



Муниципальное автономное общеобразовательное
учреждение Новосибирского района Новосибирской области -
лицей № 13 п. Краснообск

*Приложение к ООП СОО, утверждённой
приказом директора от 31.08.2026 № 62/2-
од
с внесёнными изменениями приказом
директора от 01.09.2026 № 2-од*

РАБОЧАЯ ПРОГРАММА
Курса «Программирование на Python»
для обучающихся 10 – 11 классов

Пояснительная записка

В настоящее время мы переживаем большие изменения в развитии современного общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Нами был выбран язык Python. Его синтаксис достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ОГЭ и ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

В основу курса «Программирование на Python» 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. Важным аспектом программы является самостоятельная работа над заданиями: школьники учатся решать задачи без помощи преподавателя. Для

этого в содержании курса фигурируют задания, в которых:

- для решения задачи необходимо найти какую-то информацию в сети Интернет;
- может потребоваться устранение ошибки, которую не так просто быстро обнаружить;
- условие сформулировано недостаточно прозрачно и ученику необходимо самостоятельно доформулировать его (или задать правильные вопросы преподавателю).

Программа «Программирование на Python» рассчитана на 136 учебных часов и 68 часов самостоятельной работы и предназначен для учеников 10 – 11 классов общеобразовательных школ.

Целью курса является создание условий для изучения методов программирования на языке программирования Python; рассмотрение различных

парадигм программирования, предлагаемых этим языком (процедурная, функциональная, объектно-ориентированная); подготовка к использованию как языка программирования, так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

Настоящий курс направлен на решение следующих задач:

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- знакомство с принципами и методами функционального программирования;
- знакомство с принципами и методами объектно-ориентированного программирования;
- приобретение навыков работы в интегрированной среде разработки на языке Python;
- изучение конструкций языка программирования Python;
- знакомство с основными структурами данных и типовыми методами обработки этих структур;
- приобретение навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;
- развитие у обучающихся интереса к программированию;
- формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- воспитание упорства в достижении результата;
- расширение кругозора обучающихся в области программирования;
- систематизации знаний по математическим основам информатики. По

окончании курса ученик приобретает следующие компетенции:

- знание основ современных языков программирования;
- умение объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- умение искать и обрабатывать ошибки в коде;
- умение разбивать решение задачи на подзадачи;
- способность писать грамотный, красивый код;
- способность анализировать как свой, так и чужой код;
- понимание основ представления, кодирования, хранения и передачи информации, логических законов построения компьютеров;
- способность работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);
- способность грамотно строить коммуникацию, исходя из целей и ситуации.

2. Новизна данной образовательной программы

Программа курса:

- обеспечивает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне;
- имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика;
- охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний;
- ориентирована на существующий парк вычислительной техники и дополнительные ограничения;
- допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального);
- предусматривает возможность индивидуальной работы с учащимися.

Практическая значимость курса заключается в том, что он способствует более успешному овладению знаниями и умениями по направлению «Программирование» через развитие самостоятельности обучающихся и оптимизацию средств и методов обучения.

Элементы программы курса могут быть рекомендованы для использования учителями информатики при проведении лабораторно-практических и практических занятий.

3. Общая характеристика курса «Программирование на Python»

3.1. Основные разделы программы

Модуль 1. Введение в программирование

Знакомство с электронной образовательной системой: регистрация, организация личного кабинета, поиск и выкладывание материалов. Знакомство с системой автоматизированной проверки задач.

Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки, интерпретатор, код программы и редактор кода.

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

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

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

Модуль 2. Базовые конструкции языка Python

Понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические

операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python.

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

Работа со списками, строками, множествами и кортежами в Python. Понятие итератора.

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

Модуль 3. Решение прикладных задач в Python

Понятие подпрограммы, процедуры, функции. Функции в Python.

Модули в Python. Подключение и использование модулей стандартной библиотеки. Модульный принцип компоновки программы. Работа с документацией в стандартной библиотеке. Понятие репозитория различных пакетов Python. Работа с внешними библиотеками Python и утилитой pip. Решение задач.

Модуль 4. Введение в ООП на языке Python

Понятие класса и объекта. Создание классов на языке Python. Атрибуты и экземпляры объектов. Методы и перегрузка операторов. «Магические» методы в Python. Наследование и множественное наследование. Проектирование классов.

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

Тематика самостоятельных и контрольных работ:

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

Выполненные самостоятельные работы загружаются в автоматизированную тестирующую систему для дальнейшей оценки.

3.2. Формы организации учебных занятий

Основной тип занятий — комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится учителем самостоятельно, но с учётом рекомендованного календарно-тематического плана. С учётом регулярного повторения ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема курса начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью учитель проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в классе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом ученики не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

В каждую тему курса интегрированы вопросы, связанные с теоретическими основами изучаемого раздела. Это позволяет учащимся получить более глубокое представление о принципах работы с информацией и способах автоматизации этой деятельности.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой

предусмотрены дополнительные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

Распределение учебной нагрузки

Для организации занятий по программе «Программирование на Python» необходимо выделить 1 двояенный урок (2 по 40 минут) в неделю.

Методы организации учебного процесса

При организации занятий по программе «Программирование на Python» для достижения поставленных целей и решения поставленных задач используются формы проведения занятий с активными методами обучения:

- занятие в форме проблемно-поисковой деятельности;
- занятие с использованием межпредметных связей;
- занятие в форме мозгового штурма;
- занятие в форме частично-поисковой деятельности.

Формы и методы контроля:

- тестирование;
- устный опрос;
- самостоятельные и контрольные работы;
- участие в проектной деятельности.

Общая характеристика учебного процесса:

- при изучении курса используются практические и самостоятельные работы;
- курс обучения заканчивается написанием программы для решения одной из задач.

4. Личностные, предметные и метапредметные результаты освоения курса

«Программирование на Python»

Личностные результаты:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно- исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;

- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Предметные результаты:

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;

- формирование представлений о математических объектах информатики и об их свойствах, умение оперировать с числами в различных системах счисления, измерять количество информации, представлять информацию в структурированном виде;
- формирование умения соблюдать нормы информационной этики и права.

Метапредметные результаты:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно- исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно- исследовательской деятельности.

5. Тематический план

План составлен из расчета 2-х часов в неделю/1 часа в неделю.

Учебный год содержит 34 недели

Тематическое планирование, 10 класс, 1 полугодие		
№	Содержание урока	Кол-во
1	<i>Зачисление в LMS. Резерв</i>	2
2	Знакомство с программой и LMS. Установка и настройка среды программирования. PEP8. Ввод- вывод. Переменные	2
3	Условный оператор. Отступы. Операции над строками	2
4	Сложные условия. Вложенные структуры.	2
5	Типы данных. Операции над числами	2
6	<i>Резерв</i>	2
7	Приоритет операций. Простейшие функции	2
8	Самостоятельная работа на линейные алгоритмы	2
9	Цикл с предусловием. Алгоритм Евклида	2
10	Отладка программ	2
11	Цикл for. Диапазоны	2
12	<i>Резерв</i>	2
13	Самостоятельная работа на циклы	2
11 4	Булевы переменные. Прерывания и продолжения циклов	2
15	Вложенные циклы	2
16	<i>Резерв</i>	2
17	Контрольная работа на циклы и линейные алгоритмы	2
Тематическое планирование, 10 класс, 2 полугодие		
1	<i>Резерв</i>	2
2	Элементы теории множеств. Множества в Python	2
3	Строки. Индексация	2
4	Строки. Срезы	2

5	Самостоятельная работа на строки и множества	2
6	<i>Резерв</i>	2
7	Знакомство со списками	2
8	Кортежи	2
9	<i>Резерв</i>	2
10	Списочные выражения. Методы split() и join()	2
11	Другие методы списков и строк	2
12	<i>Резерв</i>	2
13	Самостоятельная работа на методы строк и списков	2
14	Вложенные списки	2
15	Знакомство со словарями	2
16	Методы словарей	2
17	Контрольная работа на весь пройденный за год материал	2
18	<i>Резерв</i>	2

Тематическое планирование, 11 класс, 1 полугодие		
№	Содержание урока	Кол-во
1	<i>Резерв Зачисление в LMS</i>	2/1
2	Повторение	2/1
3	Функции. Введение	2/1
4	Функции. Возвращение значений	2/1
5	Возврат из глубины функции. Отладка	2/1

6	<i>Резерв</i>	2/1
7	Самостоятельная работа на функции	2/1
8	Области видимости переменных	2/1
9	Функции. Передача параметров	2/1
10	Функции с переменным числом аргументов	2/1
11	Самостоятельная работа на функции с параметрами	2/1
11	<i>Резерв</i>	2/1
13	Функции как объект. Лямбда-функции	2/1
14	Функциональная обработка коллекций. Поточковый ввод	2/1
15	<i>Резерв</i>	2/1
16	Контрольная работа на функции	2/1
17	<i>Резерв</i>	2/1
Тематическое планирование, 11 класс, 4 полугодие		
1	Повторение функций	2/1
2	Файловая система. Файлы в Python	2/1
3	Хранение данных. Работа с csv-документами.	2/1
4	Работа с файлами json	2/1
5	Самостоятельная работа на файлы	2/1
6	Библиотеки Python. Модули math, random, datetime	2/1
7	Библиотеки Python. Модуль PIL (рисование)	2/1
8	Библиотеки Python. Модуль PIL (трансформация)	2/1
9	Введение в ООП	2/1
10	Введение в ООП. Полиморфизм	2/1
11	Определение операторов	2/1
12	Определение операторов часть 2	2/1
13	Самостоятельная работа на ООП	2/1
14	Наследование	2/1
15	Наследование. Часть №2	2/1
16	<i>Резерв</i>	2/1

17	Контрольная работа на тему ООП	2/1
18	<i>Резерв</i>	2/1

6. Список литературы

Литература, использованная при подготовке программы

1. К. Ю. Поляков, Е. А. Еремин. Информатика. Углублённый уровень. Учебник для 8, 9, 10 классов. М.: БИНОМ. Лаборатория знаний, 2014 или более поздние редакции.
2. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.
3. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.
4. С. М. Окулов. Основы программирования. М.: Бином. Лаборатория знаний, 2012.

Литература, рекомендованная обучающимся

1. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.
2. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.

Ресурсы в интернете

1. Материалы и презентации к урокам в LMS Яндекс.Лицея.
2. Сайт pythonworld.ru — «Python 3 для начинающих».
3. Сайт pythontutor.ru — «Питонтьютор».