Областная государственная автономная нетиповая образовательная организация «Центр выявления и поддержки одарённых детей в Ульяновской области «Алые паруса»

СОГЛАСОВАНО

Протокол заседания
экспертного совета
«ОГАН ОО Центр «Алые паруса»
от __28.04.2026_____ № _1__

СОГЛАСОВАНО

Протокол заседания
педагогического совета
«ОГАН ОО Центр «Алые паруса»
от__29.04.2026_____ № _1__

УТВЕРЖДАЮ

Директор «ОГАН ОО Центр «Алые паруса»
_____ Т.А. Хмелевская

Приказ №_102-ОП_ от 19.05.2026

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 3A7E2856D35681539B3C3373D62F9A1D
Владелец: Хмелевская Татьяна Александровна
Действителен: с 06.04.2026 до 30.06.2027

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Основы программирования на языке Python»

направленность: техническая
уровень программы: базовый

Срок реализации программы: **9 месяцев**
Возраст обучающихся: **11 – 16 лет**

Автор-разработчик:
педагог дополнительного образования
Вазюкова Анастасия Сергеевна

Ульяновск, 2026 г.

Структура дополнительной общеразвивающей программы

1 Комплекс основных характеристик программы 3
1.1.Пояснительная записка 3
1.2.Цели и задачи программы 5
1.3.Содержание программы 6
1.3.1.Учебный план 6
1.4 Планируемые результаты 9
2. Комплекс организационно-педагогических условий 11
2.1 Календарный учебный график 11
2.2 Условия реализации программы 16
2.3 Формы аттестации 18
2.4 Критерии оценки 18
2.5 Воспитательный компонент 19
Список литературы 21
Приложение 23

1 Комплекс основных характеристик программы
1.1 Пояснительная записка

Нормативно-правовое обеспечение программы

• Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» (ст. 2, ст. 15, ст.16, ст.17, ст.75, ст. 79).

• Приказ Минпросвещения РФ от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

• Методические рекомендации по проектированию дополнительных общеразвивающих программ (Письмо Минобрнауки России от 18.11.2015 г.

№ 09-3242).

• Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» (зарегистрировано в Минюсте России 18.12.2020 № 61573)

• Постановление Правительства Российской Федерации об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ от 11 октября 2023 г. n 1678.

• Распоряжение Правительства РФ от 31.03.2022 N 678-р «Об утверждении Концепции развития дополнительного образования детей и признании утратившим силу Распоряжения Правительства РФ от 04.09.2014 N 1726-р».

• Устав «ОГАН ОО Центр «Алые паруса».

• Локальные акты «ОГАН ОО Центр «Алые паруса».

Уровень программы: базовый.

Направленность программы: техническая.

Программа «Основы программирования на языке Python» направлена на формирование алгоритмического мышления, базовых навыков программирования на языке Python и практических компетенций в области разработки программных продуктов, а также на развитие познавательного интереса к IT-сфере и подготовку к осознанному профессиональному самоопределению в области информационных технологий

Дополнительность программы по отношению к программам общего образования. Программа направлена на значительное расширение школьного курса информатики.

Актуальность программы. В условиях глобальной информатизации всех сфер жизни возрастает потребность в специалистах с высоким уровнем владения информационными компетенциями. Python остаётся одним из самых востребованных языков программирования в мире, используется в веб-разработке, анализе данных, машинном обучении, автоматизации и других областях.

Python отличается лаконичным, интуитивно понятным синтаксисом,

близостью к естественному языку и богатой экосистемой библиотек, что делает его идеальным инструментом для первоначального обучения программированию. Его простота позволяет детям безболезненно осваивать сложные концепции программирования.

Программа соответствует приоритетным направлениям развития дополнительного образования, закреплённым в Концепции развития дополнительного образования детей до 2030 года, и отвечает социальному заказу общества на подготовку квалифицированных кадров в области программирования

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

Возникающие при этом познавательные переживания обусловливают сознательное отношение к изучению основных теоретических положений информатики.

Новизна программы состоит в том, в рамках предлагаемой программы изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др.

Отличительные особенности программы заключаются в том, что она является мощным образовательным инструментом, позволяющим дать обучающимся навыки по программированию на языке Python. Реализация программы основана на деятельностном подходе, большая часть времени отводится практической деятельности, способствующей развитию активной познавательной деятельности, творчества и достижению высоких результатов в области информационно-коммуникационных технологий.

Педагогическая целесообразность программы состоит в том, что в процессе её реализации обучающиеся овладевают прочными теоретическими знаниями и умениями применять эти знания для решения практических и экспериментальных задач. В процессе работы развивается логическое и творческое мышление, а также такие качества, как гибкость, конструктивность и критичность мышления.

Адресат программы: обучающиеся 11 – 16 лет. Выбор данной возрастной категории для освоения программы обуславливается психологическими особенностями детей среднего и старшего школьного возраста в восприятии материала, мотивации к учебной деятельности, коммуникативной и аналитической деятельности, формированию мировоззрения.

***Форма обучения*:** очная, с применением дистанционных образовательных технологий и электронного обучения.

Формы занятий: для очного обучения чаще всего применяются комбинированные и практические занятия.

Возможны следующие формы проведения занятий с применением дистанционных образовательных технологий и электронного обучения:

- Видеоконференция – обеспечивает двухстороннюю аудио- и видеосвязь между педагогом и обучающимися. Преимуществом такой формы виртуального общения является визуальный контакт в режиме реального времени. Охватывает большое количество участников образовательного процесса.

- Чат–занятия – это занятия, которые проводятся с использованием чатов – электронной системы общения, проводится синхронно, то есть все участники имеют доступ к чату в режиме онлайн.

- Онлайн–консультации – это наиболее эффективная форма взаимодействия между педагогом и обучающимися. Преимущество таких консультаций в том, что, как при аудио и тем более видео контакте, создается максимально приближённая к реальности атмосфера живого общения. К наиболее приемлемым для дополнительного образования можно отнести также такие формы, как мастер классы, дистанционные конкурсы, фестивали, выставки, электронные экскурсии.

Виды занятий: лекции и практические работы. Основной формой является групповое обучение.

Объём программы: 148 часов.

Срок освоения программы: 9 месяцев.

Режим занятий: 2 раза в неделю по 2 академических часа с перерывом 10 минут.

1.2 Цели и задачи программы

Цель программы:

Создать условия для формирования базовых знаний и практических навыков программирования на языке Python, развития алгоритмического и логического мышления, а также подготовка к осознанному выбору профессиональной деятельности в сфере информационных технологий

Задачи:

Образовательные:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- познакомить с основными конструкциями языка программирования Python, позволяющими работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

Развивающие:

• развить умение критически оценивать и интерпретировать информацию с разных позиций, распознавать и фиксировать противоречия в информационных источниках;

• развить умение находить и приводить критические аргументы в отношении действий и суждений другого; спокойно и разумно относиться к критическим замечаниям в отношении собственного суждения, рассматривать их как ресурс собственного развития;

• развить умение применять знания в нестандартных и проблемных ситуациях.

Воспитательные:

• воспитать самостоятельность, уверенность в своих силах;

• воспитать ценностное отношение к знаниям, интерес к изучаемому предмету;

• развить коммуникативные навыки;

• воспитать трудолюбие, стремление добиваться поставленной цели;

• при осуществлении групповой работы быть как руководителем, так и членом команды в разных ролях (генератор идей, критик, исполнитель, выступающий, эксперт).

1.3 Содержание программы

1.3.1 Учебный план

№ п\п	Тема занятия	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Входная диагностика		***12***	***4***	***8***	Тестирование, решение задач
Раздел 1 Введение в программирование		***53***	***14***	***39***	
1.1	Знакомство с языком Python	8	2	6	Устный опрос. Решение поставленных задач. Тестирование
1.2	Переменные и выражения	8	2	6	Устный опрос. Решение поставленных задач. Тестирование
1.3	Условный оператор	8	2	6	Решение поставленных задач. Зачетная работа. Тестирование
1.4	Знакомство с циклом while	7	2	5	Решение поставленных задач. Тестирование
1.5	Отладчик	6	2	4	Решение поставленных задач. Зачетная работа. Тестирование

1.6	Цикл for	8	2	6	Решение поставленных задач. Творческая работа. Тестирование
1.7	True, False, Break, continue	8	2	6	Решение поставленных задач. Тестирование
Раздел 2 Базовые конструкции Python		***44***	***26***	***18***	
2.1	Вложенные циклы	4	2	2	Решение поставленных задач. Тестирование
2.2	Множества	4	2	2	Решение поставленных задач. Тестирование
2.3	Строки. Индексация	4	2	2	Решение поставленных задач. Тестирование
2.4	Строки. Срезы	4	2	2	Решение поставленных задач. Тестирование
2.5	Знакомство со списками	6	4	2	Решение поставленных задач. Творческая работа. Тестирование
2.6	Кортежи. Преобразование коллекций	6	4	2	Решение поставленных задач. Творческая работа. Тестирование
2.7	Методы split и join. Списочные выражения	4	2	2	Решение поставленных задач. Тестирование
2.8	Словари и множества	12	8	4	Решение поставленных задач. Творческая работа. Тестирование
Раздел 3 Решение прикладных задач в Python		***39***	***12***	***27***	
3.1	Функции	7	2	5	Решение поставленных задач. Тестирование
3.2	Обзор стандартной библиотеки Python	7	1	6	Решение поставленных задач. Тестирование
3.3	Дополнительные функции Python	7	2	5	Решение поставленных задач. Тестирование
3.4	Модуль Turtle	5	1	4	Устный опрос. Решение поставленных задач
3.5	Введение в ООП	6	4	2	Устный опрос. Решение поставленных задач

3.6	Стиль программирования и отладка программ	7	2	5	Решение поставленных задач. Зачетная работа
Итого:		***148***	***56***	***92***	

Содержание учебного плана

Входная диагностика

Теория. Техника безопасности при работе в кабинете информатики. Правила поведения в коллективе. Знакомство с коллективом. Задача как предмет изучения в процессе обучения.

Практика. Тестирование на определение уровня способностей проводится на платформе Moodle под авторизованным доступом. Знакомство с библиотекой, электронными ресурсами. Решение задач. Оперирование вопросами при решении задач разного вида.

Форма контроля. Тестирование, решение задач.

Раздел 1 Введение в программирование

Теория. Знакомство с основными понятиями программирования: исполнитель, система команд, алгоритм, программа, среда разработки, интерпретатор, код программы и редактор кода.

Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

Практика. На этом этапе обучающиеся разрабатывают первые алгоритмы и программы, а также анализируют, на какие функциональные блоки может быть разбита программа, и определяют работоспособность разработанной программы.

В течение модуля ученики решают большое количество задач: от самых простых до сложных.

Форма контроля. Устный опрос. Решение поставленных задач. Тестирование. Творческая и зачетная работа.

Раздел 2 Базовые конструкции языка Python

Теория. Понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python.

Устройство циклов for. Основные управляющие конструкции циклического алгоритма в Python. Простейшие циклы и циклы с переменными.

Практика. Работа со списками, строками, множествами и кортежами в Python. Понятие итератора. Понятие подпрограммы, процедуры, функции. Функции в Python. Решение задач.

Форма контроля. Решение поставленных задач. Тестирование. Творческая работа.

Раздел 3 Решение прикладных задач в Python

Теория. Понятие ассоциативного массива. Словари в Python. Решение задач.

Модули в Python. Подключение и использование модулей стандартной библиотеки. Модульный принцип компоновки программы. Работа с

документацией в стандартной библиотеке. Понятие репозитория различных пакетов Python. Работа с внешними библиотеками Python и утилитой pip. Основы ООП. Решение задач.

Практика. Самостоятельная работа: работа с конспектом, который описывает способы решения задач, разработка проекта по индивидуальному заданию, составление отчёта о выполнении индивидуальной или совместной работе.

Тематика самостоятельных работ:

- задачи, в которых по заданному алгоритму необходимо написать программу;
- задачи, в которых необходимо составить алгоритм решения и написать по нему программу;
- задачи, для решения которых необходимо найти некоторую информацию в Интернете.

Выполненные самостоятельные работы загружаются в автоматизированную систему проверки задач Яндекс. Контест для дальнейшей оценки.

Зачётное занятие: выполнение итогового индивидуального задания по предложенной теме.

Форма контроля. Устный опрос. Решение поставленных задач. Тестирование. Зачетная работа.

1.4 Планируемые результаты

Личностные результаты:

- умение ясно, точно, грамотно излагать свои мысли в устной и письменной речи, понимать смысл поставленной задачи, выстраивать аргументацию, приводить примеры и контрпримеры;
- критичность мышления, умение распознавать логически некорректные высказывания, отличать гипотезу от факта;
- самостоятельность в приобретении новых знаний и практических умений;
- мотивация образовательной деятельности обучающихся как основы саморазвития и совершенствования личности на основе личностно-ориентированного подхода;
- креативность мышления, активность при решении логических задач.

Метапредметные результаты:

- умение самостоятельно ставить и формулировать для себя новые задачи в изучении программирования;
- умение осуществлять самостоятельный поиск информации с использованием различных источников (учебных текстов, справочных и научно-популярных изданий, компьютерных баз данных, ресурсов Интернета), ее обработку и представление в разных формах;
- развитие познавательных интересов в изучении;
- умение самостоятельно планировать пути достижения целей, в том числе альтернативных;

- умение осознанно выбирать наиболее эффективные способы решения поставленных задач;
- умение осуществлять самоконтроль за своей деятельностью в процессе достижения результатов;
- умение оценивать правильность выполнения учебной задачи.

Предметные результаты:

- умение составлять алгоритмы для решения задач;
- умение реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владение основными навыками программирования на языке Python;
- умение отлаживать и тестировать программы, написанные на языке Python;
- создание выпускного проекта, который ученики смогут использовать для своей проектной деятельности в школах.

2 Комплекс организационно-педагогических условий
2.1 Календарный учебный график

Количество учебных недель: **34**
Количество учебных дней: **68**
Даты начала и окончания учебного периода: **14.09.26-31.05.27**

№ п/п	Тема занятия	Кол-во часов	Форма занятия	Форма контроля	Дата проведения занятия		Причина изменения даты
					планируемая	фактическая	
Входная диагностика – 12 часов							
1.	Входная диагностика	2	Комбинированное занятие	Тестирование			
2.	Входная диагностика	2	Комбинированное занятие	Тестирование			
3.	Входная диагностика	2	Комбинированное занятие	Решение задач			
4.	Входная диагностика	2	Комбинированное занятие	Решение задач			
5.	Входная диагностика	2	Комбинированное занятие	Решение задач			
6.	Входная диагностика	2	Комбинированное занятие	Решение задач			
Раздел 1 Введение в программирование – 53 часа							
7.	Знакомство с языком Python	2	Комбинированное занятие	Устный опрос. Решение поставленных задач. Тестирование			
8.	Знакомство с языком Python	2	Комбинированное занятие	Устный опрос. Решение поставленных задач. Тестирование			
9.	Знакомство с языком Python	2	Комбинированное занятие	Устный опрос. Решение поставленных задач. Тестирование			
10.	Знакомство с языком Python	2	Комбинированное занятие	Устный опрос. Решение поставленных задач. Тестирование			
11.	Переменные и выражения	2	Комбинированное занятие	Устный опрос. Решение поставленных задач. Тестирование			
12.	Переменные и выражения	2	Комбинированное занятие	Устный опрос. Решение поставленных задач. Тестирование			
13.	Переменные и выражения	2	Комбинированное занятие	Устный опрос. Решение поставленных задач. Тестирование			
14.	Переменные и выражения	2	Комбинированное занятие	Устный опрос. Решение поставленных задач. Тестирование			

15.	Условный оператор	2	Комбинированное занятие	Решение поставленных задач. Зачетная работа. Тестирование			
16.	Условный оператор	2	Комбинированное занятие	Решение поставленных задач. Зачетная работа. Тестирование			
17.	Условный оператор	2	Комбинированное занятие	Решение поставленных задач. Зачетная работа. Тестирование			
18.	Условный оператор	2	Комбинированное занятие	Решение поставленных задач. Зачетная работа. Тестирование			
19.	Знакомство с циклом while	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
20.	Знакомство с циклом while	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
21.	Знакомство с циклом while	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
22.	Знакомство с циклом while	1	Комбинированное занятие	Решение поставленных задач. Тестирование			
23.	Отладчик	2	Комбинированное занятие	Решение поставленных задач. Зачетная работа. Тестирование			
24.	Отладчик	2	Комбинированное занятие	Решение поставленных задач. Зачетная работа. Тестирование			
25.	Отладчик	2	Комбинированное занятие	Решение поставленных задач. Зачетная работа. Тестирование			
26.	Цикл for	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
27.	Цикл for	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
28.	Цикл for	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
29.	Цикл for	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
30.	True, False, Break, continue	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
31.	True, False, Break, continue	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
32.	True, False, Break, continue	2	Комбинированное занятие	Решение поставленных задач. Тестирование			

33.	True, False, Break, continue	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
Раздел 2 Базовые конструкции Python – 44 часа							
34.	Вложенные циклы	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
35.	Вложенные циклы	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
36.	Множества	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
37.	Множества	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
38.	Строки. Индексация	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
39.	Строки. Индексация	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
40.	Строки. Срезы	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
41.	Строки. Срезы	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
42.	Знакомство со списками	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
43.	Знакомство со списками	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
44.	Знакомство со списками	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
45.	Кортежи. Преобразование коллекций	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
46.	Кортежи. Преобразование коллекций	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
47.	Кортежи. Преобразование коллекций	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
48.	Методы split и join. Списочные выражения	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
49.	Методы split и join. Списочные выражения	2	Комбинированное занятие	Решение поставленных задач. Тестирование			

50.	Словари и множества	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
51.	Словари и множества	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
52.	Словари и множества	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
53.	Словари и множества	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
54.	Словари и множества	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
55.	Словари и множества	2	Комбинированное занятие	Решение поставленных задач. Творческая работа. Тестирование			
Раздел 3 Решение прикладных задач в Python – 39 часов							
56.	Функции	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
57.	Функции	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
58.	Функции	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
59.	Функции	1	Комбинированное занятие	Решение поставленных задач. Тестирование			
60.	Обзор стандартной библиотеки Python	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
61.	Обзор стандартной библиотеки Python	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
62.	Обзор стандартной библиотеки Python	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
63.	Обзор стандартной библиотеки Python	1	Комбинированное занятие	Решение поставленных задач. Тестирование			
64.	Дополнительные функции Python	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
65.	Дополнительные функции Python	2	Комбинированное занятие	Решение поставленных задач. Тестирование			
66.	Дополнительные функции Python	2	Комбинированное занятие	Решение поставленных задач. Тестирование			

67.	Дополнительные функции Python	1	Комбинированное занятие	Решение поставленных задач. Тестирование			
68.	Модуль Turtle	2	Комбинированное занятие	Устный опрос. Решение поставленных задач			
69.	Модуль Turtle	2	Комбинированное занятие	Устный опрос. Решение поставленных задач			
70.	Модуль Turtle	1	Комбинированное занятие	Устный опрос. Решение поставленных задач			
71.	Введение в ООП	2	Комбинированное занятие	Устный опрос. Решение поставленных задач			
72.	Введение в ООП	2	Комбинированное занятие	Устный опрос. Решение поставленных задач			
73.	Введение в ООП	2	Комбинированное занятие	Устный опрос. Решение поставленных задач			
74.	Стиль программирования и отладка программ	2	Комбинированное занятие	Устный опрос. Решение поставленных задач			
75.	Стиль программирования и отладка программ	2	Комбинированное занятие	Устный опрос. Решение поставленных задач			
76.	Стиль программирования и отладка программ	2	Комбинированное занятие	Устный опрос. Решение поставленных задач			
77.	Стиль программирования и отладка программ	1	Комбинированное занятие	Устный опрос. Решение поставленных задач			
Итого: 148 часов							

2.2 Условия реализации программы

Материально-техническое обеспечение

Кабинет обеспечен соответствующей мебелью: рабочими столами, стульями, компьютерами, программным обеспечением, выходом в интернет, мультимедийной доской, столом для руководителя.

Группа обучающихся состоит не более чем из 15 человек.

Рабочее место оснащено столом, стульями, персональным компьютером или ноутбуком, компьютерной мышью, программным обеспечением.

Методическое обеспечение

Методическое обеспечение программы «Основы программирования на языке Python» представляет собой комплекс организационно-педагогических условий, методов, форм и приёмов, обеспечивающих эффективную реализацию образовательного процесса и достижение планируемых результатов.

Дидактический материал: таблицы, схемы, мультимедийные материалы.

Методы обучения:

– Словесные (лекция, объяснение, беседа, инструктаж); – наглядные (демонстрация программ, презентации, схемы алгоритмов); – практические (написание программ, отладка, решение задач).

– Объяснительно-иллюстративный (теоретический материал); – репродуктивный (отработка навыков по образцу); – проблемный (постановка задачи, самостоятельный поиск решения); – частично-поисковый (разработка алгоритма по заданию); – исследовательский (самостоятельный поиск информации в интернете, разработка проектов).

– Метод пошаговой отладки (трассировка); – метод «от простого к сложному» (постепенное наращивание функциональности); – метод «разбиения на подзадачи» (декомпозиция); – метод модульного тестирования (проверка работоспособности отдельных частей программы).

Технологии обучения:

Технология системно-задачного подхода – обучение через решение задач, начиная с самых простых и постепенно переходя к сложным.

Технология проектного обучения – разработка индивидуального проекта по предложенной теме с составлением отчёта.

Технология автоматизированной проверки – использование системы Яндекс.Контест для автоматической проверки решений задач.

Технология развития критического мышления – анализ алгоритмов, поиск ошибок, сравнение разных способов решения одной задачи.

Технология уровневой дифференциации – задания разного уровня сложности, позволяющие учитывать индивидуальные особенности учащихся.

Информационно-коммуникационные технологии (ИКТ) – использование интерактивных учебников, онлайн-платформ, системы дистанционного обучения.

К работе обучающиеся приступают после проведения соответствующего инструктажа по правилам техники безопасной работы, объявления темы занятия, плана работы. Новую тему педагог объясняет с применением технологий мультимедиа.

Методические материалы

1. Учебно-методические разработки педагога

– Рабочая программа – календарно-тематический план с указанием тем, количества часов, форм контроля.

– Конспекты теоретических занятий – по темам: введение в программирование, базовые конструкции Python, работа со списками, строками, множествами, словарями, модули и функции, основы ООП.

– Методические рекомендации по проведению занятий – по организации работы с разновозрастными группами, использованию автоматизированной системы проверки.

– Фонд оценочных средств – тесты, практические задания, критерии оценки творческих и зачётных работ.

– Разработки мастер-классов – по темам: «Угадай число», «Turtle-рисовалка», «Калькулятор настроения», «Викторина», «Мини-игра в Pygame».

– Методические разработки по отдельным темам – например, «Реализация ветвлений в языке программирования Python».

2. Дидактические материалы

– Раздаточные материалы для обучающихся:

• конспекты с описанием способов решения задач;

• памятка «Структура программы на Python»;

• памятка «Основные операторы Python»;

• словарь терминов программирования;

• шаблоны для составления отчётов.

Наглядно-иллюстративный материал:

• схемы алгоритмов;

• примеры программ с пояснениями;

• таблицы операторов и их синтаксиса;

• блок-схемы основных алгоритмических конструкций.

Банк задач:

• задачи на отработку базовых конструкций (ввод-вывод, if, while, for);

• задачи на работу со списками, строками, множествами, кортежами;

• задачи на разработку алгоритмов и написание программ;

• задачи на поиск информации в интернете и разработку проектов.

Чек-листы самопроверки:

• «Как проверить программу перед сдачей»;

• «Как составить алгоритм решения задачи»;

• «Как оформить отчёт о выполнении проекта».

Информационное обеспечение

• Персональный компьютер с пакетом Microsoft Office (Microsoft Word, Microsoft Power Point) и выходом в сеть Интернет.

• Домашняя страница Python www.python.org. Справочные материалы, официальная документация.

• Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», http://www.intuit.ru/studies/

courses/12179/1172/info.

- Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» http://www.intuit.ru/studies/courses/49/49/info.
- Сайт проекта Ореп BooK Project openbookproject.net содержит серию практических примеров на Python Криса Мейерса.
- Python. Подробный справочник Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
- Python. Справочник Марка Лутца. Справочник по наиболее часто использующимся функциям и модулям.
- «Грокаем Алгоритмы» Адитья Бхаргава. Иллюстрированное пособие для программистов.

Кадровое обеспечение

В реализации программы занят один педагог дополнительного образования Вазюкова Анастасия Сергеевна.

2.3 Формы аттестации

Формы проведения аттестации: опрос, решение поставленных задач, тестирование, зачетная работа, творческая работа.

2.4 Критерии оценки

Формы подведения итогов:

- итоговый контроль;
- соревнования различного уровня.

Способы определения результативности.

В образовательном процессе для диагностики успешности освоения учебной программы используются:

- метод наблюдения;
- метод анализа продуктов образовательной деятельности обучающегося.

Виды контроля:

- предварительный: анкетирование, опрос;
- текущий: тестирование, дискуссии, решение поставленных задач;
- итоговый: конкурсы внутри объединения, зачетная работа.

Формы диагностики:

1. Промежуточная диагностика проводится по завершении каждого модуля в форме тестирования. Примерные тестовые вопросы по программе представлены в Приложении.

2. Итоговая диагностика, проводится после завершения всей учебной программы. Предметом оценки служат умения и знания в предметной области «Информатика».

Оперативный контроль учебных достижений осуществляется на протяжении всех занятий и имеет своей целью оценку систематичности учебной работы обучающихся по формированию знаний и умений в рамках освоения

данного материала. Проводится в процессе устного опроса, проведения практических работ, выполнения индивидуальных заданий и т.п.

Задачи текущего контроля:

- повышение мотивации обучающихся к регулярной учебной работе;
- развитие навыков самостоятельной работы;
- обеспечение обратной связи между обучающимися и преподавателем, позволяющей определить уровень освоения учебного материала;
- дифференциация итоговой оценки знаний.

Таблица 1 – Шкала оценки знаний

Оценка	Значение	Средняя оценка	Баллы
A+	Замечательно	5	100
A	Отлично	4,75	95 – 99
A–	Почти отлично	4,5	90 – 94
B+	Очень хорошо	4,25	85 – 89
B	Хорошо	4	80 – 84
B-	Почти хорошо	3,75	75 – 79
C+	Вполне удовлетворительно	3,5	70 – 74
C	Удовлетворительно	3,25	65 – 69
C–	Почти удовлетворительно	3	60 – 64
F	Неудовлетворительно	0	59 – 0

Таблица 2 – Рейтинг – шкала

Форма контроля	Баллы
Текущий	10 – 20
Промежуточный	15 – 30
Домашний	5 – 10
Итоговый контроль	30 – 40
Всего:	100

2.5 Воспитательный компонент

Цель воспитательной работы:

Формирование интеллектуально развитой, ответственной и социально адаптированной личности через освоение основ алгоритмизации и программирования на языке Python, развитие настойчивости, самодисциплины и способности к системному мышлению.

Задачи воспитательной работы:

1. Развитие интереса к программированию:

• Создание увлекательной образовательной среды, способствующей мотивации к изучению программирования.

• Проведение мероприятий и конкурсов, направленных на популяризацию программирования.

2. Формирование навыков сотрудничества:

• Организация групповых проектов, где обучающиеся могут работать в

команде, делиться идеями и опытом.

• Проведение совместных обсуждений и мозговых штурмов для решения задач.

3. Развитие критического мышления:

• Обучение анализу и синтезу информации, необходимой для решения программных задач.

• Формирование навыков самоконтроля и саморефлексии через анализ выполненных заданий.

4. Создание условий для творческого самовыражения:

• Стимулирование творческого подхода к разработке программных решений.

• Поддержка инициатив обучающихся в создании собственных проектов.

5. Обучение основам этики и ответственности в цифровом мире:

• Ознакомление с правилами безопасного поведения в интернете.

• Формирование ответственности за свои действия в цифровом пространстве.

Приоритетные направления воспитательной деятельности:

Формирование интеллектуально развитой, ответственной и самостоятельной личности через освоение алгоритмического мышления и практического программирования, подготовка к осознанному профессиональному самоопределению в сфере информационных технологий.

Формы воспитательной работы:

- Беседа, лекция, дискуссия.

Методы воспитательной работы:

- Создание и реализация совместных проектов.

- Использование игровых форм обучения для повышения мотивации.

- Проведение обсуждений на темы этики технологий и их влияния на общество.

- Организация практических занятий с использованием современного оборудования.

Планируемые результаты воспитательной работы:

1. Когнитивные результаты:

• Владение базовыми концепциями программирования на языке Python, включая синтаксис, структуры данных и алгоритмы.

• Навыки логического мышления и аналитического подхода к решению задач.

2. Социальные результаты:

• Умение эффективно работать в команде, развивая навыки коммуникации и сотрудничества.

• Повышение уровня взаимопонимания и уважения среди участников программы.

3. Личностные результаты:

• Уверенность в своих силах и способности решать сложные задачи.

• Ответственность за результаты своей деятельности и уважение к труду других.

4. Творческие результаты:

• Создание собственного проекта, демонстрирующее творческий подход к программированию.

• Повышение уровня креативности через реализацию нестандартных идей и решений.

5. Этические результаты:

• Понимание основ цифровой этики и безопасности, умение защищать свои данные и уважать чужие права в сети.

• Понимание важности ответственности за свои действия в цифровом пространстве.

Календарный план воспитательной работы

№ п/п	Тема занятия	Воспитательный компонент
1.	Раздел 1 Введение в программирование	Цель: формирование интеллектуально развитой, ответственной и творческой личности через освоение основ алгоритмизации и программирования на языке Python. Задачи: 1. Воспитывать настойчивость и терпение при решении задач, умение преодолевать трудности при возникновении ошибок, не сдаваться, искать разные способы решения, доводить программу до рабочего состояния. 2. Формировать самостоятельность и ответственность, способность самостоятельно находить информацию, работать с документацией, планировать этапы разработки, соблюдать сроки выполнения заданий. 3. Развивать логическое и критическое мышление, умение анализировать задачи, разбивать их на подзадачи, составлять алгоритмы, проверять правильность решений, оценивать эффективность программ. 4. Формировать коммуникативные навыки, умение обсуждать решения, объяснять свой код, помогать товарищам, работать в команде, конструктивно воспринимать критику. 5. Развивать творческую инициативу и интерес к IT-сфере – поощрение поиска нестандартных решений, участие в разработке собственных проектов, интерес к современным технологиям и профессиональной деятельности в области программирования.
2.	Раздел 2 Базовые конструкции языка Python	
3.	Раздел 3 Решение прикладных задач в Python	

Список литературы

Литература для педагогов

1. Вандер Плас, Дж. Python для сложных задач. Наука о данных и машинное обучение / Дж. Вандер Плас. – Санкт-Петербург: Питер, 2018. – 576 с.

2. Васильев, А. Python на примерах. Практический курс / А. Васильев. – М.: Наука и техника, 2018. – 432 с.

3. Доусон, М. Программируем на Python / М. Доусон. – Санкт-Петербург: Прогресс книга, 2019. – 416 с.

4. Златопольский, Д.М. Основы программирования на языке Python / Д.М. Златопольский. – М.: ДМК Пресс, 2018. – 284 с.

5. Лутц, М. Изучаем Python. В 2-х т / М. Лутц. – М.: Вильямс, 2019. – 720 с.

6. Любанович, Б. Простой Python. Современный стиль программирования / Б. Любанович. – Санкт-Петербург: Питер, 2019. – 592 с.

7. Окулов, С.М. Основы программирования / С.М. Окулов. – М.: Бином. Лаборатория знаний, 2012. – 339 с.

8. Пратик, Дж. Искусственный интеллект с примерами на Python. Создание приложений искусственного интеллекта с помощью Python для взаимодействия с окружающим миром / Дж. Пратик. – М.: Вильямс, 2019. – 448 с.

9. Рамальо, Л. Python к вершинам мастерства / Л. Рамальо. – М.: ДМК Пресс, 2016. – 768 с.

10. Саммерфильд, М. Python на практике / М. Саммерфильд, пер. А.А. Слинкин – М.: ДМК-Пресс, 2014. – 338 с.

Литература для обучающихся

1. Бриггс, Дж. Python для детей. Самоучитель по программированию / Дж. Бриггс. – М.: Манн, 2018. – 320 с.

2. Бэрри, П. Изучаем программирование на Python / П. Бэрри. – М.: Эксмо, 2017. – 624 с.

3. Вордэрман, К. Программирование на Python. Иллюстрированное руководство для детей / К. Вордэрман. – М.: Манн, Иванов и Фербер, 2018. – 224 с.

4. Мэтиз, Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения / Э. Мэтиз. – Санкт-Петербург: Питер, 2020. – 496 с.

5. Свейгарт, Э. Учим Python, делая крутые игры / Э. Свейгарт. – М.: Бомбора, 2018. – 416 с.

6. Томашевский, П. Привет, Python. Моя первая книга по программированию / П. Томашевский. – М.: Наука и техника, 2020. – 256 с.

Литература для родителей

1. Жуков, Р. Язык программирования Python: практикум. Учебное пособие / Р. Жуков. – М.: Инфра-М, 2020. – 216 с.

2. МакГрат, М. Программирование на Python для начинающих / М. МакГрат. – М.: Эксмо, 2015. – 192 с.

3. Пэйн, Б. Python для детей и родителей. Играй и программируй / Б. Пэйн. – М.: Эксмо, 2018. – 352 с.

4. Солем, Я.Э. Программирование компьютерного зрения на языке Python / Я.Э. Солем. – М.: ДМК Пресс, 2016. – 312 с.

ПРИЛОЖЕНИЕ

Примерные тестовые вопросы по программе

Вопрос 1 Как получить данные от пользователя?

- Использовать метод read()
- Использовать метод get()
- Использовать метод cin()
- Использовать метод readLine()
- Использовать метод input()

Вопрос 2 Какая функция выводит что-либо в консоль (на экран монитора)?

- out()
- log()
- print()
- write()

Вопрос 3 Какие существуют типы переменных у чисел (выбрать несколько вариантов)?

- float
- list
- num
- int
- integer

Вопрос 4 Переменная int:

- вещественная переменная
- символьная строка
- логическая переменная
- целая переменная

Вопрос 5 Переменная float:

- целая переменная
- вещественная переменная
- логическая переменная

Вопрос 6 Имена переменных не могут включать:

- русские буквы
- латинские буквы
- пробелы
- скобки, знаки + = ! ? и др.

ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ

Вопрос 7 Какие имена являются правильными в PYTHON (выбрать несколько)?

- N
- sum
- 41And
- A+B

Вопрос 8 Что будет в результате выполнения программы?

```
main.py
1  a=5
2  b=a+5
3  a=b*100
4  print(a)
```

- 25
- 250
- 2500
- 1000

Вопрос 9 Что будет в результате выполнения следующего действия print (23 % 2)?

- 1
- 10
- 0

Вопрос 10 Результатом вычисления print (25 // 3) будет число:

- 8
- 10
- 8.3

Вопрос 11 Что будет в результате выполнения программы? Входные данные:

```
a = int(input())
b = int(input())
if a < b:
    print(a)
else:
    print(b)
```

a=10? b=20.

- 10
- 20
- 30
- -10

ОКОНЧАНИЕ ПРИЛОЖЕНИЯ

Вопрос 12 Что выведется на экран монитора в результате выполнения программы? Входные данные: a=10, b=20, c=30.

```
.py
1
2  a=int(input('a= '))
3  b=int(input('b= '))
4  c=int(input('c= '))
5  if a==b and b==c and c==a:
6      print('треугольник равносторонний')
7  else:
8      print('треугольник не равносторонний')
```

- равносторонний
- равнобедренный
- не равносторонний

Вопрос 13 Что будет выведено на экран монитора в результате выполнения программы?

```
main.py
1  z=5
2  z1=z**3
3  print(z1)
4
```

- 5
- 25
- 125

Вопрос 14 Тело цикла – это ...

- группа команд, не входящих в циклическую структуру
- произвольный текст
- произвольная группа команд
- группа команд, повторяющихся некоторое число раз

Вопрос 15 Если говорят, что программа "зациклилась", то это значит ...

- тело цикла не выполняется
- тело цикла выполняется бесконечно долго
- тело цикла выполняется только один раз