

Муниципальное общеобразовательное учреждение
«Гимназия № 12 Краснооктябрьского района Волгограда»

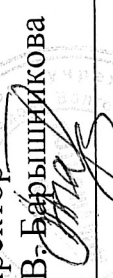
РАССМОТРЕНО:

на заседании МО
учителей математики,
информатики и ИКТ
Протокол № 1
от «28» августа 2025г.
Руководитель МО:
Т.И.Степина 

СОГЛАСОВАНО:

Заместитель директора
по учебно-воспитательной
работе
И.Д. Скрыбина 
«29» августа 2025г.

УТВЕРЖДЕНО:

Директор
Н.В. Барышийкина 
«29» августа 2025г.
Приказ № 266 от 29.08.2025 г.

РАБОЧАЯ ПРОГРАММА
учебного курса «Программирование на языке Python»
для обучающихся 10-х классов
на 2025/2026 учебный год

Составитель рабочей программы:
учитель информатики и ИКТ
Бухтиярова Е.Г.

Волгоград, 2025

РАБОЧАЯ ПРОГРАММА Дополнительного образования «Python»

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Актуальность и направленность программы. В настоящее время прослеживается острый кризис квалифицированных кадров в различных сферах рынка труда. Работодатели, хоть и обращают внимание на область предметных знаний, но все же отдают предпочтение универсальным навыкам. Подобная позиция связана с постоянно изменяющимися условиями труда, технологическим прогрессом. Любые универсальные навыки (способность работать в команде, многозадачность, творческий подход, критическое мышление, продуктивное мышление, нацеленность на результат) вырабатываются в процессе профессионального становления.

Характерной чертой развития общества на протяжении последних десятилетий является его все более расширяющаяся информатизация. Отражением и следствием этой тенденции явилась потребность в подготовке подрастающего поколения к вступлению в информационное общество, любая профессиональная деятельность в котором будет связана с информатикой и информационными технологиями. Изучение данного курса имеет важное значение для развития мышления подростков. В современной психологии отмечается значительное влияние изучения информатики и использования компьютеров в обучении на развитие у школьников теоретического, творческого мышления, а также формирование нового типа мышления, так называемого операционного мышления, направленного на выбор оптимальных решений; открывает новые возможности для овладения такими современными методами научного познания, как формализация, моделирование, компьютерный эксперимент и т.д. Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и некоторые общие мыслительные навыки, полезные и в более широком контексте.

Направленность дополнительной образовательной программы — техническая. Она заключается в развитии технического и творческого мышления у обучающихся через разработку алгоритмов в цифровой среде, взаимодействие с приложениями, сервисами и инструментами вне зависимости от платформ или интерфейса, формирование навыка определения различных видов ошибок (логическая, синтаксическая, ошибка среды и ошибка взаимодействия) и принципах их исправления в процессе разработки с помощью процедур отладки. Также развитие у детей умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, формирование навыков сотрудничества и коллективиста и работы в цифровой среде.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного мышления и навыков проектной деятельности. Программа составлена с учетом задач национального проекта «Цифровая экономика» (федеральный проект «Кадры для цифровой экономики») и национального проекта «Образование» (федеральный проект «Молодые профессионалы»): создание условий для формирования у учащихся основ цифровых компетенций, необходимых для дальнейшего участия в разработке и поддержке инфраструктуры передачи, обработки и хранения данных, а также цифровых продуктов для граждан, бизнеса и власти; разработка, внедрение цифровых технологий и платформенных решений.

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 веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователей, численное программирование и в других. ...

Программа «Программирование на Python» представляет собой углубленный курс по программированию, дающий представление о понятиях структурного программирования (данных, операций, переменных, ветвлениях в программах, циклах и функциях, массивах, файлах). Программа направлена на развитие логического и пространственного мышления обучающегося, способствует раскрытию творческого потенциала личности, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Наш выбор Python для преподавания призван помочь обучающимся сделать первые шаги по одному из современных и перспективных путей развития IT-индустрии.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для появления индигиуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

Данная программа относится к углубленному уровню, так как направлена на овладение знаниями и развитие навыков, она дает возможность ребенку поргизиться в атмосфере дополнителного образования, погробовать себя в новом лично значимом виде деятельности. Родителям дает возможность разобгаться с логикой дополнителного образования и наметить лично индигиуального развития своего ребенка.

Программа базируется на решении кейсов и проектной деятельности. Для повышения качества образования данные кейсы основываются на реальных ситуациях и проблемах потенциальных работодателей. Это дает возможность ранней профессиональной ориентации обучающегося.

По форме организации содержания и процесса педагогической деятельности программа «Программирование на Python» интегрированная, так как объединяет в целое отдельные образовательные области на основе единства математики и информатики. Главной целью является приобретение обучающимся теоретических знаний и практических навыков и компетенций, достаточных для дальнейшего успешного обучения в системе непрерывного образования: школа – дообразование – ВУЗ – предприятие. Сущностью программы можно определить как развитие мышления и формирование мировоззрения в условиях преподавания интегрированного курса информатики, математики, программирования обучающимся проникать в сущность изучаемых проблем. В теории интеграции в данном случае вкладывается понятие взаимосвязи, взаимопонимания математики и информатики, что предполагает качественное изменение в параметрах нового объекта. В школьном курсе информатики вопросы программирования рассматриваются лишь в ознакомительном плане и на это выделяется недостаточное количество часов, как следствие – формальное восприятие обучающимися основ современного программирования. Образовательная программа «Программирование на Python» направлена на устранение данного пробела.

Отличительные особенности программы и новизна. Отличительной особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной практической задачи, представляющую в виде адаптированного кейса из реальной жизни, составления алгоритма решения и его реализации с помощью средств программирования. Для этого, учащиеся проходят через следующие этапы: определяют и составляют последовательность действий, ведущую к цели, и записывают ее с помощью формального языка. Полученные знания учащиеся

применяют при создании собственных проектов, которые защищаются перед другими учениками, педагогами и родителями. Использование компьютерных технологий в работе с детьми среднего школьного возраста является стремительно развивающейся методикой в образовании во всем мире. С ее помощью можно более эффективно решать образовательные задачи, которые будут способствовать качественному улучшению обучения ребенка в школе.

Адресат программы. Программа ориентирована на дополнительное образование обучающихся 12-15 лет, интересующихся программированием.

Срок и этапы реализации программы. Программа рассчитана на 1 год обучения - 68 академических часа. Основной формой обучения являются групповые занятия. В основе образовательного процесса лежит проектный подход.

Режим занятий: 1 раз по 2 часа в неделю.
Продолжительность 1 занятия: 2 академических часа.

Структура двухчасового занятия:

- 40 минут – рабочая часть;
- 10 минут – перерыв (отдых);
- 40 минут – рабочая часть.

Основная форма работы теоретической части – лекционные занятия в группах до 15 человек. Практические задания планируются выполнять индивидуально, в парах и в малых группах. Занятия проводятся в виде бесед, семинаров, лекций: для наглядности изучаемого материала используется разный мультимедийный материал – презентации, видеоролики.

Цель программы

Целью курса – способствовать формированию алгоритмического и критического мышления, навыков проектной деятельности через использование языка программирования Python как инструмента для реализации идей в области своих интересов.

Задачи программы

Достижение поставленной цели складывается из выполнения следующих задач:

Обучающие:

1. Познакомить с базовыми понятиями основ программирования.
2. Познакомить с особенностями синтаксиса языка программирования Python.

3. Познакомить с принципами объектно-ориентированного программирования.
4. Сформировать навыки правильного оформления кода.
5. Сформировать навыки использования итеративного подхода при решении различных задач.
6. Формировать навыки решения задач и разработку проектов с помощью Python.
7. Формировать навыки разработки игр с помощью языка программирования Python.
8. Формировать навыки тестирования и оптимизации созданных проектов.

Развивающие:

1. Формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление.
2. Формировать навыки публичного выступления и презентации.
3. Формировать навыки планирования деятельности и основ тайм-менеджмента.
4. Формировать и расширять словарный запас на английском языке.
5. Формировать функциональную грамотность.

6. Расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.
7. Совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.

Воспитательные:

1. Воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развивать эмоциональный интеллект.
 2. Формировать умение давать ответственную обратную связь и реагировать на нее.
 3. Формировать информационную культуру.
- В результате освоения дисциплины обучающийся должен:

уметь:

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- настраивать рабочую среду Python;

знать:

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;

- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования;
- основы создания приложений в Python;
- инструментальные средства Python.

Методы: кейс-метод, проектная деятельность.

Формы работы:

- практическое занятие;
- занятие – соревнование;
- консультация;
- самостоятельная работа.

Практические занятия составляют важную часть теоретической и профессиональной подготовки. Они направлены на формирование практических навыков и умений. Практические занятия создают оптимальные условия для деятельности по освоению обучающимися содержания, методологии и методологии изучаемой дисциплины, использование специального оборудования, технических средств. Ведущей дидактической целью практических занятий является формирование практических умений – профессиональных (выполнять определённые действия, операции, необходимые в последующем в профессиональной деятельности).

Целью занятия-соревнования является создание условий для развития интеллекта и проявления способности к творчеству. Задачами такого занятия могут быть - повышение интереса к данному курсу в частности и познанию в целом, формирование навыков принятия необходимых решений. Занятие-соревнование отличается от «традиционного» тем, что он учит выполнению работы по предмету, использованию теории на практике, коллективной деятельности, делать выводы.

Консультации являются одной из форм руководства самостоятельной работой обучающихся и оказания им помощи в освоении учебного материала. Они могут проводиться регулярно или внепланово по мере необходимости. Консультации помогают обучающемуся избирать правильные методы работы, ни в коей мере не снижая его ответственности за принятые решения. Часто применяются при работе обучающихся над итоговым проектом.

Самостоятельная работа обучающихся – это разнообразные виды деятельности обучающихся, осуществляемые под руководством, но без непосредственного участия педагога в специальном отведённом для этого аудиторном или внеаудиторном время. Это особая форма обучения по заданиям педагога, выполнение которых требует активной мыслительной, поисково – исследовательской и аналитической деятельности. Методологическую основу самостоятельной работы обучающихся составляет деятельностный подход, когда цели обучения ориентированы на формирование умений решать типовые и нетиповые задачи, т. е. на реальные ситуации, где обучающимся надо проявить знание конкретной дисциплины, использовать внутрипредметные и межпредметные связи.

Виды учебной деятельности:

- анализ проблемных учебных ситуаций;
- построение гипотезы на основе анализа имеющихся данных;
- поиск необходимой информации;
- выполнение практических работ;
- подготовка выступлений и докладов с использованием разнообразных источников информации;
- публичное выступление.

Требования к результатам освоения программ:

Результат (освоенные компетенции)		Основные показатели оценки результата		Формы и методы контроля и оценки	
Личностные компетенции	Метапредметные компетенции	Знание основ ТРИЗ, навыки публичного выступления и презентации результатов, навык генерации идей	Знание основ и принципов программирования	Знание и понимание основных алгоритмических конструкций	- проектная деятельность, выполнение кейсов; - участие в конференциях, выставках,
Личностные компетенции	Метапредметные компетенции	Знание основ ТРИЗ, навыки публичного выступления и презентации результатов, навык генерации идей	Знание основ и принципов программирования	Знание и понимание основных алгоритмических конструкций	- проектная деятельность, выполнение кейсов; - участие в конференциях, выставках,
Личностные компетенции	Метапредметные компетенции	Знание основ ТРИЗ, навыки публичного выступления и презентации результатов, навык генерации идей	Знание основ и принципов программирования	Знание и понимание основных алгоритмических конструкций	- проектная деятельность, выполнение кейсов; - участие в конференциях, выставках,

конкурсах, соревнованиях и т.п.; - выполнение практических заданий	знанием основ и овладение практическими базисными знаниями программирования в Python	
	знанием основ и овладение практическими базисными навыками разработки игр в PyGame	

Формы подведения итогов реализации программы

Основной формой подведения итогов дополняющей программы «Язык программирования Python» является решение задач, проектная деятельность (разработка и отладка программ на языке программирования Python).
Критерии оценки защиты проекта:

Критерии	Ф.И.О. обучающегося, тема проекта		
Постановка цели, проблематизация:	1. Проектная работа соответствует цели и отвечает на проблемные вопросы – 3 балла 2. Проектная работа соответствует цели и отвечает на некоторые проблемные вопросы – 2 балла 3. Проектная работа не совсем точно отражает цель проекта и его проблемные вопросы – 1 балл		
Формулировка задач проекта:	1. Постановленные задачи ведут к достижению цели проекта – 3 балла 2. Не все задачи ведут к достижению цели проекта – 2 балла 3. Представленные задачи не ведут к достижению цели проекта – 1 балл		
Результаты работы:	1. Результаты работы, представленные при помощи компьютерных средств, оформлены в соответствии с правилами – 3 балла 2. Результаты работы, представленные при помощи компьютерных средств, содержат незначительные ошибки в оформлении – 2 балла 3. Результаты работы, представленные при помощи компьютерных средств, содержат значительные ошибки в оформлении – 1 балл		
Выступление:	1. Устное выступление участника логично, отсутствуют грамматические и лексические ошибки – 3 балла		

Год обучения	Год	34
Уровень	Вводный уровень	Кол-во часов

2.1. Объем программы

2. СТРУКТУРА ПРОГРАММЫ

2. Устное выступление участника логично, присутствуют незначительные грамматические и лексические ошибки, не мешающие пониманию материала – 2 балла	3. Устное выступление участника не всегда логично, присутствуют грамматические и лексические ошибки, которые затрудняют понимание – 1 балл	Соответствие выступления и презентации:	
		1. Выступление не повторяет текст презентации или публикации – 3 балла	2. Выступление частично повторяет текст презентации или публикации – 2 балла
3. Обучающийся затруднялся давать правильные ответы на вопросы – 1 балл	2. В ходе устного выступления даны ответы на некоторые вопросы – 2 балла	1. В ходе устного выступления даны ответы на все вопросы – 3 балла	Ответы на вопросы:
			3. Выступление полностью повторяет текст презентации или публикации – 1 балл

2.2. Учебный план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1.	Раздел 1. Введение в Python	2	1	1	Тест по модулю
	1.1. Инструкции и структура программы.	1	1	0	Устный опрос
	1.2. Установка Python. Ввод и вывод данных.	1	0	1	Решение задач
2.	Раздел 2. Типы данных и операции	3	1	2	Тест по модулю
	2.1. Перемennые. Операторы.	2	1	1	Тест по теме
	2.2 Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)	1	0	1	Решение задач
3.	Раздел 3. Инструкции и синтаксис	8	2	6	Тест по модулю
	3.1. Условный оператор if, Альтернативный условный оператор elif, else.	2	1	1	Решение задач
	3.2. Циклы while. Цикл for. Обработка вклочений. Функция random. Случайные числа.	6	1	5	Решение задач с использованием циклов
4.	Раздел 4. Функции и модули в программировании	10	2	8	Тест по модулю
	4.1. Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результаты.	4	1	3	Тест по теме
	4.2. Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк.	6	1	5	Решение задач с использованием строк
5.	Раздел 5. Сложные типы данных	6	1	5	Тест по модулю
	5.1. Списки. Срезы списков. Решение задач со списками	3	1	2	Решение задач
	5.2. Матрицы	1	0	1	Решение задач
	5.3. Словари	1	0	1	Решение задач
	5.4. Множества в языке Python	1	0	1	Решение задач

	Итого		34	7	27	
	Раздел 6. Написание и отладка программ.	Самостоятельная работа	5	0	5	Защита проекта

2.3. Содержание учебного плана

Раздел 1. Введение в Python

1.1. Инструкции и структура программ.

Техника безопасности. История создания. Области применения и перспективы. Инструкции и структура программ. Основы

Алгоритмизации. Достоинства языка.

1.2. Установка Python. Ввод и вывод данных.

Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы

Алгоритмизации.

Практическая работа: Решение задач для развития логики и понимания работы алгоритма. Установка программы Python. Написание

первой программы вывод на экран.

Раздел 2. Типы данных и операции

2.1.Переменные. Операторы.

Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций.

Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран.

Практическая работа: Решение задач на элементарные действия с числами. Создание программ простейших калькулятор.

2.2.Типы данных. Базовые функции для работы с различными типами данных.

Типы данных. Преобразование типов. Различные типов данных. Базовые функции (abs,round, int, math)

Практическая работа: Решение математических задач с использованием функции import math.

Раздел 3. Инструкции и синтаксис

3.1. Условные операторы if, elif, else.

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not).

Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация

ветвления в языке Python.

Практическая работа: Практическое закрепление знаний по условным операторам. Создание программ.

3.2. Циклы while, for. Обработка исключений. Случайные числа.

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randint. Функция random. Примеры решения задач с циклом.

Практическая работа: Числа Фибоначчи. Решение задачи с циклом for. Создание игры угадай число. Повторение пройденного.

Раздел 4. Функции и модули в программировании

4.1. Встроенные и пользовательские функции

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала.

Практическая работа: Создание игры русская рулетка. Создание всех созданных программ с использованием функций.

4.2. Строки - последовательности символов.

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строки. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа: Написание программ. Работа со строками.

Раздел 5. Сложные типы данных

5.1. Списки. Срезы списков.

Списки. Тип списка (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Практическая работа: Написание программ. Работа со списками.

5.2. Матрицы

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Установка Python. Доступ к документации. Ввод и вывод данных. Первая программа на языке Python.

5.3. Кортежи

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

5.4. Словари

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Практическая работа: Создание программ. Получение закрепленных знаний.

5.5. Множества в языке Python

Множества в языке Python. Множества. Тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа: Создание программ. Получение закрепленных знаний.

Раздел 6. Написание и отладка программ. Самостоятельная работа

Практическая работа: Стиль программирования. Отладка программ. Проектная работа по курсу «Программирование на языке Python»

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

3.1. Материально-техническое оснащение

Компьютерное оборудование:

– персональный компьютер – 15 шт.

Программное обеспечение:

– ОС Windows

– Python

– Текстовый редактор Блокнот

– Microsoft Power point

3.2. Методическое обеспечение реализации программы

Методическое обеспечение программы включает приёмы и методы организации образовательного процесса, дидактические материалы, техническое оснащение занятий.

Методы и приёмы организации образовательного процесса при реализации программы:

Словесные методы: объяснение, беседа, комментированное чтение, рассказ. Практические методы: работа с текстом, составление планов, работа над проектами, выполнение творческих заданий: составление кроссвордов, сочинение загадок, рассказов, выпуск бюллетеней, сборников или альбомов с творческими работами и проектами.

Игровые методы: фантазирование, театральная импровизация, живая наглядность.

Наглядные методы: показ видеоматериалов, посещение выставок, проведение экскурсий.

Виды дидактических материалов, используемые при реализации программы:

Для обеспечения наглядности и доступности учебного материала педагог использует наглядные пособия следующих видов:

• схематические или символические (оформленные стенды и планшеты, таблицы, схемы, рисунки, графики, плакаты, диаграммы, чертежи, шаблон и т.п.);

• картинные (иллюстрации, слайды, фотоматериалы и др.);

- звуковые (аудиозаписи);
 - смешанные (видеозаписи, учебные кинофильмы и т.д.);
 - дидактические пособия (карточки, рабочие тетради, раздаточный материал, вопросы и задания для опроса, тесты, практические задания, упражнения и др.);
 - компьютерные программы в электронном виде (компьютеры с программами, CD, флеш-носители);
 - учебные пособия, журналы, книги, Интернет-ресурсы.
- При реализации программ с целью повышения качества и эффективности процесса обучения применяются современные эффективные технологии обучения, ориентированные не на накопление знаний, а на организацию активной деятельности обучающихся:

- технологии проектной деятельности;

- компьютерные (информационные) технологии;

- технологии учебно-игровой деятельности (моделирование);

- технологии коммуникативно-диалоговой деятельности;

- модульные технологии;

- квест-технологии;

- технологично-ориентированного обучения;

- кейс-технологии.

Информационные технологии используются в различных видах деятельности:

• при подготовке и проведении занятий;

• для создания авторских мультимедийных презентаций;

• в рамках индивидуальной и групповой проектной деятельности;

• для самостоятельной работы;

• для накопления демонстрационных материалов к занятиям (видеоматериалы, таблицы, презентации, карты);

Одним из основных методов является метод проектного обучения, так как он является неотъемлемой частью учебного процесса. Исходный логичный основателей системы проектного обучения – «Все из жизни, все для жизни». Обучение строится на активной основе, через практическую деятельность ученика, ориентируясь на его личный интерес и практическую востребованность полученных знаний в дальнейшей жизни, обучающийся имеет возможность через проектную деятельность освоить получаемые знания. Проекты представляются в виде готовых программ, презентаций проектов, научных докладов, моделей, демонстраций видеofilьма. Достоинствами проектной деятельности являются:

- Уметь работать в коллективе;
- брать ответственность за выбор решения на себя;
- Разделять ответственность с другими;

- Предоставлять свободу выбора темы, методов работы;
- Понимание каждым обучающимся важности работы и др.

4. СПИСОК ЛИТЕРАТУРЫ

4.1. Список литературы, используемой педагогом

Основная:

1. Клейнберг Дж. Алгоритмы: разработка и применение. СПб: Питер, 2019. - 800 с.

Дополнительная:

1. Лунц М. Изучаем Python. СПб: Симво-пресс, 2021. - 1280 с.
2. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. М: Ямб, 2021. - 520 с.
4. Златопольский Д.М. Сборник задач по программированию. СПб: БХВ-Литербур, 2020. - 295 с.
5. ООП на Python: концепции, принципы и примеры реализации [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/python-ooop/>

4.2. Список рекомендуемой литературы для обучающихся

Основная:

1. Свейгар.Эл. Учим python, делая крутые игры М: Эксмо, 2021. – 416 с.
2. Семкин И.Г. Основы алгоритмизации и программирования. М.: Академия, 2019. - 304 с.

Дополнительная:

6. Лунц М. Изучаем Python. СПб: Симво-пресс, 2019. - 1280 с.
7. Уроки по Python для начинающих [Электронный ресурс]. – Режим доступа: <https://pythonu.com/wroki/vvedenie-wroki-po-python-dlja-nachinalushih>
8. Алгоритмизация. Программирование Python 3 [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/7215/promo>

