

Министерство образования Ярославской области  
Государственное профессиональное образовательное  
автономное учреждение Ярославской области  
Ростовский колледж отраслевых технологий

Центр цифрового образования детей «IT-куб»

УТВЕРЖДАЮ:

Директор ГПОАУ ЯО

Ростовский колледж

отраслевых технологий

Т.Н. Кудрявцева



**Дополнительная общеобразовательная  
общеразвивающая программа технической  
направленности**

**«Программирование на Python»**

Направленность: техническая

Уровень программы: стартовый

Срок реализации: 1 год (72 часа)

Возраст обучающихся : 12-15 лет

2024 год

## СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	3
УЧЕБНЫЙ ПЛАН .....	14
СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА .....	16
КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ .....	16
СПИСОК ЛИТЕРАТУРЫ .....	23
Приложение 1 .....	24

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### 1. Нормативно-правовая база разработки и реализации программы.

Программа разработана на основе следующих нормативных документов:

- Федерального закона от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020);
- Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденного приказом Министерства просвещения Российской Федерации от 27.07.2022 г. № 629;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
- Приказа Минобрнауки России №882, Минпросвещения России №391 от 05.08.2020 «Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ» (вместе с «Порядком организации и осуществления образовательной деятельности при сетевой форме реализации образовательных программ»);
- Методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» (письмо Министерства просвещения Российской Федерации от 10 ноября 2021 г. № ТВ-1984/04);
- Стратегии развития воспитания в Российской Федерации на период до 2025 года (Утверждена распоряжением Правительства РФ от 29.05.2015 № 996-

р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»);

– Концепции развития дополнительного образования детей до 2030 года, утвержденной распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р.

## **2. Новизна, актуальность, педагогическая целесообразность программы.**

В настоящее время мы переживаем большие изменения в развитии общества. Информационные технологии стали неотъемлемой частью нашей жизни. Большое значение приобретает умение работать с компьютером, притом на уровне профессионала, а не обывателя.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. В данной программе выбран язык программирования Python. Выбор обусловлен синтаксисом языка, который является достаточно простым и интуитивно понятным, что понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При это 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.

Данная программа составлена с учётом современных потребностей рынка в специалистах в области информационных технологий, особенно в области программирования. Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Знания и умения, приобретенные в результате освоения курса, могут быть использованы обучающимися при сдаче 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.

**Отличительные особенности программы.** Особенность программы «Программирование на Python» заключается в изучении основ программирования на языке Python, основных приёмов написания программ на современном языке программирования, развитие алгоритмического мышления учащихся, творческих способностей, аналитических и логических компетенций.

**Направленность программы:** техническая.

### **3. Адресат программы.**

Адресатом программы являются дети в возрасте от 12 до 15 лет.

Содержание и объем стартовых знаний, необходимых для начального этапа освоения программы: понимание основных алгоритмических структур.

### **4. Срок реализации программы.**

Срок реализации программы составляет 1 год.

**Уровень программы:** стартовый.

Программа предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка, гарантированно обеспечивают трансляцию общей и целостной картины в рамках содержательно-тематического направления программы.

### **5. Форма реализации программы.**

Форма обучения – очная.

Образовательные технологии: информационные технологии, проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

Форма организации содержания и процесса педагогической деятельности – комплексная.

Тип организации работы учеников: групповая работа, индивидуальная, коллективная.

Виды занятий: лекции и практические занятия.

**Наполняемость группы:** от 10 до 12 человек.

## **6. Объём программы и режим работы.**

**Объём программы:** 72 часа.

**Режим занятий:** 2 - 3 раза в неделю по 2 академических часа.  
Продолжительность часа – 40 минут.

## **7. Цель программы:**

Целью программы является создание условий для изучения методов программирования на языке Python; подготовка к использованию языка программирования и методов программирования на Python в учебной и последующей деятельности в различных предметных областях.

## **8. Задачи программы.**

Программа направлена на решение следующих задач:

### **Обучающие:**

- познакомить с правилами безопасного использования цифровыми инструментами и компьютерным оборудованием, организации рабочего места;
- формирование навыков работы в интегрированной среде разработки на языке Python;

- изучение конструкций языка программирования Python;
- формирование навыков разработки эффективных алгоритмов и программ на языке программирования Python;

#### **Развивающие:**

- развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- развитие навыков поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- способствовать формированию познавательных и регулятивных универсальных учебных действий;
- формирование творческого подхода к поставленной задаче;
- развитие навыков эффективной деятельности в проекте;
- формирование навыков рефлексивной деятельности.

#### **Воспитательные:**

- формирование стремления к получению качественного законченного результата в проектной деятельности;
- воспитание социально-значимых качеств личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

### **9. Планируемые результаты освоения программы.**

В результате освоения программы, учащиеся будут:

#### **Знать:**

- правила работы с компьютером и технику безопасности;
- основы языка Python;
- особенности работы с интегрированной средой разработки;

- базовые и сложные конструкции, способы организации процедур и функций в языке программирования Python;

**Уметь:**

- работать в интегрированной среде разработки на языке программирования Python;
- использовать специальные средства и библиотеки языка Python;

**Владеть:**

- навыками разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;

**Метапредметные результаты:**

- навыками алгоритмического и логического мышления, грамотной разработки программ;
- развитие навыков поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- формирование познавательных и регулятивных универсальных учебных действий;
- формирование творческого подхода к решению поставленной задачи.

**Личностные результаты:**

- формирование навыков рефлексивной деятельности;
- формирование стремления к получению качественного законченного результата в проектной деятельности;
- воспитание социально-значимых качеств личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

**10. Формы представления результатов.**



Формы аттестации: беседа, демонстрация решения, наблюдение, защита проектов.

Входной контроль осуществляется в начале реализации программы в форме беседы и наблюдения и имеет диагностические задачи. Цель входной диагностики – зафиксировать начальный уровень подготовки обучающихся, имеющиеся знания, умения и навыки, связанные с предстоящей деятельностью.

Наблюдение осуществляется в течение реализации программы.

Промежуточный контроль осуществляется в целях диагностики теоретических знаний и практических умений и навыков по итогам освоения одного из разделов курса. Проводится в форме тестирования.

Итоговый контроль проводится с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения – представляет из себя защиту проекта.

## **11. Оценочные материалы, формирующие систему оценивания.**

**Промежуточный контроль** осуществляется в целях диагностики теоретических знаний и практических умений и навыков по итогам освоения одного из разделов курса. Проводится в форме тестирования.

Максимальный балл за промежуточное тестирование: 55 баллов.

Оценивание тестирования осуществляется по следующим уровням:  
высокий уровень – учащийся набрал не менее 80% от максимально возможного количества баллов (от 44 балла).

средний уровень – учащийся набрал не менее 50% от максимально возможного количества баллов (от 27 баллов).

низкий уровень – учащийся набрал менее 50% от максимально возможного количества баллов (менее 27 баллов).

**Итоговый контроль** представляет из себя защиту проекта. Проект представляет собой программную реализацию на языке программирования Python, по предложенной теме. В проекте реализуются основные функции автоматизирующие процессы предметной области и направленные на оптимизацию рабочих процессов.

### Критерии оценки проекта

№	Название критерия	Максимальный балл
1.	Актуальность и проработанность проблемы	До 5 баллов
2.	Четкость формулировки целей и задач	До 5 баллов
3.	Технологическая сложность проекта. Наличие нестандартных структур данных и алгоритмов, специфика которых направлена на решение конкретной задачи.	До 10 баллов
4.	Новизна и оригинальность решения	До 5 баллов
5.	Качество разработанного продукта. Наличие удобного и понятного интерфейса, как абстракции между реализацией и пользователем.	До 5 баллов
6.	Защита проекта: <ul style="list-style-type: none"> <li>• качество презентации;</li> <li>• четкость и ясность изложения, умение взаимодействовать с аудиторией, отвечать на вопросы</li> </ul>	До 5 баллов
7.	Наличие самооценки и перспектив дальнейшей разработки проекта	До 5 баллов
Итого		40 баллов

Оценивание проекта осуществляется по следующим уровням:

Высокий уровень – учащийся набрал не менее 32 баллов по итогам защиты проекта.

Средний уровень – учащийся набрал от 20 до 31 балла по итогам защиты проекта.

Низкий уровень – учащийся набрал менее 20 баллов по итогам защиты проекта.

Общими критериями оценки результативности обучения являются:

- оценка уровня теоретических знаний: свобода восприятия теоретической информации, осмысленность и свобода использования специальной терминологии, свобода ориентации в теоретическом материале;

- оценка уровня практической подготовки: соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением;

- оценка уровня достижения личностных результатов: культура организации самостоятельной деятельности, культура работы с информацией, аккуратность и ответственность при работе.

**Оценка итоговых результатов** освоения программы осуществляется по трем уровням:

Высокий уровень – достижение 80- 100% показателей освоения программы.

Средний уровень – достижение 50- 79% показателей освоения программы.

Низкий уровень - достижение менее чем 50% показателей освоения программы.

Достигнутые обучающимся знания, умения и навыки заносятся в сводную таблицу результатов обучения.

### Оценка уровней освоения программы

Уровни	Параметры	Показатели
<b>Высокий уровень (80-100%)</b>	Теоретические знания	Обучающийся освоил материал в полном объеме. Знает и понимает значение терминов, употребляет их осознанно и в полном соответствии с содержанием. Самостоятельно ориентируется в содержании материала по темам.
	Практические умения и навыки	Обучающийся овладел 80-100% умений и навыков, предусмотренных программой за конкретный период. Умет работать самостоятельно, применяя практические умения и навыки. Правильно и по назначению применяет инструменты. Умеет выполнять основные логические действия (анализ, синтез, установление причинно-следственных связей). Способен планировать и регулировать свою деятельность по реализации проекта.

		Умеет осуществлять поиск информации, в том числе в сети Интернет; выслушивать собеседника и вести диалог; выражать свои мысли в соответствии с задачами и условиями коммуникации.
	Личностные результаты	Обучающийся обладает внутренней мотивацией. Способен самостоятельно организовать собственную деятельность. Сформирована культура работы с информацией. Работу выполняет аккуратно, доводит до конца. Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.
<b>Средний уровень (50-79%)</b>	Теоретические знания	Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу. Использует специальную терминологию, однако сочетает её с бытовой
	Практические умения и навыки	Владеет базовыми навыками и умениями, но не всегда может выполнить задание самостоятельно, просит помощи педагога. В основном выполняет задания на основе образца. Способен разработать проект с помощью преподавателя. Встречаются отдельные случаи неправильного применения инструментов. Делает ошибки в работе, но может устранить их после наводящих вопросов или самостоятельно Испытывает незначительные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей). Способен планировать и регулировать свою деятельность по реализации проекта с помощью педагога. Испытывает незначительные сложности в осуществлении коммуникации с педагогом и сверстниками.
	Личностные результаты	Внутренняя мотивация к обучению сочетается с внешней. В работе допускает небрежность. Работу не всегда выполняет аккуратно и/или доводит до конца. Оценить результаты своей деятельности может с подсказкой педагога.
<b>Низкий уровень (меньше 50%)</b>	Теоретические знания	Владеет минимальными знаниями, ориентируется в содержании материала по темам только с помощью педагога. Избегает употреблять специальные термины.
	Практические умения и навыки	Владеет минимальными начальными навыками и умениями. Учащийся способен выполнять каждую операцию только с подсказкой педагога или товарищей. Часто неправильно применяет необходимый инструмент или не использует его вовсе. В работе допускает грубые ошибки, не может их найти даже после указания преподавателя. В состоянии выполнять лишь простейшие практические задания педагога.

		<p>Испытывает существенные затруднения при выполнении основных логических действий (анализ, синтез, установление причинно-следственных связей).</p> <p>Не способен планировать и регулировать свою деятельность по реализации проекта.</p> <p>Испытывает значительные сложности в осуществлении коммуникации с педагогом и сверстниками.</p>
	Личностные результаты	<p>Преобладает внешняя мотивация к обучению.</p> <p>Работу часто выполняет неаккуратно и/или не доводит до конца.</p> <p>Не способен самостоятельно и объективно оценить результаты своей работы.</p>

### **Сводная таблица результатов обучения по программе**

№ п/п	Фамилия, имя обучающегося	Оценка теоретических знаний	Оценка практических умений и навыков (предметных и метапредметных)	Личностные результаты	Итоговая оценка
1.					
2.					
3.					

# УЧЕБНЫЙ ПЛАН

## Учебно-тематический план

№ п/п	Название разделов, тем	Количество часов			Форма аттестации/ контроля
		Всего	Теория	Практика	
<b>1</b>	<b>Введение в программирование</b>	<b>10</b>	<b>4</b>	<b>6</b>	
1.1	Вводное занятие. Инструктаж по ТБ. Основы алгоритмизации.	2	2	-	Опрос
1.2	Знакомство со средой PyCharm. Организация вывода данных в программе. Функция Print.	2	1	1	Опрос
1.3	Организация ввода данных в программе. Функция Input.	2	1	1	Наблюдение
1.4	Создание программ линейной структуры	4	-	4	Демонстрация решений
<b>2</b>	<b>Базовые конструкции языка Python</b>	<b>40</b>	<b>12</b>	<b>28</b>	
2.1	Условный оператор IF. Создание программ разветвляющейся структуры	4	1	3	Наблюдение
2.2	Операторы циклов while и for.	4	2	2	Опрос, наблюдение
2.3	Создание программ циклической и разветвляющейся структуры	4	-	4	Демонстрация решения
2.4	Строки. Операции со строками. Методы split и join.	4	1	3	Наблюдение
2.5	Списки. Индексы элементов списка. Ввод/вывод элементов списка. Методы для работы со списком.	4	1	3	Наблюдение
2.6	Работа со списком: поиск, замена, перестановка элементов. Сортировка элементов списка	4	1	3	Наблюдение
2.7	Понятие массива. Индексы элементов массива. Ввод/вывод элементов массива.	4	2	2	Демонстрация решения
2.8	Работа с массивом (вектором, матрицей)	4	-	4	Наблюдение

2.9	Кортежи. Множества. Словари.	4	2	2	Наблюдение
2.10	Понятие подпрограммы. Создание процедур и функций.	4	2	2	Тест
<b>3</b>	<b>Создание графического интерфейса</b>	<b>22</b>	<b>2</b>	<b>20</b>	
3.1	Кроссплатформенная библиотека для разработки графического интерфейса TKInter	4	2	2	Наблюдение
3.2	Создание приложений с графическим интерфейсом	6	-	6	Демонстрация решения
3.3	Выбор темы, планирование работы над проектом, изучение предметной области, проектирование	2	-	2	Наблюдение
3.4	Разработка итогового проекта	6	-	6	Наблюдение
3.5	Подготовка к защите проектов	2	-	2	Беседа
3.6	Защита итоговых проектов	2	-	2	Демонстрация проектов
	<b>Итого</b>	<b>72</b>	<b>18</b>	<b>54</b>	

## СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА

### **Раздел 1. Введение в программирование.**

#### **Тема 1.1** Вводное занятие. Теория - 2 часа.

Теория. Инструктаж по ТБ. Основы алгоритмизации. Понятие алгоритма, свойства алгоритма (дискретность, понятность, результативность, массовость, определенность), понятие исполнителя алгоритма. Различные формы представления алгоритмов, правила оформления блок-схем алгоритмов.

**Тема 1.2** Знакомство со средой PyCharm. Организация вывода данных в программе. Функция Print. Теория - 1 час, практика - 1 час.

Теория. Знакомство со средой PyCharm. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода. Переменные. Константы. Арифметические операторы в Python. Функция Print. Рассмотрение примеров.

Практика. Реализация простейших программы с выводом на экран.

**Тема 1.3** Организация ввода данных в программе. Функция Input. Теория - 1 час, практика - 1 час.

Теория. Организация ввода данных в программе. Функция Input. Создание программ линейной структуры. Рассмотрение примеров.

Практика. Программирование алгоритмов линейной структуры.

**Тема 1.4** Создание программ линейной структуры. Практика - 4 часа.

Практика. Разработка программ линейной структуры.

### **Раздел 2. Базовые конструкции языка Python**

**Тема 2.1** Условный оператор IF. Создание программ разветвляющейся структуры. Теория - 1 час, практика - 3 часа.

Теория. Условный оператор IF. Создание программ разветвляющейся структуры. Рассмотрение примеров.

Практика. Программирование алгоритмов разветвляющейся структуры.

**Тема 2.2** Операторы циклов While и For. Теория - 2 часа, практика - 2 часа.



Теория. Операторы циклов While и For. Создание программ циклической структуры. Рассмотрение примеров.

Практика. Программирование алгоритмов циклической структуры.

**Тема 2.3** Создание программ циклической и разветвляющейся структуры.

Практика - 4 часа.

Практика. Решение задач с использованием операторов ветвлений и циклов.

**Тема 2.4** Строки. Операции со строками. Методы split и join. Теория - 1 час, практика - 3 часа.

Теория. Строки. Операции со строками. Методы split и join

Практика. Написание программ на тему «Работа со строками».

**Тема 2.5** Списки. Индексы элементов списка. Ввод/вывод элементов списка. Методы для работы со списком. Теория - 1 час, практика - 3 часа.

Теория. Списки. Индексы элементов списка. Ввод/вывод элементов списка. Методы для работы со списком.

Практика. Решение задач на ввод, вывод элементов списка.

**Тема 2.6** Работа со списком: поиск, замена, перестановка элементов. Сортировка элементов списка. Теория - 1 час, практика - 3 часа.

Теория. Работа со списком: поиск, замена, перестановка элементов. Сортировка элементов списка.

Практика. Решение задач на обработку списков.

**Тема 2.7** Понятие массива. Индексы элементов массива. Ввод/вывод элементов массива. Теория - 2 часа, практика - 2 часа.

Теория. Понятие массива. Индексы элементов массива. Ввод/вывод элементов массива. Рассмотрение примеров работы с массивом.

Практика. Решение задач на ввод, вывод элементов списка.

**Тема 2.8** Работа с массивом (вектором, матрицей). Практика - 4 часа.

Практика. Работа с массивом (вектором, матрицей): поиск, замена, перестановка элементов. Формирование массива из элементов другого массива.

**Тема 2.9** Кортежи. Множества. Словари. Теория - 2 часа, практика - 2 часа.

Теория. Кортежи. Словари (dict) и работа с ними. Методы словарей. Множества (set и frozenset).

Практика. Решение задач на тему «Словари и множества».

**Тема 2.10.** Понятие подпрограммы. Создание процедур и функций. Теория - 2 часа, практика - 2 часа.

Теория. Понятие подпрограммы. Процедуры и функции. Правила составления и вызова процедур и функций в программном коде. Локальные и глобальные переменные.

Практика. Создание процедур и функций.

### **Раздел 3. Создание графического интерфейса.**

**Тема 3.1** Кроссплатформенная библиотека для разработки графического интерфейса TKInter. Теория - 2 часа, практика - 2 часа.

Теория. Кроссплатформенная библиотека для разработки графического интерфейса TKInter.

Практика. Решение задач на тему «Модуль TKInter».

**Тема 3.2** Создание приложений с графическим интерфейсом. Практика - 6 часов.

Практика. Разработка приложений с графическим интерфейсом.

**Тема 3.3** Выбор темы, планирование работы над проектом, изучение предметной области, проектирование. Практика - 2 часа.

Практика. Выбор темы, планирование работы над проектом, изучение предметной области, проектирование

**Тема 3.4** Разработка итогового проекта. Практика - 6 часов.

Практика. Создание программы для итогового проекта.

**Тема 3.5** Подготовка к защите проектов. Практика - 2 часа.

Практика. Обсуждение плана выступления защиты индивидуального проекта.

**Тема 3.6** Защита индивидуальных итоговых проектов. Практика - 2 часа.

Практика. Защита индивидуальных итоговых проектов.

# КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

## Материально-техническое обеспечение

Комплекс условий реализации программы:

Аппаратное и техническое обеспечение:

а) Рабочее место учащегося

- ноутбук с выходом в сеть Интернет;

б) Рабочее место наставника

- ноутбук с выходом в сеть Интернет;

- технические средства обучения (мультимедийное устройство).

## Методическое обеспечение программы

Методы обучения:

- объяснительно-иллюстративный (беседы, объяснения);

- репродуктивный (деятельность обучаемых носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях);

- метод проблемного изложения;

- эвристический (метод обучения заключается в организации активного поиска решения выдвинутых в обучении (или самостоятельно сформулированных) познавательных задач в ходе подготовки и реализации творческих проектов);

- исследовательский.

В процессе обучения учащиеся работают со средами программирования, поддерживающими язык Python, текстовыми редакторами, тренажерами быстрого набора текста, табличными редакторами.

Педагогические технологии: информационные технологии, проектная технология, здоровьесберегающие технологии, технология проблемного обучения.

Проектная технология дает возможность самостоятельно конструировать свои знания, ориентироваться в информационном пространстве, развивает критическое и творческое мышление, создаёт условия для формирования и развития внутренней мотивации учащихся к более качественному овладению знаниями, повышения мыслительной активности и приобретения навыков логического мышления.

Здоровьесберегающие технологии позволяют создать максимально возможные условия для сохранения, укрепления и развития эмоционального, интеллектуального и физического здоровья, в том числе в условиях работы с компьютерной техникой.

Проблемное обучение — это тип развивающего обучения, содержание которого представлено системой проблемных задач различного уровня сложности, в процессе решения которых учащиеся овладевают новыми знаниями и способами действия, а через это происходит формирование творческих способностей: продуктивного мышления и познавательной мотивации.

### **Учебно-методические средства обучения**

- специализированная литература по направлению, подборка журналов,
- образцы программ и систем, выполненные обучающимися и педагогом,
- учебно-методические пособия для педагога и обучающихся, включающие дидактический, информационный, справочный материалы на различных носителях, компьютерное и видео оборудование.

Применяемое на занятиях дидактическое и учебно-методическое обеспечение включает в себя электронные учебники, справочные материалы и системы используемых Программ, Интернет, рабочие тетради обучающихся.

### **Кадровое обеспечение**

Программу реализуют педагоги структурного подразделения Центр цифрового образования детей «IT-куб».

### **Информационное обеспечение**

Для реализации программы планируется использование следующих информационных ресурсов:

- курс «Программирование на Python» (<https://stepik.org/course/67>);
- курс «Алгоритмы: теория и практика. Методы» (<https://stepik.org/course/217>).

## СПИСОК ЛИТЕРАТУРЫ

### Список литературы для педагогов

1. Бэрри П. Изучаем программирование на Python. — М., 2017. — 624 с.
2. Лутц М. Изучаем Python, пер. с англ. 3-е изд. — СПб.: Символ Плюс, 2009. — 848 с.
3. Мэтиз Э. Изучаем Python. 3-издание. — СПб.: Питер, 2021 – 511 с.
4. Стивенс Р. Алгоритмы. Теория и практическое применение. – М.: Эксмо, 2022 – 547 с.
5. Фёдоров Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата. — М.: Издательство Юрайт, 2019. — 161 с.

### Список литературы для учащихся

1. Буйначев С. К. Основы программирования на языке Python: учебное пособие. — Екатеринбург: Изд-во Урал. ун-та, 2014. — 91 с.
2. Бхаргава А. Грожаем алгоритмы: иллюстрированное пособие для программистов и любопытствующих. — СПб.: Питер, 2017. — 288 с.
3. Гэддис Т. Начинаем программировать на Python / пер. с англ. 4-е изд. — СПб.: БХВ-Петербург, 2019. — 768 с.
4. Луридас П. Алгоритмы для начинающих: теория и практика для разработчика. — М.: Эксмо, 2018. — 608 с.
5. Мюллер Дж. Python для чайников. — СПб.: Диалектика, 2019. — 416 с.

### Примерные задания для итоговой аттестации

1. Веб-страница с расписанием занятий и темами ЦЦОД «It-Куб».
2. Сайт-агрегатор.
3. Система шифрования и архивирования файлов.
4. Система рейтинга обучающихся ЦЦОД «It-Куб».
5. Модуль проверки личной информации абитуриентов ЦЦОД «It-Куб».
6. Чат-бот, собирающий информацию о интересующих темах для обучения ЦЦОД «It-Куб».
7. Интернет-магазин с возможностью заказа из чат-бота.
8. Парсер.
9. Веб-сайт по ведению проекта.
10. Сайт-портфолио обучающегося ЦЦОД «It-Куб».