Министерство образования Нижегородской области

Государственное бюджетное профессиональное образовательное учреждение «Семеновский индустриально-художественный техникум»

Центр цифрового образования детей «IT-Куб»

СОГЛАСОВАНО:
Начальник ЦЦОД «IT-Куб»
_____ А.Н. Дубовицкий
«13» августа 2026 г.

УТВЕРЖДАЮ:
Директор ГБПОУ «СИХТ»
_____ И.Е. Напылов
Приказ № 274-од от «14» 08 2026 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«PYTHON - ПРОГРАММИРОВАНИЕ»

Направленность – техническая

Возраст обучающихся: 12-18 лет

Срок реализации: 1 год

Объем: 136 часов

Автор-составитель:
Новожилов Игорь Сергеевич

г. Семенов, 2026 г.

Содержание

1. Комплекс основных характеристик программы 3

1.1. Пояснительная записка 3

1.2. Цели и задачи курса 10

1.3. Содержание общеразвивающей программы 12

1.3.1. Учебный план 12

1.3.2. Содержание учебного плана 16

1.4. Планируемые результаты 25

2. Комплекс организационно-педагогических условий реализации общеразвивающей программы 26

2.1. Календарный учебный график на 2026-2027 учебный год 26

2.2 Условия реализации программы 27

2.3 Формы аттестации и оценочные материалы 28

2.4 Методическое обеспечение 33

Список литературы 35

1. Комплекс основных характеристик программы

1.1. Пояснительная записка

Программирование занимает одну из значительных ниш в современном мире. Нынешнее программирование многогранно и используется в таких важных сферах как строительство, бизнес и экономика, медицина, биология и физика. Большой процент физического труда в промышленности заменен на машинный и роботизированный труд, который управляется посредством программного обеспечения, что обеспечивает существенный прирост скорости, точности операций и эффективности производства. Такое богатство разнообразия применений обеспечивается солидным выбором языков программирования, у каждого из которых есть свои плюсы и минусы.

Для реализации программы выбран язык Python – это один из самых популярных, быстро развивающихся языков в современном мире, при этом он имеет достаточно простой синтаксис и подходит для начального обучения.

Актуальность программы отвечает современным потребностям обучающихся и родителей в получении необходимых теоретических знаний и навыков в сфере программирования для личностного развития и успешной адаптации к требованиям жизни в современном информационном обществе. Работая индивидуально, парами или в командах, обучающиеся могут решать различные задачи. Изучение программирования развивает логическое мышление, креативность, исследовательские и творческие качества. Писать программы – занятие увлекательное и временами непростое, однако полученный опыт пригодится и в школе, и дома, да и на рынке труда профессия программиста востребована.

Дополнительная общеобразовательная программа «Python- программирование» имеет ***техническую направленность***, в ее основу заложены принципы модульности и практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на детальное изучение алгоритмизации, реализацию межпредметных связей, организацию проектной и исследовательской деятельности обучающихся.

Основанием для проектирования и реализации данной общеразвивающей программы служит перечень следующих нормативных правовых актов:

- ✓ Конституция Российской Федерации (принята всенародным голосованием 12.12.1993 с изменениями, одобренными в ходе общероссийского голосования 01.07.2020);
- ✓ Указ Президента Российской Федерации от 09.11.2022 №809 «Об утверждении Основ государственной политики по сохранению и укреплению традиционных российских духовно-нравственных ценностей»;
- ✓ Указ Президента Российской Федерации от 07.05.2024 №309 «О национальных целях развития Российской Федерации на период до 2030 года и на перспективу до 2036 года»;
- ✓ Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- ✓ Федеральный закон от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребенка в РФ»;
- ✓ Постановление Правительства РФ от 26 декабря 2017 г. N 1642 "Об утверждении государственной программы Российской Федерации "Развитие образования"
- ✓ Концепция развития дополнительного образования до 2030 года (утверждена распоряжением Правительства РФ от 31.03.2022 № 678-р); -
- ✓ Распоряжение Правительства Нижегородской области от 12.09.2022 № 1057-р "О мерах по реализации Концепции развития дополнительного образования детей до 2030 года в Нижегородской области"
- ✓ Приказ Министерства просвещения РФ от 27 июля 2022 г. N 629 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам"

- ✓ Постановление Правительства РФ от 11.10.2023 N 1678 «Об утверждении правил применения организациями, осуществления образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- ✓ Профессиональный стандарт «Педагог дополнительного образования детей и взрослых» (приказ Министерства труда и социальной защиты РФ от 22 сентября 2021 г. № 652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»);
- ✓ Методические рекомендации по созданию и функционированию центров цифрового образования «IT-куб» (утверждены письмом Министерства просвещения Российской Федерации от 30 ноября 2023 г. N АЗ- 1750/04);
- ✓ Письмо Минпросвещения России от 29.09.2023 N АБ-3935/06 "О методических рекомендациях" (вместе с "Методическими рекомендациями по формированию механизмов обновления содержания, методов и технологий обучения в системе дополнительного образования детей...";
- ✓ Письмо Министерства образования и науки РФ от 18.11.2015г. № 093242. «О направлении Методических рекомендаций по проектированию дополнительных общеразвивающих программ (включая разноуровневые)»;
- ✓ Санитарные правила СП 2.4.2.4283-26 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи".
- ✓ локальные акты образовательной организации.

Направленность техническая. Заключается в формировании базовых навыков программирования и понимании ключевых концепций языка Python. Программа ориентирована на начинающих разработчиков и предполагает освоение фундаментальных аспектов написания программного кода.

Основные направления программы включают:

- освоение базовой структуры языка Python: синтаксис, переменные, типы данных, условия и циклы.

- формирование навыков алгоритмизации и решение типовых задач программирования;

- понимание особенностей объектов и классов в Python, методов и атрибутов;

- овладение инструментами ввода-вывода данных, обработки ошибок и исключительных ситуаций;

- создание графических приложений с PySide6 и ботов ВКонтакте;

Программа также способствует развитию критического мышления, способности анализировать проблемы и находить оптимальные пути их решения посредством программирования. Курс нацелен на подготовку к дальнейшему изучению продвинутых разделов Python и смежных дисциплин.

Актуальность программы обусловлена современным этапом развития общества, характеризующимся ускоренными темпами освоения техники и технологий, потребностью общества в технически грамотных специалистах и полностью отвечает социальному заказу по подготовке квалифицированных кадров в области IT-технологий. Непрерывно требуются новые идеи для создания конкурентоспособной продукции, подготовки высококвалифицированных кадров. Творческие способности и профессиональное мастерство специалистов становится главной производительной силой общества, и в целях приумножения достижений во всех областях науки и техники, необходимо планомерное и заблаговременное развитие у молодежи творческих и технических способностей, а также повышение статуса информационного образования в обществе.

Python – это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание

вебсценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. Как считают многие, один из самых используемых языков программирования в мире. Реализация этого направления позволяет стимулировать интерес и любознательность, развивать способности к решению проблемных ситуаций, умение исследовать проблему, анализировать имеющиеся ресурсы, выдвигать идеи, планировать решения и реализовывать их. Кроме того, реализация данного направления помогает развитию коммуникативных навыков у обучающихся за счет активного взаимодействия обучающихся в ходе групповой образовательной и проектной деятельности.

В то же время, объективные процессы информатизации российского общества формируют социальный заказ в сфере образования, в общем, и в сфере дополнительного образования, в частности на увеличение внимания к информационной грамотности обучающихся. Поэтому, в структуру предлагаемой программы включены теоретический материал и практические задачи, направленные на формирование начальной компьютерной грамотности, и информационной культуры, начальных навыков использования компьютерной техники, и современных информационных технологий для решения учебных, и практических задач.

Прогностичность дополнительной общеобразовательной общеразвивающей программы «Python-программирование» заключается в том, что она отражает требования и актуальные тенденции не только сегодняшнего, но и завтрашнего дня, а также имеет междисциплинарный характер, что полностью отражает современные тенденции построения как дополнительных общеобразовательных общеразвивающих программ, так и образования в целом.

Дополнительная общеобразовательная общеразвивающая программа «Python-программирование» вовлекает обучающихся в осознанный процесс саморазвития. В процессе они получат дополнительные знания в области

математики, электроники и информатики, а также знания в области технического английского языка.

Отличительная особенность программы. Дополнительная общеобразовательная общеразвивающая программа «Python-программирование» сочетает освоение основ языка и объектно-ориентированного программирования с разработкой графических приложений и ботов ВКонтакте. Обучающиеся изучают оба прикладных направления, а формат итогового проекта выбирают по своим интересам.

На обучение по данной программе принимаются дети в возрасте 12-18 лет. Такое распределение осуществляется по причине возрастных особенностей обучающихся, а также уже имеющихся знаний и умений, полученных в общеобразовательных учреждениях.

Адресат общеразвивающей программы. Дополнительная общеобразовательная общеразвивающая программа «Python-программирование» предназначена для обучающихся в возрасте 12-18 лет, не имеющих ограниченных возможностей здоровья, проявляющих интерес к программированию. Группа формируется ***по возрасту***12-18 лет.

Объём общеразвивающей программы 136 академических часов.

Формы обучения очная, с применением дистанционных образовательных технологий (Закон №273-ФЗ, гл.2, ст.17, п.2.).

Срок освоения программы определяется содержанием программы и составляет 1 учебный год и составляет 136 академических часов.

Режим занятий длительность одного занятия составляет 2 академических часа (по 45 минут) с перерывом (переменой) в 10 минут для проветривания помещения и отдыха обучающихся. Периодичность занятий – 2 раза в неделю в соответствии с расписанием.

Педагогическая целесообразность программы «Python-программирование» заключается в том, что в современных условиях технологическое образование становится необходимостью, поскольку настоящий этап развития

общества характеризуется интенсивным внедрением во все сферы человеческой деятельности новых наукоемких технологий. Поэтому раннее привлечение обучающихся к техническому творчеству в процессе изучения языков программирования является актуальным и полностью отвечает интересам обучающихся этой возрастной группы, их способностям и возможностям.

Место проведения занятий 606650, Нижегородская область, г. Семенов, ул. 1-е Мая, д. 27, каб. № 303.

1.2. Цели и задачи курса

Целью программы являются создание условий для изучения методов программирования на языке Python, рассмотрение различных парадигм программирования, подготовка к использованию как языка программирования, так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

Задачи программы:

***Обучающие*:**

— формирование умений и навыков программирования;

— освоение программирования в компьютерной среде PyCharm и VSCode;

— формирование умения самостоятельно решать технические задачи;

— обучение основам программирования на языке Python;

— стимулирование мотивации обучающихся к получению знаний, помогать формировать творческую личность обучающегося.

***Развивающие*:**

— формирование у обучающихся интереса к программированию;

— развитие навыков применения программирования в решении конкретных задач;

— развитие коммуникативных компетенций: навыков сотрудничества в коллективе, малой группе (в паре), участия в беседе, обсуждении;

— формирование и развитие информационных компетенций: навыков работы с различными источниками информации, умения самостоятельно искать, извлекать и отбирать необходимую для решения учебных задач информацию.

Воспитательные:

— формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;

— воспитание упорства в достижении результата;

— способствование воспитанию личностных качеств: целеустремленности, настойчивости, самостоятельности, чувства коллективизма и взаимной поддержки, чувство такта;

— формирование коммуникативных качеств через коллективную творческую деятельность и реализацию коллективных проектов;

— формирование адекватной самооценки через проявление себя в выставках, конкурсах, мероприятиях, подведении итогов деятельности;

— формирование устойчивого интереса к выбранному виду деятельности;

— патриотическое воспитание через изучение истории страны и развития науки и техники во время занятий творчеством;

— формирование нравственных смыслов и духовных ориентиров;

— формирование у обучающихся гражданской ответственности и правового самосознания, основанного на уважении к закону, знании прав человека и умении найти пути решения жизненных проблем;

— воспитание целеустремлённости, настойчивости и трудолюбия путём достижения учебных целей и подготовки к конкурсам и мероприятиям.

— воспитание доброго, бережного и качественного отношения к технике, взаимного уважения со своими товарищами, любви к трудовой деятельности.

1.3. Содержание общеразвивающей программы

1.3.1. Учебный план

Таблица 1

№, п/п	Название раздела, темы	Описание тем	Кол-во часов		
			Всего	Теория	Практика
Модуль 1. Базовые концепции языка программирования Python			**28**	**10**	**18**
1-2	Знакомство и вводное занятие	Знакомство, командообразование, инструктаж по ТБ. Входная диагностика. Среда разработки и запуск первой программы.	2	1	1
3-4	Переменные, операторы и выражения	Переменные и основные типы данных. Операторы и выражения. Ввод и вывод данных, преобразование типов.	2	1	1
5-6	Практика: ввод данных и вычисления	Практическое занятие на ввод данных, вычисления и вывод результатов без условных операторов.	2	-	2
7-10	Условные операторы	Операторы if, elif, else. Операции сравнения и логические выражения. Решение задач с ветвлением.	4	2	2
11-12	Встроенные функции	Применение встроенных функций для решения простых задач.	2	1	1
13-18	Циклы	Циклы for и while, функция range(), операторы break и continue. Дополнительная практика на повторяющиеся действия.	6	2	4
19-24	Функции	Определение и вызов функций, параметры и возвращаемое значение. Разделение задачи на функции.	6	2	4
25-26	Модули и библиотеки	Импорт и применение стандартных модулей random и math.	2	1	1
27-28	Практическая работа по модулю 1	Проверка освоения базовых концепций языка Python.	2	-	2
Модуль 2. Структуры данных и работа со строками			**20**	**6**	**14**

29-32	Работа со строками	Индексы, срезы, основные методы строк и форматирование. Обработка текстовых данных.	4	2	2
33-36	Списки	Создание списков, доступ к элементам, добавление и удаление, перебор и сортировка.	4	1	3
37-38	Кортежи и множества	Кортежи и множества. Простые примеры использования.	2	1	1
39-42	Словари	Ключи и значения, добавление, изменение, поиск и перебор данных.	4	2	2
43-44	Практика: списки словарей	Поиск и отбор данных в списках словарей.	2	-	2
45-46	Комплексные задачи на строки и коллекции	Закрепление работы со строками, списками и словарями.	2	-	2
47-48	Практическая работа по модулю 2	Проверка освоения строк и структур данных.	2	-	2
Модуль 3. Объектно-ориентированное программирование			**16**	**6**	**10**
49-50	Введение в ООП	Классы, объекты, атрибуты и методы. Моделирование знакомых объектов.	2	1	1
51-52	Создание классов и объектов	Собственные классы, метод __init__ и состояние объекта.	2	1	1
53-56	Наследование	Родительские и дочерние классы, расширение поведения и переопределение методов.	4	2	2
57-60	Полиморфизм	Общий способ взаимодействия с объектами разных классов.	4	2	2
61-62	Практика: взаимодействие объектов	Разработка небольшой программы с взаимодействующими объектами.	2	-	2
63-64	Практическая работа по модулю 3	Проверка освоения классов, объектов, наследования и полиморфизма.	2	-	2
Промежуточная аттестация					
65-66	**Промежуточная аттестация**	**Комплексное практическое задание по пройденным модулям.**	**2**	**-**	**2**

Модуль 4. Файлы и обработка ошибок			**10**	**4**	**6**
67-68	Работа с текстовыми файлами	Чтение и запись текстовых файлов, конструкция with, кодировка UTF-8.	2	1	1
69-70	Сохранение данных в JSON	Сохранение и загрузка данных в JSON. Ознакомление с форматом CSV.	2	1	1
71-72	Обработка исключений	Конструкции try и except, проверка ввода и понятные сообщения об ошибках.	2	1	1
73-74	Файлы и каталоги	Пути к файлам, проверка существования, основные операции с каталогами.	2	1	1
75-76	Практическая работа по модулю 4	Программа с сохранением данных и обработкой ошибок.	2	-	2
Модуль 5. Работа с веб-API			**8**	**3**	**5**
77-80	Основы HTTP и API	Запрос и ответ, коды состояния, ограничение времени ожидания. Библиотека requests.	4	2	2
81-82	Обработка ответа API	Получение JSON, извлечение нужных полей, обработка неудачного запроса.	2	1	1
83-84	Практическая работа по модулю 5	Получение и вывод данных из веб-API.	2	-	2
Модуль 6. Графические интерфейсы с PySide6			**16**	**6**	**10**
85-88	Введение в PySide6	Структура графического приложения, создание окна и основные виджеты.	4	2	2
89-92	Размещение элементов и обработка событий	Компоновки, сигналы и слоты, обработка действий пользователя.	4	2	2
93-96	Разработка графического интерфейса	Кнопки, поля ввода, списки или таблицы, простые диалоги. Применение ООП.	4	2	2
97-98	Практика: графическое приложение	Создание небольшого графического приложения с применением ООП.	2	-	2
99-100	Практическая работа по модулю 6	Проверка освоения графических интерфейсов с PySide6.	2	-	2
Модуль 7. Разработка ботов ВКонтакте			**16**	**5**	**11**

101-102	Создание и настройка бота ВКонтакте	Бот сообщества, настройка доступа, безопасное хранение токена, запуск и первый ответ.	2	1	1
103-106	Сообщения и команды	Получение сообщений, обработка команд и отправка ответов.	4	1	3
107-110	Кнопки и меню	Клавиатура бота, кнопки и организация меню.	4	2	2
111-112	Пошаговый диалог	Состояние пользователя, проверка ввода и отмена действия.	2	1	1
113-114	Практика: небольшой бот	Разработка небольшого бота с командами, кнопками и простым диалогом.	2	-	2
115-116	Практическая работа по модулю 7	Проверка освоения разработки ботов ВКонтакте.	2	-	2
Модуль 8. Итоговый проект			**18**	**-**	**18**
117-118	Выбор проекта и постановка задачи	Выбор графического приложения или бота ВК. Основные возможности и план работы.	2	-	2
119-120	Проектирование	Классы, данные, эскиз интерфейса или схема диалога.	2	-	2
121-122	Создание основы проекта	Запускаемая основа приложения или бота.	2	-	2
123-126	Реализация основной логики	Разработка основных возможностей проекта с применением ООП.	4	-	4
127-128	Работа с данными проекта	Хранение и обработка данных. Сохранение в файл при необходимости.	2	-	2
129-132	Проверка и исправление ошибок	Проверка ввода, тестирование основных сценариев и исправление ошибок.	4	-	4
133-134	Подготовка к защите	Инструкция по запуску, демонстрация и презентация проекта.	2	-	2
Итоговая аттестация					
135-136	**Итоговая аттестация**	**Защита и презентация проекта, ответы на вопросы комиссии.**	**2**	**-**	**2**
Итого:			**136**	**40**	**96**

Содержание учебного плана

Модуль 1. Базовые концепции языка программирования Python.

1.1. Тема: Знакомство и вводное занятие.

Теория: Правила работы в учебном кабинете и за компьютером. Назначение среды разработки, порядок создания и запуска программы.

Практика: Знакомство участников, командное задание и входной опрос. Открытие среды разработки и запуск первой программы по образцу.

1.2. Тема: Переменные, операторы и выражения.

Теория: Переменные, числа и строки. Арифметические операторы и порядок действий. Функции input(), print(), int(), float() и str().

Практика: Создание программ с вводом данных, вычислением выражений и выводом результатов. Преобразование введённого текста в числа.

1.3. Тема: Практика: ввод данных и вычисления.

Практика: Решение задач на расчёт стоимости покупки, перевод единиц и вычисление времени. Самостоятельная проверка результатов на нескольких наборах данных.

1.4. Тема: Условные операторы.

Теория: Условные операторы if, elif, else, операции сравнения, логические значения и операции and, or, not.

Практика: Разработка программ, выбирающих действие в зависимости от введённых данных. Проверка граничных значений и разбор ошибок в условиях.

1.5. Тема: Встроенные функции.

Теория: Назначение и применение функций abs(), round(), min(), max() и len(). Аргументы и результат вызова встроенной функции.

Практика: Решение задач с использованием встроенных функций. Сравнение результата программы с ожидаемым ответом.

1.6. Тема: Циклы.

Теория: Повторение действий с помощью for и while. Счётчик, накопление результата, условие завершения цикла. Применение range(), break и continue.

Практика: Решение задач на повторный ввод, подсчёт и накопление значений. Создание игры «Угадай число», поиск и исправление ошибок в циклах.

1.7. Тема: Функции.

Теория: Создание функций с помощью def, передача параметров и возврат результата через return. Отличие вывода результата от его возвращения.

Практика: Создание и проверка собственных функций. Переработка знакомых задач с выделением повторяющихся действий в функции.

1.8. Тема: Модули и библиотеки.

Теория: Назначение модулей и библиотек, основные способы импорта. Возможности стандартных модулей random и math.

Практика: Использование случайных чисел и математических функций в небольших программах.

1.9. Тема: Практическая работа по модулю 1.

Практика: Самостоятельное выполнение задания с вводом и выводом, условиями, циклами и функциями. Объяснение решения и проверка результатов.

Модуль 2. Структуры данных и работа со строками.

2.1. Тема: Работа со строками.

Теория: Доступ к символам, срезы, поиск, замена, разбиение и объединение строк. Форматирование результатов.

Практика: Решение задач на обработку текста: поиск слов, изменение строк, подготовка сообщений пользователю.

2.2. Тема: Списки.

Теория: Список как упорядоченная коллекция. Индексы, основные методы, перебор элементов и сортировка.

Практика: Создание списка дел или результатов викторины. Добавление, изменение и удаление элементов, поиск и отбор значений.

2.3. Тема: Кортежи и множества.

Теория: Кортеж как неизменяемая последовательность, множество как коллекция уникальных элементов. Основные операции с множествами.

Практика: Хранение фиксированных наборов значений в кортежах, удаление повторов и поиск общих элементов с помощью множеств.

2.4. Тема: Словари.

Теория: Структура словаря, доступ по ключу, добавление и изменение значений, проверка наличия ключа и перебор элементов.

Практика: Создание словаря для хранения сведений о вопросе викторины, книге или пользователе. Поиск и обновление информации.

2.5. Тема: Практика: списки словарей.

Практика: Обработка списка вопросов, книг или задач. Поиск записи по признаку, отбор подходящих записей и подсчёт результатов.

2.6. Тема: Комплексные задачи на строки и коллекции.

Практика: Решение комплексных задач с функциями и коллекциями. Проверка пустых коллекций, повторяющихся значений и отсутствующих данных.

2.7. Тема: Практическая работа по модулю 2.

Практика: Самостоятельное создание программы для обработки текстовых данных и коллекций. Объяснение выбора структуры данных.

Модуль 3. Объектно-ориентированное программирование.

3.1. Тема: Введение в ООП.

Теория: Назначение объектно-ориентированного программирования. Понятия класса, объекта, атрибута и метода.

Практика: Создание простых объектов и работа с их атрибутами и методами на примере вопроса викторины или игрового персонажа.

3.2. Тема: Создание классов и объектов.

Теория: Определение собственного класса. Метод __init__, параметры создания объекта, атрибуты экземпляра и изменение состояния через методы.

Практика: Создание нескольких экземпляров класса, изменение их состояния и вызов методов. Проверка независимости данных объектов.

3.3. Тема: Наследование.

Теория: Наследование как способ расширения класса. Родительский и дочерний классы, переопределение методов и обращение к родительскому классу.

Практика: Создание нескольких разновидностей игровых персонажей или вопросов викторины. Использование общих и собственных методов.

3.4. Тема: Полиморфизм.

Теория: Полиморфизм на примере одинакового вызова метода у объектов разных классов. Использование переопределённых методов.

Практика: Создание списка объектов разных классов и вызов общего метода для каждого объекта. Объяснение различий в поведении.

3.5. Тема: Практика: взаимодействие объектов.

Практика: Разработка модели викторины, каталога или игры с несколькими классами. Разделение обязанностей между объектами.

3.6. Тема: Практическая работа по модулю 3.

Практика: Самостоятельное выполнение задания с классами и объектами. Демонстрация наследования, переопределения методов и полиморфного вызова.

Модуль 4. Файлы и обработка ошибок.

4.1. Тема: Работа с текстовыми файлами.

Теория: Открытие, чтение и запись файлов. Режимы работы, кодировка UTF-8 и закрытие файла с помощью конструкции with.

Практика: Чтение исходных данных и сохранение результатов программы в текстовый файл.

4.2. Тема: Сохранение данных в JSON.

Теория: Структура JSON и связь со списками и словарями Python. Принципы сохранения и загрузки данных. Назначение CSV на простом примере.

Практика: Сохранение списка задач или результатов викторины в JSON и загрузка после повторного запуска программы.

4.3. Тема: Обработка исключений.

Теория: Причины ошибок ввода и работы с файлами. Обработка ожидаемых исключений с помощью try и except.

Практика: Обработка неправильного числового ввода, отсутствующего файла и некорректных данных. Вывод понятного сообщения пользователю.

4.4. Тема: Файлы и каталоги.

Теория: Абсолютные и относительные пути, рабочая папка. Основные средства модуля os для работы с путями и каталогами.

Практика: Проверка существования файла и создание папки для данных учебного проекта.

4.5. Тема: Практическая работа по модулю 4.

Практика: Создание программы, которая читает и сохраняет данные, проверяет ввод и обрабатывает основные ошибки работы с файлами.

4.6. Тема: Промежуточная аттестация.

Практика: Выполнение задания с использованием основных конструкций Python, коллекций, классов и файлов. Объяснение решения и проверка результатов.

Модуль 5. Работа с веб-API.

5.1. Тема: Основы HTTP и API.

Теория: Назначение HTTP и веб-API. Адрес запроса, ответ сервера, код состояния и ограничение времени ожидания. Основы использования requests.

Практика: Отправка простого запроса к учебному или публичному API. Просмотр ответа и проверка успешности запроса.

5.2. Тема: Обработка ответа API.

Теория: Преобразование JSON-ответа в структуры Python. Проверка наличия полей и обработка ошибок запроса.

Практика: Извлечение и вывод нужных данных. Обработка отсутствующего поля, неудачного запроса и превышения времени ожидания.

5.3. Тема: Практическая работа по модулю 5.

Практика: Создание программы, получающей данные из API и представляющей их пользователю. Проверка нормального ответа и ситуации сбоя.

Модуль 6. Графические интерфейсы с PySide6.

7.1. Тема: Введение в PySide6.

Теория: Назначение PySide6 и Qt. Структура приложения, главное окно и базовые виджеты.

Практика: Подготовка среды, запуск приложения и создание окна с надписями, кнопками и полями ввода.

7.2. Тема: Размещение элементов и обработка событий.

Теория: Размещение виджетов с помощью компоновок. Сигналы и слоты, связь действий пользователя с методами программы.

Практика: Создание интерфейса с компоновками, обработка нажатий кнопок и получение данных из полей ввода.

7.3. Тема: Разработка графического интерфейса.

Теория: Организация окна в виде класса. Связь интерфейса с логикой программы, отображение коллекций, назначение простых диалогов.

Практика: Разработка интерфейса викторины, калькулятора или списка дел. Проверка введённых значений и обновление отображаемых данных.

7.4. Тема: Практика: графическое приложение.

Практика: Объединение интерфейса и логики в работающее приложение. Проверка основного пользовательского сценария.

7.5. Тема: Практическая работа по модулю 6.

Практика: Самостоятельная разработка небольшого приложения с окном, элементами ввода и обработкой действий пользователя.

Модуль 7. Разработка ботов ВКонтакте.

8.1. Тема: Создание и настройка бота ВКонтакте.

Теория: Назначение бота сообщества и API ВКонтакте. Получение событий, ключ доступа и правила его безопасного хранения.

Практика: Настройка учебного сообщества под руководством педагога. Запуск подготовленного примера и получение первого ответа бота.

8.2. Тема: Сообщения и команды.

Теория: Событие нового сообщения, текст команды и ответ пользователю. Разделение обработки команд на функции и методы.

Практика: Создание приветствия, справки и нескольких команд. Обработка неизвестной команды и проверка ответов бота.

8.3. Тема: Кнопки и меню.

Теория: Назначение клавиатуры, кнопок и меню. Связь выбора пользователя с действием программы.

Практика: Создание меню и обработка нажатий кнопок. Реализация переходов между основными действиями и возврата в меню.

8.4. Тема: Пошаговый диалог.

Теория: Шаги простого диалога. Хранение состояния отдельно для каждого пользователя, проверка ответа и отмена действия.

Практика: Создание короткого опроса или викторины из нескольких шагов. Проверка неправильного ввода и возврата в меню.

8.5. Тема: Практика: небольшой бот.

Практика: Разработка бота-викторины, справочника или помощника. Проверка команд, меню и завершения диалога.

8.6. Тема: Практическая работа по модулю 7.

Практика: Самостоятельная реализация заданного сценария бота. Демонстрация команд, кнопок, пошагового диалога и обработки неверного ввода.

Модуль 8. Итоговый проект.

9.1. Тема: Выбор проекта и постановка задачи.

Практика: Выбор задачи и формата проекта: приложение на PySide6 или бот ВКонтакте. Описание пользователя, основного сценария и необходимых возможностей. Согласование плана с педагогом.

9.2. Тема: Проектирование.

Практика: Определение классов, их атрибутов и методов. Выбор структур данных. Подготовка эскиза интерфейса приложения или схемы диалога бота.

9.3. Тема: Создание основы проекта.

Практика: Создание файлов проекта и основных классов. Запуск окна приложения или обработчика сообщений бота. Проверка первого действия пользователя.

9.4. Тема: Реализация основной логики.

Практика: Реализация согласованных возможностей с использованием классов, объектов и методов. Объединение логики с интерфейсом или командами бота. Проверка работы каждой добавленной возможности.

9.5. Тема: Работа с данными проекта.

Практика: Организация данных в коллекциях и объектах. Чтение и запись текстового файла или JSON, если это требуется по задаче. Использование внешнего API является дополнительной возможностью проекта.

9.6. Тема: Проверка и исправление ошибок.

Практика: Составление перечня проверок и выполнение основного сценария. Проверка неправильного ввода, пустых данных и ожидаемых ошибок. Исправление обнаруженных проблем, повторная проверка программы.

9.7. Тема: Подготовка к защите.

Практика: Подготовка краткой инструкции по запуску и использованию программы. Создание презентации и демонстрационного примера. Объяснение назначения классов и методов, личного вклада и результатов работы.

9.8. Тема: Итоговая аттестация.

Практика: Демонстрация приложения или бота ВКонтакте. Объяснение кода и принятых решений, ответы на вопросы комиссии.

1.4. Планируемые результаты

Предметные результаты:

- сформированы умения создавать программы с вводом и выводом данных, условиями, циклами и функциями;
- освоены создание, запуск и проверка программ в используемой на занятиях среде разработки;
- сформированы умения обрабатывать строки, списки и словари, читать и сохранять данные в текстовых файлах и JSON;
- изучены классы, объекты, наследование и полиморфизм; сформированы умения применять ООП при решении учебных задач;
- освоены получение данных из веб-API, создание простого графического приложения и бота ВКонтакте; сформированы умения проверять ввод, исправлять ошибки и объяснять собственное решение.

Личностные результаты:

- развит интерес к программированию у обучающихся;
- развиты умения применять идеи программирования в решении конкретных задач;
- развиты коммуникативные компетенции: навыки сотрудничества в коллективе, малой группе (в паре), участия в беседе, обсуждении;
- сформированы и развиты информационные компетенции: навыки работы с различными источниками информации, умения самостоятельно искать, извлекать и отбирать необходимую для решения учебных задач информацию.

Метапредметные результаты:

- сформирована самостоятельность и творческий подход к решению задач с использованием средств вычислительной техники;
- воспитано упорство в достижении результата;
- воспитаны личностные качества: целеустремленность, настойчивость, самостоятельность, чувства коллективизма и взаимной поддержки, чувство такта.

2. Комплекс организационно-педагогических условий реализации общеразвивающей программы

2.1. Календарный учебный график на 2026-2027 учебный год

Таблица 2.

Год обучения	Сентябрь					октябрь				ноябрь					декабрь				январь				февраль					март				апрель				май				Всего учебных недель/ часов	Всего часов по программе
даты	01.09 – 06.09	07.09 – 13.09	14.09 - 20.09	21.09 – 27.09	28.09 – 04.10	05.10 – 11.10	12.10 – 18.10	19.10 – 25.10	26.10 – 01.11	02.11 – 08.11	09.11 – 15.11	16.11 – 22.11	23.11 – 29.11	30.11 – 06.12	07.12 – 13.12	14.12 – 20.12	21.12 – 27.12	28.12 – 03.01	04.01 – 10.01	11.01 – 17.01	18.01 – 24.01	25.01 – 31.01	01.02 – 07.02	08.02 – 14.02	15.02 – 21.02	22.02 – 28.02	01.03 – 07.03	08.03 – 14.03	15.03 – 21.03	22.03 – 28.03	29.03 – 04.04	05.04 – 11.04	12.04 – 18.04	19.04 – 25.04	26.04 – 02.05	03.05 – 09.05	10.05 – 16.05	17.05 – 23.05	24.05 – 30.05		
неделя	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39		
1ПП			2	4	4	4	4	4	4	2*	4	4	4	4	4	4	4	4		4	2 2	4	4	4	4	4	4	2*	4	4	4	4	4	4	4	4	2*	4	2 2	36/136	136

Условные обозначения:

	Занятия по расписанию		Каникулярный период
	Приемная кампания		Промежуточная и итоговая аттестация
*праздничный день			

2.2 Условия реализации программы

Материально-техническое обеспечение

Требования к помещению:

— помещение для занятий, отвечающее требованиям СанПин для учреждений дополнительного образования;

— качественное освещение;

— столы, стулья по количеству обучающихся и 1 рабочее место для педагога.

Оборудование:

— Ноутбук педагога MSI M5-17LS– 1 шт.

— Ноутбуки ASUS FA707 - 13 шт.

— МФУ – Pantum BM5100DW – 1 шт.

— Интерактивная доска – 1 шт.

— Тележка для зарядки и хранения ноутбуков – 1 шт.

— Флипчарт магнитно-маркерный на треноге – 1 шт.

— Доска магнитно-маркерная настенная - 1 шт.

— Наушники 13 шт.

Информационное обеспечение:

— Операционная система Windows

— Среда разработки PyCharm

— Среда разработки VSCode

— Браузер Yandex

— Текстовый редактор Notepad++

Кадровое обеспечение

Программа реализуется педагогом дополнительного образования – Новожиловым Игорем Сергеевичем, имеющим удостоверение повышения квалификации по данному направлению.

2.3 Формы аттестации и оценочные материалы

Оценочные материалы для дополнительной образовательной программы «Python - Программирование» стартового уровня предназначены для проверки уровня усвоенных знаний и сформированности начальных навыков программирования. Эти материалы позволяют объективно оценивать достижения обучающихся и обеспечивать своевременную коррекцию образовательных процессов.

Система отслеживания результатов обучающихся выстроена следующим образом:

- определение начального уровня знаний, умений и навыков;
- текущий контроль;
- промежуточная аттестация.
- итоговая аттестация.

Входная диагностика по программе «Python - программирование» проводится в начале освоения программы в форме опроса для выявления первоначальных знаний и навыков. Обучающимся предлагается объяснить, что такое программа, привести пример последовательности действий, описать знакомые способы работы с файлами и рассказать об опыте программирования. Ответы используются для выбора темпа и уровня поддержки на занятиях.

Текущий контроль осуществляется путем наблюдения, определения качества выполнения заданий, отслеживания динамики развития обучающегося. Способы проверки уровня освоения тем: опрос, выполнение упражнений, наблюдение. Формы проведения итогов по каждой теме и каждому разделу общеразвивающей программы соответствуют целям и задачам ДООП. Для всех модулей 0 баллов означает, что задание не выполнено либо представленный результат не позволяет оценить освоение темы.

Таблица. 3

№	Раздел (модуль)	Форма контроля	Критерии/успеваемость (0–3)
1	Базовые концепции Python	Практическая работа	1 — задача решена частично, есть ошибки в условиях, циклах или функциях; 2 — основная задача решена, остаются отдельные недочёты; 3 — решение корректно, ввод и вывод понятны, обучающийся объясняет код.
2	Структуры данных и строки	Практическая работа	1 — операции со строками и коллекциями выполнены частично; 2 — основные операции работают; 3 — задача решена корректно, проверены пустые коллекции, повторы и отсутствие нужных данных.
3	ООП	Практическая работа	1 — классы созданы, но поведение реализовано частично; 2 — классы и методы работают, наследование применено с отдельными недочётами; 3 — корректно применены классы, наследование и полиморфизм, обучающийся объясняет решение.
4	Файлы и обработка ошибок	Практическая работа	1 — чтение и запись работают с ошибками; 2 — данные сохраняются и загружаются; 3 — работа с файлами корректна, основные ошибки обработаны, сообщения понятны.
5	Работа с веб-API	Практическая работа	1 — получен ответ, но данные обработаны частично; 2 — нужные данные получены и выведены; 3 — предусмотрены ограничение времени ожидания, проверка ответа и понятное сообщение при сбое.
6	Графические интерфейсы с PySide6	Практическая работа	1 — окно создано, основные действия работают частично; 2 — интерфейс и основная логика работают; 3 — элементы размещены понятно, ввод проверяется, классы и методы применены осмысленно.
7	Боты ВКонтакте	Практическая работа	1 — отдельные команды работают, сценарий неполный; 2 — команды, меню и простой диалог работают; 3 — сценарий выполняется корректно, проверяется ввод, предусмотрены отмена и возврат в меню.

Промежуточная аттестация обучающихся осуществляется в форме выполнения практического задания и оценивается по 10-бальной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице:

Таблица 4.

Баллы, набранные обучающимся	Уровень освоения
0–4 баллов	низкий
5–7 баллов	средний
8–10 баллов	высокий

Критерии и баллы промежуточной аттестации:

Соответствие заданию — 2 (реализованы требуемые действия программы).

Самостоятельность — 2 (обучающийся объясняет решение и выполняет основные действия самостоятельно).

Качество кода — 2 (понятные имена, читаемая структура, осмысленное использование функций и классов).

Полнота решения — 3 (программа работает на контрольных примерах, проверяет ввод и обрабатывает основные ожидаемые ошибки).

Сроки — 1 (сдано вовремя).

В завершение освоения программы проводится итоговая аттестация в виде защиты индивидуальных или групповых проектов. Обучающиеся выбирают графическое приложение на PySide6 или бота ВКонтакте. Тему и обязательные возможности проекта согласуют с педагогом. Использование внешнего API и сохранение данных в файл определяются задачей проекта. Список предлагаемых тем согласуется с учебной частью не позднее чем за 3 месяца до итоговой аттестации.

Индивидуальный/групповой проект оценивается формируемой комиссией по 50-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице:

Таблица 5.

Баллы, набранные обучающимся	Уровень освоения
0–20 баллов	низкий
21–35 баллов	средний
36–50 баллов	высокий

Критерии и баллы итоговой аттестации:

Соответствие требованиям и сценарию пользователя — 5 (проект решает заявленную задачу, сценарии воспроизводимы).

ООП и качество кода — 8 (осмысленные классы, атрибуты и методы, разделение задач между объектами и функциями, понятные имена и читаемый код).

Интерфейс и взаимодействие — 7 (для PySide6: понятное расположение элементов и корректная обработка действий; для бота ВКонтакте: понятные команды, меню и последовательность диалога).

Работа с данными — 6 (подходящие структуры данных и корректная обработка значений; сохранение и загрузка из файла — если предусмотрены задачей проекта).

Проверка ввода и обработка ошибок — 6 (обработка неправильного ввода и основных ожидаемых ошибок, понятные сообщения пользователю).

Функциональная полнота — 10 (реализованы согласованные основные возможности, демонстрационный сценарий выполняется корректно).

Описание и защита проекта — 6 (краткая инструкция по запуску, понятная демонстрация, объяснение кода и ответы на вопросы).

Сроки — 2 (работающий проект и материалы к защите представлены вовремя).

Обязательность защиты. Защита проекта — обязательный этап завершения программы. При положительных результатах защиты обучающиеся получают сертификат, подтверждающий успешное освоение программы.

Состав комиссии (не менее 3 человек): в обязательном порядке входит педагог; приветствуется привлечение ИТ - профессионалов, представителей высших и других учебных заведений, администрации учебной организации.

Компонентами оценки индивидуального проекта являются (по мере убывания значимости): качество ИП, отзыв педагога, уровень презентации и защиты проекта. Если проект выполнен группой учащихся, то при оценивании

учитывается не только уровень исполнения проекта в целом, а и личный вклад каждого из авторов. Решение принимается коллегиально.

2.4 Методическое обеспечение

Дидактические материалы для дополнительной образовательной программы «Python - программирование» стартового уровня представляют собой комплекс ресурсов, направленных на успешное освоение материала и закрепление полученных знаний и навыков. Материалы разработаны таким образом, чтобы соответствовать потребностям начинающих программистов и способствовать постепенному погружению в мир Python-программирования.

Структура дидактических материалов:

Рабочая тетрадь представляет собой сборник упражнений и заданий, соответствующих каждой теме курса. Она предназначена для самостоятельной работы и самоконтроля. Каждая тема сопровождается пояснениями, примерами решений и контрольными заданиями различной степени сложности.

Примеры содержимого рабочей тетради:

Задания на выполнение арифметических операций и работу с переменными. Упражнения на использование условных операторов и циклов. Примеры задач на обработку строк и списков. Простые проекты, направленные на разработку небольших программ.

Презентация педагога используется для объяснения новых тем и демонстрации ключевых моментов. Она содержит иллюстративный материал, схемы, таблицы и фрагменты кода, облегчающие восприятие учебного материала.

Пример содержания презентации: Введение в Python: структура программы, синтаксис, среда разработки. Типы данных и операции над ними. Управление потоком исполнения: if-else конструкции, while и for циклы. Работа с функциями и модулями.

Тестовые задания позволяют оценить уровень усвоения материала студентами. Они представлены в форме вопросов с выбором правильного ответа, открытого типа или заполнения пропусков. Тесты охватывают весь учебный материал и могут служить инструментом промежуточного контроля.

Проектные задания являются важной частью обучения и направлены на применение полученных знаний на практике. Студенты разрабатывают небольшие программы, соответствующие уровню начального этапа обучения. Проекты сопровождаются четкими техническими заданиями и рекомендациями по выполнению.

Примеры проектов: графическая викторина, калькулятор или список дел на PySide6; бот-викторина, справочник или помощник ВКонтакте. Итоговый проект предполагает применение классов и объектов, проверку ввода и объяснение кода обучающимся.

Онлайн-платформа поддержки обеспечивает удаленный доступ к учебным ресурсам, возможность отправки выполненных заданий и получение обратной связи от преподавателя. Платформа позволяет отслеживать прогресс каждого обучающегося и своевременно выявлять трудности в обучении.

Использование перечисленных дидактических материалов позволит студентам успешно освоить базовые концепции программирования на языке Python и подготовиться к продолжению обучения на следующем уровне.

Список литературы

Основные источники

1. Лутц, М. Изучаем Python. Т. 1 / Марк Лутц ; [пер. с англ. Ю. Н Артеменко]. — 5-е изд. — Москва : Диалектика ; СПб., 2020. — 824 с.

2. Поляков, К.Ю. Информатика. Углубленный уровень. Учебник для 10 класса : в 2 ч. / К.Ю. Поляков, Е.А. Еремин. – М. : БИНОМ. Лаборатория знаний, 2019. – 352 с.

3. Окулов, С.М. Задачи по программированию. 2-е изд. / С.М. Окулов. – М. : БИНОМ. Лаборатория знаний, 2019. – 823 с.

Дополнительные источники

1. Вейдт, В.П. О воспитании подростков: в помощь классному руководителю : методическое пособие / В.П. Вейдт.– Калининград : Издательство Калининградского областного института развития образования, 2019.– 140 с.

2. Сэнд, У. Hello World! Занимательное программирование / У. Сенд, К. Сенд. – СПб. : Питер, 2016. – 400 с.

3. Джексон, Н. Классный учитель: Как работать с трудными учениками, сложными родителями и получать удовольствие от профессии / Н. Джексон. – Москва : Альпина Диджитал, 2015. – 332 с.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Домашняя страница Python [Электронный ресурс]. - URL: http://www.python.org (дата обращения: (дата обращения 04.08.2025).

2. Справочные материалы, официальная документация Python. Введение в программирование [Электронный ресурс]. - URL: https://younglinux.info/python.php (дата обращения 04.08.2025).

3. Питонтьютор. Бесплатный курс по программированию с нуля [Электронный ресурс]. - URL: https://pythontutor.ru/ (дата обращения 04.08.2025).

4. Python 3 для начинающих [Электронный ресурс]. - URL: https://pythonworld.ru/samouchitel-python (дата обращения 04.08.2025).

5. Документация PySide6 [Электронный ресурс]. - URL: https://doc.qt.io/qtforpython/ (дата обращения 04.08.2025)

6. Документация requests [Электронный ресурс]. - URL: https://requests.readthedocs.io/en/latest/ (дата обращения 04.08.2025).

7. Как создать бота ВКонтакте: гайд для старта на платформе. Блог компании VK [Электронный ресурс]. - URL: https://habr.com/ru/companies/vk/articles/570486/ (дата обращения 23.09.2026).