



**Департамент образования
АДМИНИСТРАЦИЯ ГОРОДА НИЖНЕГО НОВГОРОДА
Муниципальное бюджетное общеобразовательное учреждение
«Школа №189»
(МБОУ «Школа № 189»)**

УТВЕРЖДЕНА
приказом
директора МБОУ
«Школа № 189»
И.В. Довгаль
от 14.07.2025 № 5-О

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0042A0004E400D89948E7860908F700EE
Владелец: Довгаль Ирина
Владимировна. Действителен: с
29.06.2025 до 22.08.2026

**Дополнительная общеобразовательная программа-
дополнительная общеразвивающая программа
«Программирование на Python»**

Направленность: техническая
Возраст обучающихся: 10 - 14 лет
Срок реализации: 1 год

**Нижний Новгород
2025-2026
учебный год**

Содержание

1.	Пояснительная записка	стр.3-5
2.	Учебный план	стр. 6
3.	Учебно-тематический план	стр. 6
4.	Календарный учебный график	стр. 7
5.	Рабочая программа	стр. 8-9
6.	Оценка достижения результатов	стр. 9-10
7.	Организационно-педагогические условия	стр. 10-11
8.	Материально-техническое обеспечение	стр. 12
9.	Методическое обеспечение	стр. 12
10.	Список литературы и интернет-ресурсов	стр. 13

1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» (далее – Программа) направлена на мотивацию к обучению и проявление интереса обучающихся к ИТ-технологиям, приобретению навыков программирования, исследовательских и инженерных способностей в области информатики и ИТ-проектирования на языке Python.

Программа разработана с учетом следующих нормативных документов:

- Федеральный закон от 29.12.2012 №273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);

- Распоряжение Правительства РФ от 31.03.2022г. №678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 года».

- Письмо Министерства просвещения Российской Федерации от 29.09.2023г. «Методическими рекомендациями по формированию механизмов обновления содержания, методов и технологий обучения в системе дополнительного образования детей, направленных на повышение качества дополнительного образования детей, в том числе включение компонентов, обеспечивающих формирование функциональной грамотности и компетентностей, связанных с эмоциональным, человека, значимых для вхождения Российской Федерации в число десяти ведущих стран мира по качеству общего образования, для реализации приоритетных направлений научно-технологического и культурного развития страны».

- Постановление главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания и обучения, отдыха и оздоровления детей и молодежи»;

- Постановление Главного государственного санитарного врача от 28.01.2021г. №2 «Об утверждении санитарных правил и норм СП 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

- Приказ Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении порядка организации осуществления образовательной деятельности по дополнительным общеобразовательным программам».

Направленность – техническая. Программа ориентирована на развитие исследовательских и инженерных способностей учащихся в области информатики и ИТ-проектирования на языке Python. Содержание сочетает теорию и практику с постепенным переходом к созданию учебных проектов (игры, мини-приложения, простые базы данных), что обеспечивает формирование универсальных учебных действий и навыков XXI века (логика, критическое мышление, командная работа, презентация результатов).

Новизна Программы заключается в сочетании алгоритмического мышления и проектной деятельности. В отличие от традиционных занятий по информатике, курс ориентирован не только на

решение учебных задач, но и на создание собственных цифровых продуктов (мини-игр, простых приложений, графических проектов).

Актуальность: Язык Python — один из самых востребованных и доступных для освоения языков программирования школьниками; он используется в веб-разработке, анализе данных, автоматизации, создании игр и графических приложений. Доступный синтаксис Python позволяет сосредоточиться на алгоритмическом мышлении и проектной работе, а богатая экосистема библиотек (Tkinter, Pygame, sqlite3 и др.) делает возможным быстрый выход на практический результат.

Цель: создать условия для освоения основ алгоритмизации и программирования на Python через выполнение учебных и мини-проектов, формируя у обучающихся уверенные практические навыки работы с кодом и культуру инженерного мышления.

Задачи программы:

образовательные:

- познакомить с базовыми конструкциями языка Python, работой со строками, списками, циклами и условиями;

- ввести функции, основы ООП;

познакомить с библиотеками Tkinter и Pygame;

- показать основы работы с файлами и простейшими БД (sqlite3);

развивающие:

- формировать алгоритмическое и логическое мышление;

- навыки командной разработки (Git/GitHub);

- навыки отладки, тестирования и рефакторинга;

воспитательные:

- развивать ответственность за качество результата, культуру представления проекта (презентация, защита), умение сотрудничать и помогать одноклассникам, следовать нормам цифровой этики и безопасности.

Планируемые результаты освоения программы:

Личностные результаты:

- устойчивый интерес к программированию и цифровым технологиям как средству самореализации и решения практических задач;

- мотивация к обучению, саморазвитию и осознанному выбору индивидуальной образовательной траектории в сфере ИТ;

- ответственное отношение к результатам собственной деятельности и деятельности команды;

- готовность к сотрудничеству, уважительное отношение к мнению других участников образовательного процесса;

- способность к самооценке, рефлексии и критическому анализу собственных достижений;

- понимание значимости соблюдения норм цифровой культуры и этики при работе с программным кодом и информацией.

Метапредметные результаты:

- планировать свою деятельность при решении учебных и проектных задач, ставить цели и выбирать способы их достижения;

применять логическое, алгоритмическое и критическое мышление при анализе и решении задач;

- использовать различные источники информации, включая техническую документацию, для поиска и обработки необходимых данных;

- работать с современными цифровыми инструментами и средами разработки, адаптируясь к изменяющимся условиям цифровой среды;

- осуществлять контроль и коррекцию своей деятельности, выявлять и исправлять ошибки с использованием средств отладки и тестирования;

- взаимодействовать в группе: распределять роли, договариваться, аргументировать свою позицию и принимать совместные решения;

- представлять результаты своей деятельности в устной и визуальной форме, участвовать в обсуждении и защите проектов.

Предметные результаты:

- понимать синтаксис и особенности языка программирования Python;

- использовать базовые типы данных, арифметические операции, условные конструкции и циклы;

- разрабатывать и применять функции для структурирования и оптимизации программного кода;

- соблюдать культуру программирования и стандарты оформления кода (PEP 8);

- ориентироваться в документации модулей и библиотек Python и применять её на практике;

- использовать инструменты отладки, тестирования и контроля качества кода;

- создавать простые программные проекты, решающие реальные практические задачи;

- получать представления о кроссплатформенной разработке и этапах проектной деятельности в программировании.

Объём нагрузки в неделю составляет 2 час.

Количество учебных часов: 68 часов в год.

Итогом работы является защита проекта.

2. Учебный план

№ п/п	Наименование разделов и тем	Количество часов			Виды контроля
		Теорет.	Практ.	Всего	
1	Вводное занятие. Организация среды, ТБ	2	0	2	Инструктаж
2	Основы программирования на Python	10	12	22	Проверочная работа
3	Функции и основы командной разработки (Git/GitHub)	4	6	10	Мини-практикум
4	Графика на Tkinter	3	5	8	Практическая работа
5	Игры на Pygame	6	10	16	Практическая работа
6	ООП и работа с файлами/БД (sqlite3)	2	4	6	Практическая работа
7	Проект и защита	0	4	4	Защита проекта
	Всего	27	41	68	

3. Учебно-тематический план

№ п/п	Наименование разделов и тем	Количество часов			Виды контроля	
		Теорет.	Практ.	Всего		
1	Модуль 1. Основы программирования на Python					
1.1	Введение в курс и Python. Рабочая среда		2	2	4	Инструктаж, практическая работа
1.2	Типы данных и операции		2	2	4	Практические задания
1.3	Условия и логика		2	2	4	Самостоятельная работа
1.4	Циклы		2	2	4	Практическая работа
1.5	Списки и кортежи		2	2	4	Практическая работа
1.6	Итог по разделу		1	1	2	Контрольная работа
2	Модуль 2. Функции и основы командной разработки				10	
2.1	Введение в функции		2	2	4	Практическая работа
2.2	Области видимости переменных		1	1	2	Самостоятельная работа
2.3	Git и основы командной работы		1	1	2	Практическая работа
2.4	GitHub и совместная разработка		1	1	2	Взаимоконтроль обучающихся
3	Модуль 3. Графика на Tkinter				8	
3.1	Основы графического интерфейса		1	1	2	Практическая работа

	Tkinter				
3.2	События и обработка действий	1	1	2	Практическая работа
3.3	Мини-проект «Таймер / Кликер»	1	3	4	Мини-проект
4	Модуль 4. Игры на Pygame			16	
4.1	Введение в Pygame. Игровой цикл	2	2	4	Практическая работа
4.2	Отрисовка объектов и графика	2	2	4	Практическая работа
4.3	Управление и события	2	2	4	Самостоятельная работа
4.4	Мини-проект «Аркада»	1	3	4	Проектная работа
5	Модуль 5. ООП и работа с файлами/БД			6	
5.1	Основы объектно-ориентированного программирования	1	1	2	Практическая работа
5.2	Работа с файлами	1	1	2	Самостоятельная работа
5.3	Базы данных SQLite	1	1	2	Практическая работа
6	Модуль 6. Проект и защита			6	
6.1	Итоговый проект: разработка	1	3	4	Проектная работа
6.2	Защита проекта	1	1	2	Презентация, самоанализ
	Всего	30	38	68	

4. Календарный учебный график

№ п/п	Основные характеристики образовательного процесса	
1	Количество учебных недель	34
2	Количество учебных дней	68
3	Количество учебных часов в неделю	1
4	Количество учебных часов	68
5	Недель в 1 полугодии	17
6	Недель во 2 полугодии	17
7	Начало занятий	1 октября
8	Каникулы	31 декабря – 12 января
9	Выходные дни	Суббота, воскресенье, праздничные дни
10	Окончание учебного года	25 мая
11	Сроки промежуточной аттестации	май

5. Рабочая программа

Тема	№ занятия	Задачи	Содержание	Часы
Введение в курс и Python	1–2	Познакомить с целями курса, настроить рабочую среду, освоить базовый синтаксис	Python 3.x, интерфейс PyCharm/VS Code; «Hello, world!», переменные, ввод/вывод, арифметика	4
Типы данных	3–4	Научиться работать с текстом и числами	int, float, str; преобразование типов; методы строк; практикум на ввод/вывод и конкатенацию	4
Условия и логика	5–6	Освоить базовые конструкции ветвления	if/elif/else, логические операторы and/or/not; задачи на проверку условий	4
Циклы	7–8	Развить умение автоматизировать повторяющиеся действия	while, for, range; вложенные циклы; практикум «таблица умножения»	4
Списки и кортежи	9–10	Освоить работу с коллекциями	Создание списков, методы append/sort, срезы, кортежи; задачи на обработку данных	4
Итог по разделу	11	Проверить освоение базового материала	Контрольная работа: мини-задачи на ввод, условия, циклы, списки	2
Введение в функции	12–13	Научиться структурировать код	def, параметры, return; практика «калькулятор функций»	4
Области видимости	14	Понять различие локальных и глобальных переменных	Примеры и практика: счётчик вызовов функции	2
Git и командная работа	15	Освоить основы контроля версий	Локальный репозиторий, коммиты, структура проекта	2
GitHub и совместная разработка	16	Познакомиться с удалённым репозиторием	Создание аккаунта, push/pull, основы командной работы	2
Основы интерфейсов	17	Познакомиться с библиотекой Tkinter	Создание окна, надписи, кнопки, поля ввода	2
События и обработка	18	Научиться связывать интерфейс и логику	Привязка функций к кнопкам, обработка событий	2
Мини-проект «Таймер/Кликер»	19–20	Разработать простое приложение	Логика счётчика, обновление текста, оформление окна	4

Введение в Pygame	21–22	Освоить базовый цикл игры	Инициализация Pygame, создание окна, игровой цикл	4
Отрисовка объектов	23–24	Научиться работать с графикой	Примитивы, изображения, загрузка спрайтов, фон	4
Управление и события	25–26	Реализовать взаимодействие игрока с программой	События клавиатуры и мыши, движение персонажа	4
Мини-проект «Аркада»	27–28	Собрать полноценный прототип игры	Создание персонажа, счётчик очков, завершение игры	4
Основы ООП	29	Познакомиться с понятием класса и объекта	class, методы, конструктор <code>__init__</code>	2
Работа с файлами	30	Научиться сохранять данные	Чтение и запись файлов (txt, csv)	2
sqlite3 – простейшие базы данных	31	Получить опыт работы с БД	Создание таблицы, запись и чтение данных	2
Итоговый проект – разработка	32–33	Завершить индивидуальный или парный проект	Доработка игры/приложения, добавление функций, отладка	4
Защита проекта	34	Презентовать результат	Структура защиты: цель, задачи, демонстрация, ответы на вопросы	2

6. Оценка достижения результатов

Промежуточная аттестация освоения обучающимися Программы проводится 1 раз в год после завершения всех модулей обучения (май).

Оценка достижения планируемых результатов освоения программы осуществляется по трем уровням: высокий (от 80 до 100% освоения программного материала), средний (от 51 до 79% освоения программного материала), низкий (менее 50% освоения программного материала).

Уровни освоения	Результат
Высокий уровень освоения программы	Обучающиеся демонстрируют высокую заинтересованность в учебной, познавательной и творческой деятельности, составляющей содержание программы. На итоговом тестировании показывают отличное знание теоретического материала, практическое применение знаний воплощается в качественный продукт.

Средний уровень освоения программы	Обучающиеся демонстрируют достаточную заинтересованность в учебной, познавательной и творческой деятельности, составляющей содержание Программы. На итоговом тестировании показывают хорошее знание теоретического материала, практическое применение знаний воплощается в продукт, требующий незначительной доработки.
Низкий уровень освоения программы	Обучающиеся демонстрируют низкий уровень заинтересованности в учебной, познавательной и творческой деятельности, составляющей содержание программы. На итоговом тестировании показывают недостаточное знание теоретического материала, практическая работа не соответствует требованиям.

Формы подведения итогов

Для подведения итогов в программе используются промежуточные формы контроля: тестовые и практические задания, защита индивидуальных или парных проектов (игра/приложение на Python).

7. Организационно-педагогические условия реализации Программы

Адресат Программы: обучающиеся от 10 до 14 лет.

Срок реализации: 1 год

Форма обучения: очная.

Режим занятий: занятия проводятся один раз в неделю, 2 часа по 40 минут, перерыв между занятиями 10 минут.

Наполняемость группы: от 15 до 25 человек.

Основные принципы, заложенные в основу Программы:

- научности (детям сообщаются знания об алгоритмах, структурах данных, принципах построения программ);
- динамичности (изучение материала от простого к сложному: от базового синтаксиса к проектной разработке);
- сотрудничества (совместная деятельность педагога и обучающихся, работа в парах и командах);
- системности (учебный процесс выстроен в систему заданий и проектов, обеспечивающих постепенное усложнение материала);
- преемственности (каждый следующий модуль базируется на ранее сформированных навыках и расширяет «зону ближайшего развития» обучающихся);
- наглядности (использование демонстрационных программ, примеров кода, визуализации с помощью графических библиотек);

- информационно-коммуникационных технологий (применение современных сред разработки, Git/GitHub, онлайн-ресурсов);

- здоровьесбережения (рациональное сочетание теоретической и практической работы, чередование видов деятельности за компьютером, соблюдение правил ТБ и цифровой гигиены).

Методы и приемы, используемые на занятиях:

- вербально-информационные (объяснение нового материала, рассказ, беседа, инструктаж, обсуждение способов решения задач);

- наглядные (демонстрация примеров программного кода, алгоритмов, схем, интерфейсов программ, показ работы в средах разработки PyCharm и VS Code, визуализация алгоритмов);

- практические (пошаговая отработка навыков программирования, выполнение практических заданий, написание и отладка программ, разработка мини-проектов и игр);

- аналитические (анализ условий задач, поиск и исправление ошибок в коде, сравнение различных способов решения, самоконтроль и самооценка результатов деятельности);

- эвристические (поиск нестандартных решений, выполнение творческих и проектных заданий, разработка собственных программ, игр и приложений).

Выбор методов обучения определяется содержанием занятия, уровнем подготовки и индивидуальными особенностями обучающихся. На занятиях преобладают методы практической деятельности с показом и пошаговой демонстрацией приёмов программирования, использованием цифровых образовательных ресурсов и технических средств обучения. Значительное внимание уделяется анализу и самоанализу программного кода, поиску путей повышения его качества, а также **чередованию** видов деятельности (работа за компьютером, обсуждение решений, защита идей и результатов).

Формы организации занятий:

- практические занятия;

- индивидуальная работа за компьютером;

- парная и групповая работа;

- мини-проекты и проектная деятельность;

- викторины и интерактивные задания по программированию;

- обсуждение и защита выполненных работ;

- участие в конкурсах и презентациях проектов.

Основной формой проведения занятий является практическая работа, направленная на освоение знаний, умений и навыков программирования на языке Python и работу с современными инструментами разработки. На всех занятиях проводится инструктаж по технике безопасности при работе с компьютерной техникой. Решению воспитательных и развивающих задач

способствует выполнению проектных работ, командное взаимодействие и публичная презентация результатов деятельности. Учебно-методический комплекс к Программе разрабатывается педагогом дополнительного образования, реализующим Программу, и хранится на базе проведения занятий.

8. Материально-техническое обеспечение

Учебный кабинет информатики, оформленный в соответствии с профилем проводимых занятий и оборудованный в соответствии с санитарными нормами:

Специализированная мебель и системы хранения:

- рельсовая система с классной доской;
- стол с тумбой;
- кресло офисное;
- шкаф для хранения учебных пособий;
- доска магнитно-маркерная;
- система (устройство) для затемнения окон;
- стул ученический поворотный, регулируемый по высоте;
- стол ученический, регулируемы по высоте с компьютерами;

Технические средства обучения:

- ноутбук (лицензионное программное обеспечение, образовательный контент и система защиты от вредоносной информации);
- принтер.

9. Методическое обеспечение

Информационное обеспечение Программы:

- электронные учебники и пособия по Python;
- официальная документация Python и библиотек (docs.python.org);
- обучающие презентации по темам: «Основы Python», «Условные конструкции и циклы», «Функции», «Tkinter», «Игры на Pygame», «ООП», «Работа с файлами и базами данных»;
- образовательные видеофильмы и онлайн-курсы по программированию для школьников;
- интернет-ресурсы (ru-python.org, stepik.org, learnpython.org).

Дидактические материалы:

- методические материалы (карточки с заданиями, примеры кода, демонстрационные проекты);

- подборки скриншотов, схем, блок-схем алгоритмов;
- рабочие тетради/портфолио обучающихся с кодом и конспектами.

10. Список литературы и интернет-источников

Для педагога:

1. Лутц М. *Изучаем Python*. – СПб.: Питер, 2021.
2. Свейгарт А. *Автоматизация рутинных задач с помощью Python*. – СПб.: Питер, 2020.
3. Шоу З. *Лёгкий способ выучить Python*. – СПб.: Питер, 2019.
4. Документация Python 3. <https://docs.python.org/3/>
5. PEP 8 – Style Guide for Python Code. <https://peps.python.org/pep-0008/>
6. Методические материалы для педагогов: презентации, план-конспекты занятий, учебные задачи по Python.

Для обучающихся:

1. A Byte of Python (рус. перевод). – доступно онлайн.
2. Эрик Мэтиз. *Изучаем Python. Программирование игр, визуализация данных, веб-приложения*. – СПб.: Питер, 2021.
3. Свейгарт А. *Python для детей. Самоучитель по программированию*. – СПб.: Питер, 2018.
4. Пособия и курсы на образовательных платформах: Stepik.org («Программирование на Python»), Yandex.Practicum.
5. Учебные материалы по Tkinter и Pygame (онлайн-руководства, пошаговые проекты).

Интернет ресурсы:

1. <https://www.python.org/> – официальный сайт Python.
2. <https://docs.python.org/3/> – официальная документация Python 3.
3. <https://pypi.org/> – каталог пакетов Python.
4. <https://stepik.org/course/67/> – онлайн-курс «Программирование на Python».
5. <https://habr.com/ru/hub/python/> – статьи по Python на Habr.
6. <https://github.com/> – хостинг проектов и учебных репозиторийев.
7. <https://pygame.org/> – официальный сайт библиотеки Pygame.
8. <https://sqlite.org/> – документация по SQLite.