓ формирование опыта самоопределения (личного или профессионального) в различных сферах жизни;

- ✓ формирование современных компетентностей и грамотностей, соответствующих стратегиям социально-экономического развития РФ, актуальным вызовам будущего.

Для достижения поставленной воспитательной цели необходимо решить следующие **задачи**:

1. использовать в воспитании обучающихся возможностей занятий по дополнительным общеобразовательным общеразвивающим программам, как источника поддержки и развития интереса детей к познанию и творчеству;
2. реализовывать потенциал событийного воспитания для формирования духовно-нравственных ценностей, укрепления и развития традиций детского объединения и образовательной организации, развития субъектной позиции обучающихся;
3. развивать социально-педагогическое партнерство МБУДО «Центр информационных технологий», для более эффективного достижения целей воспитания и социализации обучающихся;
4. поддерживать различные формы детской активности и самоуправления через развитие деятельности детских общественных объединений;
5. организовать содержательное партнерство с семьями обучающихся, их родителями (законными представителями) для более эффективного достижения целей воспитания.

Формы и методы воспитательной работы.

Главное в образовательном процессе дополнительного образования – успешность ребенка как результат педагогической деятельности, а мера этой успешности определяется только относительно личностного роста каждого ребенка.

Реализация воспитательного потенциала занятия предполагает следующее:

- использование воспитательных возможностей содержания учебного занятия по определенному направлению деятельности через демонстрацию детям примеров ответственного, гражданского поведения, проявления человеколюбия и добросердечности, через подбор соответствующих текстов для чтения, задач для решения, проблемных ситуаций для обсуждения в объединении;
- применение на занятии интерактивных форм работы обучающихся: интеллектуальных игр, стимулирующих познавательную мотивацию обучающихся; дискуссий, которые дают обучающимся возможность приобрести опыт ведения конструктивного диалога; групповой работы или работы в парах, которые учат командной работе и взаимодействию с другими детьми;
- включение в занятие игровых процедур, которые помогают поддержать мотивацию детей к получению знаний, налаживанию позитивных межличностных отношений в объединении, помогают установлению доброжелательной атмосферы во время занятия;
- организация шефства мотивированных и эрудированных обучающихся над их слабоуспевающими сверстниками, дающего обучающимся социально- значимый опыт сотrudничества и взаимной помощи;
- инициирование и поддержка исследовательской деятельности обучающихся в рамках реализации ими индивидуальных и групповых исследовательских творческих

проектов, что даст обучающимся возможность приобрести навык самостоятельного решения теоретической проблемы, навык генерирования и оформления собственных идей, навык уважительного отношения к чужим идеям, оформленным в работах других исследователей, навык публичного выступления перед аудиторией, аргументирования и отстаивания своей точки зрения.

Работа педагога со всем детским объединением включает в себя:

- инициирование и поддержку участия детского объединения в ключевых культурно-образовательных событиях образовательной организации, оказание необходимой помощи детям в их подготовке, проведении/ участии и анализе;
- организацию в творческом объединении интересных и полезных для личностного развития обучающихся совместных воспитательных событий, коллективных творческих дел, способствующих укреплению традиций, формирование и развитие коллектива, в том числе разновозрастного, а также способствующих самореализации детей и подростков и получение ими социального опыта, формирование поведенческих стереотипов, одобряемым в обществе;
- выработка с обучающимися детского объединения норм и правил совместной жизнедеятельности;
- создание условий для проявления инициатив по самоуправлению жизнедеятельностью детского объединения.

Индивидуальная работа педагога дополнительного образования с обучающимися детского объединения:

- изучение особенностей личностного развития обучающихся объединения через наблюдение за поведением, отношением к выбранному виду деятельности, взаимодействием и коммуникацией с другими обучающимися в специально создаваемых педагогических ситуациях, в организуемых педагогом беседах по тем или иным нравственно-этическим темам или событиям, участником которых стал ребенок;
- поддержка ребенка в решении важных для него жизненных проблем (налаживание взаимоотношений с другими детьми, личный и социальный опыт в конкретных видах и направлениях деятельности, в том числе в рамках программного содержания);
- коррекция поведения ребенка через индивидуальные беседы с ним, его родителями (законными представителями), с другими членами детского объединения; через привлечение узких специалистов для решения выявленных проблем.

Основные направления самоанализа воспитательной работы.

Основными направлениями анализа воспитательного процесса являются следующие показатели:

1. Результаты воспитания, социализации и саморазвития обучающихся (какова

динамика личностного развития обучающихся каждого объединения; какие прежде существовавшие проблемы личностного развития школьников удалось решить; какие проблемы решить не удалось и почему; какие новые проблемы появились, над чем далее предстоит работать?).

2. Воспитательная деятельность педагога (испытывает ли педагог проблемы с реализацией воспитательного потенциала совместной с детьми деятельности);

Итогом анализа организуемого воспитательного процесса является перечень выявленных проблем, над которыми предстоит работать в дальнейшем.

№	Аспекты исследования	Диагностические средства
1.	Уровень воспитанности	<ul style="list-style-type: none"> • Методика определения общественной активности обучающихся МОУ ДО «РЦВР» (составленная Е.Н.Степановым) • Методика «Акт добровольцев» (сост. Л.В.Байбородовой)
2.	Личностное развитие обучающихся МОУ ДО «РЦВР» в участии массовых мероприятий МОУ ДО «РЦВР»	<ul style="list-style-type: none"> • наблюдение • опрос • анализ
3.	Изучение качества воспитанности обучающихся	<ul style="list-style-type: none"> • Методика оценки воспитанности обучающихся
4.	Самооценка воспитанников	<ul style="list-style-type: none"> • Методика самооценки «Дерево» (ав. Д. Лампен, в адапт. Л.П. Пономаренко)

План воспитательной работы 2024-2025 учебный год.

	Мероприятия	Срок
1. Организация муниципальных конкурсов		
1.1	Ежегодный муниципальный конкурс	апрель
2. Участие в международных, республиканских, областных и муниципальных конкурсах и акциях		
2.3	«Отечество»	в соответ. с планом
2.5	Всероссийская акция «Час кода»	декабрь
2.6	Участие в муниципальном конкурсе: «Я исследователь»	по плану
2.7	Участие в интернет - каникулах	ноябрь, январь, март
3. Работа по формированию детского коллектива, органов детского самоуправления и выработке традиций учреждения		

3.1	Выборы Совета обучающихся, составление плана работы	октябрь
3.2	Конкурс между объединениями на лучший сайт или страничку в соц.сетях	ноябрь
3.4	Организация и проведение праздника по итогам года «Наши достижения»	май
3.5	Организация и проведение новогодних праздников.	декабрь
4. Работа по пропаганде здорового образа жизни и безопасности и по профилактике правонарушений		
4.1	Акции, посвященные Международному дню отказа от курения и Дню борьбы с курением Неделя здоровья, - Всемирный день здоровья.	2 раза в год
4.2	Проведение мероприятий по профилактике нарушений и безопасности в сети интернет	сентябрь-май
4.4	Участие во всероссийском уроке безопасности школьников в сети интернет	по плану
5. Проведение тематических занятий		
5.1	Викторина «День народного единства»	ноябрь
5.2	К Международному дню инвалидов «Уроки добра»	1-2 декабря
5.3	Познавательная игра, посвященная Дню рождения города Волхова	декабрь
5.4	Рождественская викторина	декабрь-январь
5.5	«Был город фронт, была блокада» конкурс презентаций	январь
5.6	Компьютерный рисунок «День защитника отечества»	февраль
5.7	Международная неделя информатики	март
5.8	Викторина «День космонавтики»	апрель
5.9	День Победы	май
6. Диагностика учебно-воспитательного процесса		
6.1	Проведение промежуточной и итоговой аттестации обучающихся	декабрь, май
6.2	Диагностика успешности учащихся в районных, республиканских и другого уровня конкурсах.	в течение года
6.3	Диагностика участия учащихся в культурно-массовых мероприятиях.	в течение года

Список литературы

Для педагога:

1. Бизли, Дэвид М. Python. Подробный справочник. – М.: Символ-Плюс, 2010. – 450 с.
2. Лутц М. Python. Справочник. – М.: Вильямс, 2015. – 503 с.
3. Майк МакГрат. Программирование на Python для начинающих. – М.: Эксмо, 2015. – 320 с.
4. Федоров Д. Ю. Основы программирования на примере языка Python: Учебное пособие. – Санкт-Петербург: 2016. – 318 с.
5. Сэнд У., Сэнд К. Hello World! Занимательное программирование на языке Python. – М.: – 2016. – 542 с.
6. Долинский М.С. Решение сложных и олимпиадных задач по программированию: Учебное пособие. – М.: – 2006. – 53 с.
7. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования Python. – М.: Эксмо, 2001. – 127 с.
8. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина: Учебное пособие. – М.: МПГУ, 2015. – 79 с.

Для обучающихся:

1. Майк МакГрат Программирование на Python для начинающих. – М.: Эксмо, 2015.
2. Федоров Д. Ю. Основы программирования на примере языка Python: Учебное пособие. – Санкт-Петербург: ИД «Питер», 2016.
3. Сэнд У., Сэнд К. Hello World! Занимательное программирование на языке Python. – М.: – 2016.
4. <http://kpolyakov.spb.ru/school/probook/python.htm> 5.
5. <http://informatics.mccme.ru/course/view.php?id=156>
6. Язык Python – учебник информатики. <http://kpolyakov.spb.ru/school/probook/python.htm>
7. Курс Д.П.Кириенко «Программирование на Python»

Для родителей: 1.

- Справочные материалы <https://metanit.com/python/>
2. Интерактивный сборник задач для практики программирования <http://pythontutor.ru/>
 3. Адаптивный тренажер Python <https://stepik.org/course/431>
 4. Среда разработки для языка Python <https://www.jetbrains.com/pycharm/?fromMenu>
 5. Проект Open Book Project. Практические примеры на Python Криса Мейерса openbookproject.net

Интернет-источники:

1. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
2. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
3. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
4. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
5. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431>, свободный.

6. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
7. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net, свободный.

Оценочные материалы

7.1. Промежуточная аттестация

Промежуточная аттестация проходит в форме компьютерного тестирования.

Выберите БУКВУ, соответствующую верному ответу: 1) К

простым типам данных не относится тип...

- А) целый
 Б) логический В) список
 Г) символьный Д) вещественный
- 2) К коллекциям относится ... А)
 целый тип
 Б) логический тип В) множество
 Г) символьный тип Д) вещественный тип
- 3) В результате выполнения оператора $a = 4 ** 2$, переменная a примет значение А) 2
 Б) 4 В) 8 Г) -2 Д) 16
- 4) В результате выполнения оператора $a = 254 \% 10$, переменная a примет значение А)
 25.4 Б) 4 В) 25 Г) 54 Д) 2540
- 5) Самая старшая логическая операция (выполняется первой) А)
 and Б) or В) not
- 6) Между этими символами записывается аргумент функции
 А) { } Б) [] В) () Г) // Д) ‘ ‘
- 7) Какой тип должен быть у переменной a для оператора присваивания $a = 25 / 3$? А) int
 Б) bool В) float Г) str
- 8) Укажите синтаксически неправильную запись операции присваивания: А) $x = (y + 1) / 3$
 Б) $x = (y) / 3 + 1$ В) $x + 1 = (y) / 3$ Г) $x = (y) / (3 + 1)$
- 9) В результате выполнения фрагмента программы $x = 4$
 $y = 5$
 $y = y + x ** 0.5$
 $\text{print}('y =', y)$
 на экран будет выведено

A) $y = 7$ Б) $y = y$ В) $y = 21$ Г) $y = 7.0$ Д) $y = 9$

10) В результате выполнения фрагмента программы $a =$
 'kvadrat'

$b = a[1:3]$

переменная b получит значение:

А) 'kv' Б) 'kva' В) 'va' Г) 'vad'

11) Обратным условием условию $x > 4$ будет являться

А) $x < 4$ Б) $x == 4$ В) $x <= 4$ Г) $x >= 4$

12) Простым условием является А) x

< 4 or $x == y$

Б) $x < 5$ and $y == 0$ or $x == y$ В) x

$== 4$ and $y == 0$

Г) not ($x >= 4$) Д) $x +$

$1 <= y - 4$

13) В результате выполнения фрагмента программы $y =$
 4.5

$x = 6$

if $y > 4$:

$x = x + 1$ else:

$x = y - 2$

переменная x будет иметь значение:

А) 6 Б) 7 В) 2.5 Г) 6.5

14) Для организации принудительного (досрочного) выхода из цикла используется оператор...

А) exit Б) close В) break Д) end

15) Для вызова начала новой итерации цикла, используется оператор... А)

for Б) continue В) open Д) end

16) В результате выполнения фрагмента программы $x = 3$

while $x < 9$: print

('УРА') $x = x + 2$

слово «УРА» будет напечатано:

А) 0 раз Б) 1 раз В) 2 раза

17) В результате выполнения фрагмента программы for a Г) 3 раза Д) 4 раза

in range(5):

print ('РОССИЯ')

слово «РОССИЯ» будет напечатано:

А) 1 раз Б) 0 раз В) 4 раза

18) В результате выполнения фрагмента программы for a Г) 5 раз Д) 3 раза

in 'qwerty':

print('ПРИВЕТ')

слово «ПРИВЕТ» будет напечатано

А) 1 раз Б) 0 раз В) 6 раз

Г) 5 раз Д) 4 раза

Выберите БУКВЫ, соответствующие верным ответам:

19) Укажите, какими способами значения переменных a и b поменяются местами

- A) $a = b$
 $b = a$
- Б) $a, b = b, a$
- В) $b = a$
 $c = a$
 $a = b$
- Г) $c = a$
 $a = b$
 $b = c$
- Д) $c = b$
 $c = a$
 $b = a$

20) Составными условиями являются

- A) $x < 4$
- Б) $x < 5$ and $y == 0$ or $x == y$
- В) $x == 4$
- Г) $\text{not } (x >= 4)$
- Д) $x <= 4$

21) Служебное слово else относится к оператору

- A) условия if
- Б) присваивания =
- В) цикла for
- Г) цикла while

22) Какие существуют типы переменных в Python (выбрать несколько вариантов):

- a) float
- b) str
- c) num
- d) bool
- e) integer
- f) real
- g) int

23) Имена переменных могут включать (выбрать несколько):

- a) Русские буквы
- b) Латинские буквы
- c) Пробелы
- d) Скобки, знаки + = ! ? и др.
- e) Знак подчёркивания ()
- f) Цифры

24) Какие имена являются правильными в Python (выбрать несколько):

- a) N
- b) ABC
- c) sum
- d) 41And
- e) A+B
- f) _mam

25) Какие операторы цикла существуют в языке Python?

- a) for
- b) while
- c) repeat ... until
- d) loop

26) Чувствителен ли Python к регистру (большая или маленькая буквы):

- a) Да
- b) Нет

27) Установите соответствие между типом переменной и зарезервированным словом:

- | | |
|----------------------------|----------|
| 1. вещественная переменная | a) int |
| 2. символьная строка | b) str |
| 3. логическая переменная | c) float |
| 4. целая переменная | d) bool |

Ответ: 1с, 2b, 3d, 4a

28) Установите соответствие между выполняемым действием и результатом его выполнения:

- | | |
|----------------|---------|
| 1. int("88") | a) "88" |
| 2. float("88") | b) 88 |
| 3. str(88.0) | c) 88.0 |

Ответ: 1b, 2с, 3a

Выберите БУКВУ, соответствующую верному ответу:а

29) Что будет в результате выполнения программы:

```

a = int(input())
b = int(input())
if a < b:
    print(a)
else:
    print(b)
если a = 10, b = 20?

```

- a) 10
- b) 20
- c) 30
- d) -10

30) Какой ряд чисел образуется после выполнения следующего алгоритма:

```

for i in range(1,10):
    print(i)

```

- a) 1 2 3 4 5 6 7 8 9 10
- b) 1 2 3 4 5 6 7 8 9
- c) 0 1 2 3 4 5 6 7 8

31) Что будет в результате выполнения следующего алгоритма программы:

```

a = int(input())
b = int(input())
if a % 10 == 0 or b % 10 == 0:
    print('YES')
else:
    print('NO')
если a = 15, b = 45?

```

- a) YES
- b) NO

32) Как будет записано число 18 после выполнения следующего алгоритма:

```
x = float(input())
print(x)
```

- a) 18
- b) 18.0
- c) 18.00

33) Результатом выполнения алгоритма цикла while будет:

```
i = 1
while i <= 10:
    print(i**2)
    i = i + 1
```

- a) 1 2 4 8 12 14
- b) 1 2 16 24 32
- c) 1 4 9 16 25 36 49 64 81 100

34) Определите, что будет напечатано в результате работы следующей программы

```
s = 0
for i in range(8, 13):
    s = s + 12
print(s)
```

Ответ: 60

35) Определите значение переменной с после выполнения следующего фрагмента:

```
a = 100
b = 30
a -= b * 3
if a > b:
    c = a - b
else:
```

```
    c = b - a
```

- a) 20
- b) 70
- c) -20
- d) 180

36) Условный оператор:

```
if a % 2 == 0:
    print('Да')
```

```
else:
```

```
    print('Нет')
```

позволяет определить, является ли число **a**:

- a) целым
- b) двузначным
- c) чётным
- d) простым
- e) нечетным

37) Цикл в фрагменте программы

```

p = 2
while p > 0.1:
    p *= 0.1

```

будет исполнен раз

- 0
- 1
- 2
- бесконечное число раз

38) Цикл в фрагменте программы:

```

a = b = 1
while a + b < 8:
    a += 1
    b += 2

```

выполнится раз:

- 0
- 2
- 3
- бесконечное число

39) Определите значения переменных s и i после выполнения фрагмента программы:

```

s = 0
i = 5
while i >= 0:
    s += i
    i -= 1

```

- $s = 0, i = -1$
- $s = 5, i = 0$
- $s = 15, i = 5$
- $s = 15, i = -1$

40) В данном фрагменте программы:

```

s = 0
for i in range(1, 11):
    s += 2 * i

```

вычисляется:

- сумма целых чисел от 1 до 10
- сумма чётных чисел от 1 до 10
- удвоенная сумма чисел от 1 до 10
- сумма первых десяти чётных чисел

Критерии оценки:

Правильные ответы выделены в тексте заливкой. Каждый правильный ответ оценивается в один балл. Максимальное количество баллов – 40. Набранные баллы переводятся в уровень освоения по следующей шкале:

- 10 – 19 баллов: низкий уровень;
- 20 – 29 баллов: средний уровень;
- 30 – 40 баллов: высокий уровень.

7.2. Итоговая аттестация

Правила выбора темы итогового проекта

Итоговым результатом освоения обучающимися полученных в процессе обучения навыков и компетенций в рамках представленной программы является итоговая защита проекта. Так как от выбора темы проекта зависит качество проделанной самостоятельной работы, а также итоговая защита проекта, зачастую у обучающихся возникает проблема выбора темы итогового проекта.

Поэтому необходимо помочь обучающимся найти все пути, ведущие к достижению цели, выделить общепринятые, общеизвестные и нестандартные, альтернативные; сделать выбор, оценив эффективность каждого способа.

Правило 1. Тема должна быть интересна ребенку, должна увлекать его. Исследовательская работа эффективна только на добровольной основе. Тема, навязанная ученику, какой бы важной она ни казалась взрослым, не даст должного эффекта.

Правило 2. Тема должна быть выполнима, решение ее должно быть полезно участникам исследования. Натолкнуть ребенка на ту идею, в которой он максимально реализуется как исследователь, раскроет лучшие стороны своего интеллекта, получит новые полезные знания, умения и навыки, – сложная, но необходимая задача для педагога.

Правило 3. Тема должна быть оригинальной с элементами неожиданности, необычности. Оригинальность следует понимать, как способность нестандартно смотреть на традиционные предметы и явления.

Правило 4. Тема должна быть такой, чтобы работа могла быть выполнена относительно быстро. Способность долго концентрировать собственное внимание на одном объекте, т. е. долговременно, целеустремленно работать в одном направлении, у школьника ограничена.

Правило 5. Тема должна быть доступной. Она должна соответствовать возрастным особенностям детей. Это касается не только выбора темы исследования, но и формулировки и отбора материала для ее решения. Одна и та же проблема может решаться разными возрастными группами на различных этапах обучения.

Правило 6. Сочетание желаний и возможностей. Выбирая тему, педагог должен учесть наличие требуемых средств и материалов – исследовательской базы. Ее отсутствие, невозможность собрать необходимые данные обычно приводят к поверхностному решению, порождают "пустословие". Это мешает развитию критического мышления, основанного на доказательном исследовании и надежных знаниях.

Правило 7. С выбором темы не стоит затягивать. Большинство обучающихся не имеют постоянных пристрастий, их интересы ситуативны. Поэтому, выбирая тему, действовать следует быстро, пока интерес не угас.

Критерии оценки проектов

По каждому пункту оценивается уровень компетенций

Низкий уровень (1 балл)

Средний уровень (2-3 балла)

Высокий уровень (4 балла)

1. Оригинальность и качество решения – Проект уникален и продемонстрировал творческое мышление участников. Проект хорошо продуман и имеет сюжет / концепцию.

2. Зрелищность – Проект имел восторженные отзывы, смог заинтересовать на его дальнейшее изучение.

3. Сложность – Трудоемкость, многообразие используемых функций.

4. Понимание технической части – Команда продемонстрировала свою компетентность, сумела четко и ясно объяснить, как их проект работает.

5. Инженерные решения – В конструкции проекта использовались хорошие инженерные концепции.

6. Эстетичность – Проект имеет хороший внешний вид. Команда сделала все возможное, чтобы проект выглядел профессионально.

7. Навыки общения и аргументации – Участники смогли рассказать, о чем их проект, и объяснить, как он работает и ПОЧЕМУ они решили его сделать.

8. Скорость мышления – Участники команды с легкостью ответили на вопросы, касающиеся их проекта.

9. Уровень понимания проекта – Участники продемонстрировали, что все члены команды имеют одинаковый уровень знаний о проекте.

10. Сплоченность коллектива – Команда продемонстрировала, что все участники коллектива сыграли важную роль в создании и презентации проекта.

11. Командный дух – Все члены команды проявили энтузиазм и заинтересованность в презентации проекта другим.

8. Методические материалы

Основные задачи вводного уровня – привлечь детей к исследовательской и изобретательской деятельности, показать им, что направление интересно и перспективно. Задача педагога – развить у обучающихся навыки, которые им потребуются в проектной работе и в дальнейшем освоении программ.

Методика преподавания включает разнообразные формы, методы и приемы обучения и воспитания. Обоснованность применения различных методов обусловлена тем, что нет ни одного универсального метода для решения разнообразных творческих задач.

Особенности организации образовательного процесса

Работа по программе педагога с обучающимися проводится в очной (при необходимости в дистанционной форме). Также возможна реализация программы в условиях сетевого взаимодействия с образовательными организациями, при наличии материально-технического оснащения.

Методы обучения

Методы обучения, применяемые в реализации программы «Сетевое и системное администрирование», можно систематизировать на основе источника получения знания:

- словесные: рассказ, объяснение, беседа, дискуссия;
- наглядные: демонстрация дидактических материалов, видеофильмов; компьютерные игры.
- практические: работа с аудио- и видеоматериалами, тематические экскурсии, интернет-экскурсии, тренинги, участие в мероприятиях.

Вместе с традиционными методами на занятиях спешно используются активные методы обучения: мозговой штурм, моделирование, метод проектов, метод эвристических вопросов, игровые ситуации, анализ конкретных ситуаций (case-study) и др.

Выбор методов обучения зависит от дидактических целей, от характера содержания занятия, от уровня развития обучающихся.

Формы организации образовательного процесса

Занятия проводятся с использованием различных *форм организации учебной деятельности* (групповая, фронтальная, индивидуальная, индивидуальная дистанционная, групповая дистанционная).

Разнообразные формы обучения и типы занятий создают условия для развития познавательной активности, повышения интереса детей к обучению.

Формы организации учебного занятия

Основной формой проведения учебных занятий является практическое занятие. Однако в ходе реализации программы, педагог вправе применять любую из доступных форм организации учебного занятия: беседа, встреча с интересными людьми, выставка, диспут, защита проектов, игра, конкурс, конференция, круглый стол, лекция, мастер-класс, «мозговой штурм», наблюдение, олимпиада, презентация, семинар, соревнование, чемпионат, экскурсия.

Типы занятий: изучение новой информации, занятия по формированию новых умений, обобщение и систематизация изученного, практическое применение знаний, умений, комбинированные занятия, контрольно-проверочные занятия.

Педагогические технологии, используемые в образовательном процессе

1) Проектная технология, учебно-исследовательская деятельность. На протяжении всего курса обучения учащиеся вовлечены в учебно-исследовательскую деятельность, которая позволяет им находить, обрабатывать, сравнивать и систематизировать информацию, полученную из встреч с интересными людьми, публикаций в сети Интернет. В ходе образовательного процесса учащиеся создают и защищают собственные исследовательские работы, рефераты, учатся методам поиска информации, самопрезентации, которые необходимы им в дальнейшей жизни и профессиональной карьере, на практических занятиях учащиеся выполняют исследовательские проекты. Проектная деятельность позволяет учащимся принять активную гражданскую позицию, сформировать потребность в участии в общественно полезной деятельности, необходимость быть нужным обществу. На занятиях создаются и реализуются учебные мини-проекты, в которых учащиеся решают учебные задачи на основе построения последовательности этапов от цели к конкретному результату. В процессе обучения осуществляется знакомство учащихся с информационно-коммуникационными технологиями, достижениями науки техники в области инженерной мысли.

Современные педагогические технологии в сочетании с современными информационными технологиями могут существенно повысить эффективность образовательного процесса, решить стоящие перед педагогом задачи воспитания всесторонне развитой, творчески свободной личности.

2) Технология развития критического мышления помогает учащимся определять приоритеты, анализировать, оценивать, выявлять ошибки, повысить мотивацию. Осуществляется при совместной работе в группах, при взаимодействии во время

выполнения заданий, при диалоге обучающихся между собой и с педагогом. Обязательным условием является сбор данных о динамике обучающегося и анализ его достижений и трудностей.

Алгоритм формирования критического мышления, предполагающий ответы на следующие вопросы:

1. Какова цель данной познавательной деятельности?
2. Что известно?
3. Что делать?
4. Достигнута ли поставленная цель?

Таким образом, критическое мышление - значит «искусство суждения, основанное на критериях». Результат - владение стратегиями критического мышления.

3) Технология имитационной игры – это моделирование реальной деятельности в специально созданных условиях, а её элементы включают в себя взаимосвязанные знаниевые и деятельностные компоненты обучения.

Особенности:

- не моделируется труд конкретных работников;
- имитируются лишь некоторые хозяйственные, правовые, экономические, экологические, социально-психологические принципы, определяющие поведение людей и механизмы их действий (в экстремальных ситуациях);
- моделирование только среды, особенности среды знакомы играющим в основном понаслышке, что делает анализ информации более сложным и субъективным;
- общая цель всего игрового коллектива изначально не задана, и для ее достижения самими игроками может быть найден определенный механизм взаимодействия;
- отсутствуют альтернативы, участники должны действовать лишь в предложенных вариантах;
- не программируется конфликтная ситуация (как, например, в деловых играх), а представлены только различные личные (субъективные) интересы участников игры;
- описанные сценарии игр не включают технологии и механизмы специального обучения общению и коллективному принятию решений.

4) Технология проблемного обучения способствует развитию проблемного мышления учащихся и педагога.

Результаты:

- усвоение учащимися системы знаний и способов умственной деятельности;
- развитие интеллектуальных умений и навыков учащихся;

- усвоение способов организации познавательной деятельности и формирования познавательной самостоятельности;
- развитие интеллектуальных возможностей, включающих творческие способности и прошлый опыт учащихся.

Проблемный вопрос - это входящий в состав проблемной задачи или отдельно взятый учебный вопрос (вопрос-проблема), требующий ответа на него посредством мышления. Вопрос же, требующий воспроизведения по памяти, не является проблемным. Вопросы, стимулирующие мышление, начинаются с таких вопросительных слов и словосочетаний, как «почему», «отчего», «как (чем) это объяснить», «как это понимать», «как доказать (обосновать)», «что из этого следует (какой вывод)» и т.п. А вопросительные слова «кто», «что», «когда», «где», «сколько», «какой» всегда требуют ответа на основе памяти.

Проблемная задача – учебная проблема с четкими условиями, задаваемыми преподавателем (лектором) или выявленными и сформулированными кем-либо из обучаемых (студентов), и в силу этого получившую ограниченное поле поиска (в отличие от объективно возникающей перед человеком жизненной проблемы) и ставшую доступной для решения всеми обучаемыми (студентами).

Проблемная ситуация – это ситуация познавательного затруднения, вовлекающая учащихся в самостоятельное познание элементов новой темы.

5) Интерактивные технологии направлены на развитие готовности к организации группового общения. Результаты:

- готовность воспринимать многомерность информацию, работать в режиме диалога;
- способность выбирать и обосновывать выбор методов, форм и техник организации коммуникационного процесса;
- владение психологическими техниками и методами организации коммуникационного процесса.

б) Технология дискуссионного общения включает в себя взаимосвязанные компоненты:

- мотивационный (готовность, желание принять участие в дискуссии);
- познавательный (знание о предмете спора, проблемная ситуация);
- операционно-коммуникативный (умение вести спор, отстаивать свою точку зрения, владеть способами осуществления логических операций);
- эмоционально-оценочный (эмоциональные переживания, потребности,

отношения, мотивы, оценки, личностный смысл).

