

Муниципальное автономное общеобразовательное учреждение города
Новосибирска
«Инженерный лицей Новосибирского государственного технического
университета»

РЕКОМЕНДОВАНА
решением педагогического совета
МАОУ «Инженерный лицей НГТУ»
Протокол от 27.08.2026 № 1

УТВЕРЖДЕНА
приказом директора МАОУ
«Инженерный лицей НГТУ»
от 27.08.2026 №260



**Рабочая программа по курсу внеурочной
деятельности «Программирование Python»**

название учебного предмета, курса, дисциплины (модуля)

для класса(ов) 7-9

Количество часов: 2 часа в неделю

Всего: 202 часа

Из них: 7 класс 68 часа

8 класс 68 часа

9 класс 66 часа

Разработчики программы:

Орлова Анна Алексеевна, учитель информатики первой квалификационной категории.

Программа обсуждалась на заседании кафедры / методического объединения
учителей математики и информатики МАОУ «Инженерный лицей НГТУ»
Протокол заседания № 1 от 27.08.2026 г.

Подольн / Подольн Е.В
Ф.И.О. руководителя кафедры/МО

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая программа курса внеурочной деятельности «Программирование на Python» разработана для обучающихся 7–9 классов. Настоящий курс носит практико-ориентированный характер и направлен на расширение базовых знаний обучающихся в области алгоритмизации, логики и информационных технологий.

Основное внимание в курсе уделяется не просто освоению готовых информационных технологий, а обучению основам их создания. Изучение алгоритмизации позволяет перевести теоретические знания учащихся в плоскость конкретной практической деятельности, развивая системное и критическое мышление.

Данный курс является вводным курсом по программированию, дающий представление о базовых понятиях структурного программирования (данных, операциях, переменных, ветвлениях в программе, циклах и функциях). Выбор Python обусловлен тем, что это язык, обладающий рядом преимуществ перед другими языками для начинающих изучать программирование – ясность кода, быстрота реализации. По результатам исследования компании JetBrains (<https://www.jetbrains.com/ru-ru/lp/devecosystem-2019/>) язык Python – самый изучаемый и один из самых востребованных на рынке труда.

Изучение Python в школе откроет ученикам возможности дальнейшего развития в области ИТ и поможет профориентации в старших классах, пригодится в олимпиадах по программированию и решению заданий ЕГЭ.

Курс предполагает смешанный формат обучения: очное и онлайн-обучение. На очных занятиях в группах по 10-15 человек используются задания на онлайн-платформе с автоматизированной проверкой. Теоретические блоки и задания на платформе доступны ученикам для самостоятельного изучения в любое время. В ходе курса запланирован плавный переход к формату «перевернутого класса». На первых пяти занятиях ученики знакомятся с онлайн-платформой и основными инструментами программирования на Python. Начиная с шестого урока, часть классной работы организуется в форме самостоятельной деятельности с групповой взаимопроверкой. После десятого урока ученикам предлагается самостоятельное изучение новой темы с опорой на конспект онлайн-курса, а учитель является консультантом и модератором. Завершающий этап курса посвящён самостоятельной работе над проектами. Сочетание групповой работы с учителем в классе и индивидуальной работы в личном кабинете на онлайн-платформе Stepik позволяет ученикам выработать не только технические навыки программирования, но и навыки социального взаимодействия при работе над финальным проектом курса, а главное – научиться самостоятельно выстраивать свое профессиональное развитие.

Отличительной особенностью курса является его доступность и логичность, подкреплённая мощным дидактическим материалом, направленным на практику программирования и подготовку к олимпиадам, на развитие творчества и самостоятельности обучающихся. На занятиях организуется деятельность, создающая условия для творческого развития школьников на различных возрастных этапах и предусматривающая их дифференциацию по степени одарённости. Он позволяет повысить результативность обучения информатики и ИКТ при параллельном преподавании школьного основного курса и данного дополнительного, благодаря авторскому подходу к организации образовательного процесса типовой программы Благотворительного фонда развития образования «АйкьюОпшн» («Возможность Интеллекта»).

Основной целью курса является – формирование у обучающихся навыков программирования на языке Python и самонаправленного обучения.

Достижение цели предполагает решение следующих задач:

1. Сформировать основные навыки программирования на языке Python.
2. Научить применять полученные знания для решения практических задач.
3. Повысить уровень самостоятельности в обучении (по четырехступенчатой шкале Г. Гроу).

4. Научить декомпозировать задачи и работать над ними в небольших командах, с разделением ответственности и ролей.

СОДЕРЖАНИЕ КУРСА

7 класс

Тема 1. Знакомство с Python (2 ч.)

Понятие алгоритма и программы. Язык программирования Python, его особенности и преимущества. Структура программы на Python. Интерфейс среды разработки. Составление простейшей программы, её запуск. Комментарии в программе. Справочная система среды программирования. Правила оформления кода PEP8: отступы, длина строки, пустые строки, пробелы в выражениях, именованное переменных и функций, комментарии, импорт модулей.

Тема 2. Переменные и выражения (6 ч.)

Понятие переменной. Типы данных: целые числа, вещественные числа, строки. Оператор присваивания. Арифметические операции: сложение, вычитание, умножение, деление, целочисленное деление, остаток от деления, возведение в степень. Приоритет арифметических действий. Запись арифметических выражений на Python. Операторы ввода и вывода данных. Функции `input()` и `print()`. Преобразование типов данных. Решение задач на элементарные действия с числами.

Тема 3. Условный оператор (6 ч.)

Назначение условного оператора. Способы записи условного оператора. Конструкция `if`. Конструкция `if-else`. Конструкция `if-elif-else`. Вложенные условия. Логические операторы `or`, `and`, `not`. Создание сложных условий с помощью логических операторов. Логические выражения и их вычисление. Запись логических выражений на Python. Решение задач по теме «Условные операторы». Составление программ с конструкциями ветвления, используя простые и сложные условия.

Тема 4. Циклы (10 ч.)

Понятие цикла. Виды циклов. Цикл с условием `while`. Правила записи цикла с условием. Цикл с параметром `for`. Назначение и особенности использования цикла с параметром, форматы записи. Функция `range()`. Использование операторов `for` и `while` при решении задач. Реализация циклических алгоритмов. Составление программ с циклом. Определение вида цикла, наиболее удобного для решения поставленной задачи. Вложенные циклы. Операторы `break` и `continue`.

Тема 5. Работа со строками (10 ч.)

Назначение строкового типа данных. Операторы для работы со строками. Процедуры и функции для работы со строками. Операции со строками при решении задач: описание строк; соединение строк; нахождение длины строки; вырезание части строки; нахождение подстроки в строке; нахождение количества слов в строке; нахождение количества символов в строке; нахождение заданного символа в строке. Индексация символов. Срезы строк. Методы строк. Перебор символов строки.

Тема 6. Работа со списками (10 ч.)

Основы работы со списками. Создание списков. Индексация элементов списка. Методы списков при решении задач. Вывод элементов списка. Строковые методы `split()` и `join()`. Списочные выражения (генератор списка). Решение задач со списочными выражениями. Перебор элементов списка. Добавление, удаление и изменение элементов списка. Сортировка списков. Вложенные списки.

Тема 7. Функции (10 ч.)

Способы описания функции. Принципы структурного программирования при решении задач. Понятие формальных и фактических параметров подпрограмм. Различные способы передачи параметров подпрограмм. Создание и использование простых функций. Создание и использование рекурсивных функций. Использование функций при решении задач. Механизм параметров для передачи значений в программах. Возврат значений из функции. Функции с аргументами разных типов. Область видимости переменных.

Тема 8. Локальные и глобальные переменные (6 ч.)

Понятия локальных и глобальных переменных. Различия между локальными и глобальными переменными. Функции, возвращающие значения. Использование локальных и глобальных переменных в подпрограммах. Принципы структурного программирования при решении задач. Механизм параметров для передачи значений в программах. Ключевое слово `global`.

Тема 9. Работа над итоговым проектом (8 ч.)

Умение применять полученные знания на практике. Самостоятельно планировать свою работу и распределять время. Использовать доступные ресурсы для достижения цели. Оформлять и представлять результаты работы. Отвечать на вопросы по проекту. Защищать индивидуальные и групповые проекты.

8 класс

Тема 1. Повторение основных конструкций Python (6 ч.)

Структура программы на Python. Режимы работы с Python. Команды редактора программы. Организация ввода и вывода данных. Решение задач по теме «Условные операторы». Реализация циклических алгоритмов. Составление программ с циклом. Повторение строк, списков, функций.

Тема 2. Тип данных `bool` и `NoneType` (2 ч.)

Логический тип данных `bool`. Тип `NoneType`. Логические операторы. Булевы значения как числа. Функции `bool()`, `type()`, `isinstance()`. Использование логических значений в программах. Преобразование типов.

Тема 3. Вложенные списки (8 ч.)

Основы работы со списками. Методы списков при решении задач. Вывод элементов списка. Строковые методы `split()` и `join()`. Списочные выражения. Конструкция вложенных списков при решении задач. Двумерные массивы. Обработка вложенных списков. Перебор элементов вложенного списка.

Тема 4. Кортежи (6 ч.)

Основы работы с кортежами. Создание кортежей. Использование кортежей при решении задач. Вывод кортежа. Методы кортежей. Перебор кортежей. Сравнение кортежей. Сортировка кортежей. Преобразование кортежа в список и строку. Упаковка кортежей. Распаковка кортежей. Присваивание кортежей. Вложенные кортежи при решении задач.

Тема 5. Множества (10 ч.)

Определение множества в математике. Создание множеств. Операции над множествами: перебор, сравнение, сложение. Методы множеств. Использование методов множеств при решении задач. Объединение, пересечение, разность множеств. Симметрическая разность. Генераторы множеств.

Тема 6. Словари (6 ч.)

Операции создания, удаления, работы со словарями. Методы словарей. Основы работы со словарями. Использование методов словарей при решении задач. Вывод словаря. Различия методов словарей. Вложенные словари при решении задач. Перебор элементов словаря. Ключи и значения. Генераторы словарей.

Тема 7. Модули `decimal`, `fraction` и другие (8 ч.)

Работа с псевдослучайными числами. Методы модуля `random`. Метод Монте-Карло для статических экспериментов и испытаний. Дополнительные числовые типы данных модуля `decimal`: `Decimal` – десятичное число для выполнения точных расчётов; `Fraction` – число, представляющее собой обыкновенную дробь с заданным числителем и знаменателем; `Complex` – комплексное число. Использование числового типа данных `Fraction` для работы с обыкновенными дробями. Особенности модуля `turtle`. Основные команды работы с исполнителем Черепашка. Создание геометрических фигур и анимированных объектов с помощью исполнителя Черепашка.

Тема 8. Функции (10 ч.)

Способы описания функции. Принципы структурного программирования при решении задач. Понятие формальных и фактических параметров подпрограмм. Различные способы передачи параметров подпрограмм. Создание и использование простых функций. Создание и использование рекурсивных функций. Использование функций при решении задач. Механизм параметров для передачи значений в программах. Использование для функций аргументов разных типов. Функции высшего порядка. Лямбда-функции.

Тема 9. Работа с файлами (6 ч.)

Основные понятия, связанные с файлами. Методы обращения к файлу. Открытие и закрытие текстового файла. Обработка данных из текстового файла. Вывод результата в файл. Режимы открытия файлов. Чтение и запись данных. Работа с путями к файлам.

Тема 10. Работа над итоговым проектом (6 ч.)

Применять полученные знания на практике. Самостоятельно планировать работу и распределять время. Использовать доступные ресурсы для достижения цели. Находить конструктивные решения в трудных ситуациях. Оформлять и представлять результаты работы. Аргументированно отвечать на вопросы. Защищать индивидуальные и групповые проекты.

9 класс

Тема 1. Повторение основных конструкций Python (4 ч.)

Структура программы на Python. Режимы работы с Python. Команды редактора программы. Организация ввода и вывода данных. Решение задач по теме «Условные операторы». Реализация циклических алгоритмов. Составление программ с циклом. Методы строк, списков, кортежей, множеств. Организация в программе процедуры и функции. Обработка текстовых файлов.

Тема 2. Работа с датой и временем (6 ч.)

Типы данных: date, time, datetime, timedelta. Использование типов данных при решении задач. Модуль time. Модуль calendar. Применение возможностей модулей time и calendar в программах. Форматирование даты и времени. Вычисления с датами. Таймеры и задержки.

Тема 3. Работа с файлами (8 ч.)

Потоковый ввод stdin и вывод stdout для чтения файлов. Работа с файлами csv. Работа с файлами json. Работа с файлами zip. Работа с файлами pickle. Сериализация и десериализация данных. Чтение и запись структурированных данных. Обработка больших файлов.

Тема 4. Устройство памяти и особенности типов данных (5 ч.)

Принципы устройства памяти в Python. Принципы работы переменных в Python. Изменяемые и неизменяемые типы данных. Приёмы поверхностного и глубокого копирования объектов. Механизмы очистки памяти. Ссылки и объекты. Идентификаторы объектов. Сборка мусора.

Тема 5. Дополнительные типы коллекций (10 ч.)

Типы данных: list, tuple, set, dict, namedtuple, defaultdict, OrderedDict, Counter, ChainMap. Использование дополнительных типов коллекций при решении задач. Именованные кортежи. Словари с значением по умолчанию. Упорядоченные словари. Счётчики. Цепочки словарей. Сравнение и выбор подходящего типа коллекции.

Тема 6. Обработка исключений (5 ч.)

Три категории ошибок: синтаксические — возникают из-за синтаксических погрешностей кода; логические — проявляются вследствие логических неточностей в алгоритме; ошибки времени выполнения, исключения — вызваны некорректными действиями пользователя или системы. Обработка исключений. Конструкция try-except. Конструкция try-except-else-finally. Множественные исключения. Создание собственных исключений. Оператор raise.

Тема 7. Рекурсия (6 ч.)

Понятие рекурсии. Организация рекурсивной обработки данных. Базовый случай и рекурсивный шаг. Рекурсивные функции. Примеры рекурсивных алгоритмов. Глубина рекурсии. Переполнение стека. Сравнение рекурсии и итерации.

Тема 8. Функции (8 ч.)

Способы описания функции. Понятие формальных и фактических параметров подпрограмм. Различные способы передачи параметров подпрограмм. Создание и использование функций высшего порядка при решении задач. Использование функций как объектов. Вложенные функции и замыкания. Применение модуля `functools`. Декораторы. Аргументы по умолчанию. Произвольное число аргументов.

Тема 9. Итераторы и генераторы (10 ч.)

Определение и обработка итерируемых объектов. Использование коллекций и последовательностей. Создание итераторов. Создание генераторов. Отличие последовательности от итератора. Функция `iter()`. Функция `next()`. Генераторные выражения. `yield`. Ленивые вычисления. Бесконечные генераторы.

Тема 10. Регулярные выражения (6 ч.)

Принцип соответствия символов. Диапазон набора символов. Принцип соответствия с чем угодно кроме. Принцип соответствия одному из нескольких регулярных выражений. Метасимволы. Квантификаторы. Группы и скобки. Модуль `re`. Функции поиска, замены и разбиения строк. Компиляция регулярных выражений. Жадные и нежадные квантификаторы.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ КУРСА ОБУЧАЮЩИМИСЯ

Программа предполагает достижение обучающимися следующих личностных, метапредметных и предметных результатов.

В результате изучения курса «Программирование на Python» у обучающихся будут сформированы следующие **личностные результаты** в части:

Патриотическое воспитание:

- ценностное отношение к отечественному культурному, историческому и научному наследию;
- понимание значения информатики как науки в жизни современного общества.

Духовно-нравственное воспитание:

- ориентация на моральные ценности и нормы в ситуациях нравственного выбора;
- готовность оценивать своё поведение и поступки, а также поведение и поступки других людей с позиции нравственных и правовых норм с учётом осознания последствий поступков;
- активное неприятие асоциальных поступков, в том числе в Интернете.

Гражданское воспитание:

- представление о социальных нормах и правилах межличностных отношений в коллективе, в том числе в социальных сообществах;
- соблюдение правил безопасности, в том числе навыков безопасного поведения в интернет-среде;
- ориентация на совместную деятельность при выполнении учебных и познавательных задач, создании учебных проектов;
- стремление оценивать своё поведение и поступки своих товарищей с позиции нравственных и правовых норм с учётом осознания последствий поступков.

Ценность научного познания:

- наличие представлений об информации, информационных процессах и информационных технологиях, соответствующих современному уровню развития науки и общественной практики;
- интерес к обучению и познанию;
- стремление к самообразованию;
- овладение начальными навыками исследовательской деятельности, установка на осмысление опыта, наблюдений, поступков и стремление совершенствовать пути достижения индивидуального и коллективного благополучия;

- владение базовыми навыками самостоятельной работы с учебными текстами, справочной литературой и средствами ИКТ;
- умение самостоятельно определять цели обучения, ставить новые познавательные задачи и развивать мотивы познавательной деятельности.

Формирование культуры здоровья:

установка на здоровый образ жизни, в том числе и за счёт освоения и соблюдения требований безопасной эксплуатации средств ИКТ.

Трудовое воспитание:

интерес к практическому изучению профессий и труда в сферах деятельности, связанных с информатикой, программированием и информационными технологиями, основанными на достижениях науки информатики и научно-технического прогресса.

Экологическое воспитание:

наличие представлений о глобальном характере экологических проблем и путей их решения, в том числе с учётом возможностей ИКТ.

Адаптация обучающегося к изменяющимся условиям социальной среды: освоение обучающимися социального опыта, основных социальных ролей, соответствующих ведущей деятельности возраста, норм и правил общественного поведения, форм социальной жизни в группах и сообществах, в том числе в виртуальном пространстве.

Метапредметные результаты освоения курса характеризуются овладением универсальными *познавательными* действиями, универсальными *коммуникативными* действиями и универсальными *регулятивными* действиями.

Универсальные познавательные действия

Базовые логические действия:

- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логические рассуждения, делать умозаключения (индуктивные, дедуктивные и по аналогии) и выводы;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- самостоятельно выбирать способ решения учебной задачи (сравнивать несколько вариантов решения, выбирать наиболее подходящий с учётом самостоятельно выделенных критериев).

Базовые исследовательские действия:

- формулировать вопросы, фиксирующие разрыв между реальным и желательным состоянием ситуации, объекта, и самостоятельно устанавливать искомое и данное;
- оценивать применимость и достоверность информации, полученной в ходе исследования;
- прогнозировать возможное дальнейшее развитие процессов, событий и их последствия в аналогичных или сходных ситуациях, а также выдвигать предположения об их развитии в новых условиях и контекстах.

Работа с информацией:

- выявлять дефицит информации, данных, необходимых для решения поставленной задачи;
- применять основные методы и инструменты при поиске и отборе информации из источников с учётом предложенной учебной задачи и заданных критериев;
- выбирать, анализировать, систематизировать и интерпретировать информацию различных видов и форм представления;
- выбирать оптимальную форму представления информации и иллюстрировать решаемые задачи несложными схемами, диаграммами, иными графическими объектами и их комбинациями;
- оценивать достоверность информации по критериям, предложенным учителем или сформулированным самостоятельно;

- запоминать и систематизировать информацию.

Универсальные коммуникативные действия

Общение:

- сопоставлять свои суждения с суждениями других участников диалога, обнаруживать различие и сходство позиций; публично представлять результаты выполненного опыта (исследования, проекта);
- выбирать формат выступления с учётом задач презентации и особенностей аудитории и в соответствии с ним составлять устные и письменные тексты с использованием иллюстративных материалов.

Совместная деятельность (сотрудничество):

- понимать и использовать преимущества командной и индивидуальной работы при решении конкретной проблемы, в том числе при создании информационного продукта;
- принимать цель совместной информационной деятельности по сбору, обработке, передаче, формализации информации; коллективно строить действия по её достижению: распределять роли, договариваться, обсуждать процесс и результат совместной работы;
- выполнять свою часть работы с информацией или информационным продуктом, достигая качественного результата по своему направлению и координируя свои действия с другими членами команды;
- оценивать качество своего вклада в общий информационный продукт по критериям, самостоятельно сформулированным участниками взаимодействия;
- сравнивать результаты с исходной задачей и вклад каждого члена команды в достижение результатов, разделять сферу ответственности и проявлять готовность к предоставлению отчёта перед группой.

Универсальные регулятивные действия

Самоорганизация:

- выявлять в жизненных и учебных ситуациях проблемы, требующие решения; составлять алгоритм решения задачи (или его часть), выбирать способ решения учебной задачи с учётом имеющихся ресурсов и собственных возможностей, аргументировать выбор варианта решения задачи;
- составлять план действий (план реализации намеченного алгоритма решения), корректировать предложенный алгоритм с учётом получения новых знаний об изучаемом объекте.

Самоконтроль (рефлексия):

- владеть способами самоконтроля, самомотивации и рефлексии;
- учитывать контекст и предвидеть трудности, которые могут возникнуть при решении учебной задачи, адаптировать решение к меняющимся обстоятельствам;
- вносить коррективы в деятельность на основе новых обстоятельств, изменившихся ситуаций, установленных ошибок, возникших трудностей;
- оценивать соответствие результата цели и условиям.

Эмоциональный интеллект:

ставить себя на место другого человека, понимать мотивы и намерения другого.

Принятие себя и других:

- осознавать невозможность контролировать всё вокруг даже в условиях открытого доступа к любым объёмам информации;
- осознанно относиться к другому человеку, его мнению.

ПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

7 класс

К концу обучения в 7 классе обучающийся получит следующие предметные результаты:

Следовать требованиям безопасности при работе за компьютером. Использовать определения понятий: алгоритм, язык программирования, программа. Демонстрировать навыки алгоритмического мышления и понимания необходимости формального описания алгоритмов. Использовать переменные различных типов при написании программ на Python. Использовать оператор присваивания при написании программ на Python. Давать определения понятию логическое выражение. Вычислять значение логического выражения, записывать логическое выражение на Python. Реализовывать алгоритмы в виде программ и программных систем. Владеть элементарными навыками документирования программ. Сопоставлять аналитические модели задач и их компьютерные аналоги. Анализировать полученные результаты с точки зрения соответствия объекту и целям моделирования. Сравнивать разные способы записи алгоритмов как информационные модели. Владеть универсальным языком программирования высокого уровня. Применять базовые знания о типах и структурах данных при решении практических задач. Использовать основные управляющие конструкции языка, производить отладку программ с помощью встроенного отладчика.

8 класс

К концу обучения в 8 классе обучающийся получит следующие предметные результаты: Следовать требованиям безопасности при работе на компьютере; Владеть базовыми практическими знаниями по программированию на Python и применять их на практике. Разрабатывать программы на Python для рисования геометрических фигур, используя модуль Turtle. Применять практические базовые навыки разработки игр с помощью черепаший графики (библиотека turtle). Понимать различия локальных и глобальных переменных. Решать задачи с использованием глобальных переменных на Python. Сопоставлять модели задач и их компьютерные аналоги, анализировать полученные результаты с точки зрения соответствия объекту и целям моделирования. Анализировать разные способы записи алгоритмов как информационные модели. Владеть навыком написания собственных функций на Python. Разбивать задачи на подзадачи.

9 класс

К концу обучения в 9 классе обучающийся получит следующие предметные результаты: Следовать требованиям безопасности при работе на компьютере. Планировать действия, необходимые для решения задач высокого и повышенного уровня сложности с использованием языка программирования, оценивать результаты работы. Реализовывать сложные алгоритмы по обработке файлов в виде программ и программных систем. Определять необходимый тип данных, с учетом особенностей устройства памяти. Организовывать рекурсивную обработку данных. Сопоставлять знаковые модели задач и их компьютерные аналоги, анализировать полученные результаты с точки зрения соответствия объекту и целям моделирования. Анализировать разные способы записи алгоритмов как информационные модели. Владеть универсальным языком программирования высокого уровня на базисном уровне.

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

7 класс

№ п/п	Название раздела	Количество часов		
		Всего	Практические работы	Зачет
1	Знакомство с Python	2	1	
2	Переменные и выражения	6	3	
3	Условный оператор	6	3	1
4	Циклы	10	5	1
5	Работа со строками	10	5	

6	Работа со списками	10	5	1
7	Функции	10	5	1
8	Локальные и глобальные переменные	6	3	
9	Работа над итоговым проектом	8	6	

8 класс

№ п/п	Название раздела	Количество часов		
		Всего	Практические работы	Зачст
1	Повторение основных конструкций Python	6	3	
2	Тип данных bool и NoneType	2	1	
3	Вложенные списки	8	4	1
4	Кортежи	6	3	
5	Множества	10	5	1
6	Словари	6	3	1
7	Модули decimal, fraction и другие	8	4	
8	Функции	10	5	1
9	Работа с файлами	6	3	
10	Работа над итоговым проектом	6	4	

9 класс

№ п/п	Название раздела	Количество часов		
		Всего	Практические работы	Зачст
1	Повторение основных конструкций Python	4	2	
2	Работа с датой и временем	6	3	
3	Работа с файлами	8	4	1
4	Устройство памяти и особенности типов данных	5	2	1
5	Дополнительные типы коллекций	10	5	
6	Обработка исключений	5	2	
7	Рекурсия	6	3	1
8	Функции	8	4	
9	Итераторы и генераторы	10	5	1
10	Регулярные выражения	6	3	

ТЕХНИЧЕСКИЕ СРЕДСТВА

Требования к комплектации компьютерного класса

Для реализации учебного курса необходимо наличие компьютерного класса в соответствующей комплектации. Наиболее рациональным является установка в компьютерном классе 13–15 компьютеров (рабочих мест) для школьников и одного компьютера (рабочего места) для педагога. Предполагается объединение компьютеров в локальную сеть с возможностью выхода в Интернет, что позволяет использовать сетевые цифровые образовательные ресурсы.

Минимальные требования к техническим характеристикам каждого компьютера следующие:

- процессор – не ниже Celeron с тактовой частотой 2,8 ГГц;
- оперативная память – не менее 2Гб;
- жидкокристаллический монитор с диагональю не менее 15 дюймов;
- жёсткий диск – не менее 250 Гб;
- клавиатура;
- мышь;
- устройство для чтения компакт-дисков (желательно).

Электронные образовательные ресурсы

1. Курс Поколение Python: курс для начинающих на платформе Stepik – <https://stepik.org/course/58852/syllabus>. (для 7 класса)
2. Курс Поколение Python: курс для продвинутых на платформе Stepik – <https://stepik.org/course/68343/syllabus> . (для 8 класса)
3. Курс Поколение Python: курс для продвинутых на СДО лицея НГТУ (Moodle). (для 9 класса).

Используемые методические материалы

1. Типовая рабочая программа курса «Python для начинающих» - Белых Р., Гуев Т., Скобина О. (авторский коллектив благотворительного фонда развития образования «АйкьюОпшн», г. Санкт-Петербург 2022.
2. Типовая рабочая программа курса «Поколение Python: курс для продвинутых» - Белых Р., Гуев Т., Скобина О. (авторский коллектив благотворительного фонда развития образования «АйкьюОпшн» г. Санкт-Петербург 2022.
3. Методические указания для учителя - Белых Р., Гуев Т., Скобина О. (авторский коллектив благотворительного фонда развития образования «АйкьюОпшн» г. Санкт-Петербург 2023;
4. Презентации с иллюстративным изложением теоретического материала- Белых Р., Гуев Т., Скобина О. (авторский коллектив благотворительного фонда развития образования «АйкьюОпшн», г. Санкт-Петербург 2023.
5. Инструкции подвижных игр, направленных на закрепление знаний, полученных на занятии- Белых Р., Гуев Т., Скобина О. (авторский коллектив благотворительного фонда развития образования «АйкьюОпшн», г. Санкт-Петербург 2022.
6. Инструкции для проведения рефлексии процесса обучения с учениками- Белых Р., Гуев Т., Скобина О. (авторский коллектив благотворительного фонда развития образования «АйкьюОпшн», г. Санкт-Петербург 2022.
7. Авторские проектные задания для создания учащимися игр на языке Python - Белых Р., Гуев Т., Скобина О. (авторский коллектив благотворительного фонда развития образования «АйкьюОпшн», г. Санкт-Петербург 2022.

ПРОМЕЖУТОЧНЫЙ И ИТОГОВЫЙ КОНТРОЛЬ

Результаты освоения курса оцениваются по системе «зачет/незачет». Промежуточный и итоговый контроль проводится на онлайн платформе Stepik в форме защиты индивидуальных, групповых заданий по пройденному материалу и практических работ. Отметочная система контроля отсутствует.

Для допуска к четвертному зачету необходимо выполнить 70% предложенных заданий. Один раз в четверть учащиеся сдают зачет по изученным темам. Для получения зачета по курсу «Программирование на Python» необходимо выполнить 50% от объема предложенных заданий.

Один раз в год обучающиеся представляют групповые и индивидуальные проекты. Проводится обсуждение результатов в коллективе с опорой на чек-лист, исправление ошибок коррекция и закрепление полученных знаний.