



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ «БИСТАДИ»
(ООО БИСТАДИ)

Утверждаю
Директор ООО «БИСТАДИ»

Б.А. Сафонов

ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

"ОСНОВЫ ПРОГРАММИРОВАНИЯ НА PYTHON"

Форма обучения: с применением исключительно
электронного обучения, дистанционных образовательных технологий

Трудоемкость: 25 часа
Срок обучения: 6 недель

г. Москва, 2026 г.

СОДЕРЖАНИЕ

РАЗДЕЛ 1. ОБЩАЯ ХАРАКТЕРИСТИКА ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ.....	3
1.1. Нормативно-правовые основания разработки программы	Ошибка! Закладка не определена.
1.2. Цель и планируемые результаты обучения.....	3
1.3. Категория слушателей	3
1.4. Форма обучения.....	4
1.5. Трудоемкость программы.....	4
1.6. Особенности организации образовательной деятельности по программе	4
1.7. Содержание разделов.....	4
РАЗДЕЛ 2. СОДЕРЖАНИЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	5
2.1. Учебный план	5
2.2. Календарный учебный график	6
2.3. Рабочие программа.....	7
РАЗДЕЛ 3. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ПРОГРАММЕ	7
3.1. Формы аттестации	7
3.2. Контрольно-оценочные средства для текущего и итогового контроля	7
РАЗДЕЛ 4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ПРОФЕССИОНАЛЬНОЙ ПРОГРАММЫ	9
4.1. Требования к материально-техническому обеспечению.....	9
4.2. Информационно-методическое обеспечение обучения.....	10
4.3. Кадровое обеспечение реализации программы.....	11

РАЗДЕЛ 1. ОБЩАЯ ХАРАКТЕРИСТИКА ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

1.1. Нормативно-правовые основания разработки программы

Разработка программы осуществлена в соответствии с действующими нормативными правовыми актами Российской Федерации, регулирующими реализацию дополнительных образовательных программ:

- 1. Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» (в действующей редакции);**
- 2. Постановление Правительства Российской Федерации от 18 сентября 2020 г. № 1490 «О лицензировании образовательной деятельности»;**
- 3. Приказ Минпросвещения РФ от 27.07.2022 N 629 «Об утверждении [порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам](#)»;**
- 4. Устав ООО «БИСТАДИ».**

Цель: формирование базовых навыков программирования на языке Python: от установки среды разработки и освоения синтаксиса до уверенной работы с типами данных, коллекциями, функциями, итераторами и объектно-ориентированным программированием с выполнением итогового практического проекта.

В ходе обучения формируются знания и практические умения по написанию программ на Python: работе с числами, строками и датами, использованию коллекций, построению условий и циклов, обработке исключений, созданию функций и декораторов, применению генераторов и включений, а также проектированию собственных классов.

Планируемые результаты обучения:

Будут знать:

- назначение языка Python и его роль в современных цифровых профессиях;
- принципы организации среды разработки и работы с Python;
- основные типы данных, коллекции и базовые структуры языка;
- способы управления логикой программ с помощью условий, циклов и обработки исключений;
- принципы создания функций, модульного кода и повторного использования программных компонентов;
- основы работы с файлами, стандартными библиотеками и модулями Python;
- принципы объектно-ориентированного программирования, итераторов и генераторов;
- основные подходы к разработке, тестированию и структурированию программ на Python.

Будут уметь:

- настраивать рабочую среду и подготавливать Python к работе;
- писать понятный, структурированный и корректно оформленный код;
- работать с основными типами данных и преобразовывать их под задачу;

- выбирать подходящие коллекции для хранения и обработки информации;
- создавать логику программ с помощью условий и управляющих конструкций;
- обрабатывать ошибки и обеспечивать устойчивую работу программ;
- разрабатывать функции с разными типами параметров и аргументов;
- использовать замыкания и декораторы для расширения возможностей кода;
- применять итераторы, генераторы и comprehensions для эффективной обработки данных;
- создавать собственные классы и использовать основные принципы объектно-ориентированного программирования.

Будут владеть:

- практическими навыками разработки программ на Python;
- инструментами работы в IDE и интерактивных средах;
- методами обработки данных с помощью коллекций;
- приёмами функционального программирования;
- способами обработки исключений и создания устойчивого кода;
- навыками применения итераторов и генераторов;
- основами объектно-ориентированного проектирования;
- опытом создания и презентации завершённого итогового проекта.

1.3. Категория слушателей

Курс предназначен для лиц, заинтересованных в освоении основ программирования на языке Python. Программа подойдет студентам, начинающим специалистам, офисным сотрудникам, аналитикам, маркетологам, тестировщикам, предпринимателям, преподавателям и всем слушателям, которые хотят получить первый практический опыт программирования, автоматизировать рабочие задачи или подготовиться к дальнейшему обучению в области разработки, анализа данных и машинного обучения.

Курс ориентирован на слушателей без профильного технического образования и не требует предварительного опыта программирования. Обучение построено на практических примерах и задачах: от первых строк кода и работы с типами данных до создания собственных функций, классов и законченного итогового проекта.

Требования к поступающим:

Для зачисления на курс специальные знания в области программирования не требуются. Достаточно базовой компьютерной грамотности, уверенного владения персональным компьютером, навыков работы с файлами и документами, а также мотивации к освоению языка Python для решения практических задач.

1.4. Форма обучения

Заочная, с применением исключительно электронного обучения и дистанционных образовательных технологий (ЭО и ДОТ). Освоение программы происходит полностью удаленно на образовательной платформе.

1.5. Трудоемкость программы

Общий объем программы: 25 академических часа.

Продолжительность: 6 недель.

Указанные сроки освоения учебного материала носят рекомендательный характер и рассчитаны на комфортное и последовательное прохождение программы. При освоении полного объема учебного материала и успешном прохождении предусмотренных форм контроля программа может быть завершена досрочно.

Режим занятий: 2 раза в неделю по 2 академических часа.

1.6. Особенности организации образовательной деятельности по программе

Академический час составляет 45 минут. Объем учебной нагрузки в учебный день не превышает 8 часов. Образовательный процесс осуществляется с использованием электронной информационно-образовательной среды bstudy.ru.

1.7. Содержание разделов

№ п/п	Наименование раздела	Всего часов	Цель и краткое содержание	Формы контроля
1	Модуль 1. Введение. Руководство для прохождения обучения			
2	Модуль 2.1 Введение — основы Python	2	Цель: подготовить слушателей к практической работе с языком Python и настроить рабочую среду. Краткое содержание: слушатели знакомятся с характеристиками языка и Дзен Python, устанавливают Python, pip и виртуальные окружения, выбирают среду разработки: Jupyter Notebook / Lab, VS Code, PyCharm или Google Colab. Рассматриваются стандарт оформления кода PEP 8 и магические команды, пишутся первые программы.	Тестирование, практическая работа
3	Модуль 2.2 Типы данных	2,7	Цель: сформировать уверенное владение базовыми типами данных Python. Краткое содержание: слушатели изучают числа (int, float, complex) и системы счисления, строки: методы, срезы, f-строки и форматирование, логический тип bool, None и концепцию Truthy/Falsy. Отрабатывается работа с датами: модуль datetime, форматирование через strftime и разбор строк через strptime.	Тестирование, практическая работа
4	Модуль 2.3 Коллекции: список, кортеж, словарь, множества	2,5	Цель: научить выбирать и эффективно использовать встроенные коллекции Python. Краткое содержание: слушатели осваивают списки: индексы, срезы, методы	Тестирование, практическая работа

			и сортировку, кортежи: неизменяемость и хешируемость, словари: методы, вложенность, defaultdict и Counter, множества: операции и frozenset. Отрабатывается сравнение коллекций и выбор подходящей структуры данных под задачу.	
5	Модуль 2.4 Операторы и управление потоком. Функции	2,7	Цель: научить управлять ходом выполнения программы и обрабатывать ошибки. Краткое содержание: слушатели изучают арифметические, логические, побитовые операторы и операторы сравнения, присваивание и распаковку переменных, конструкции if / elif / else, тернарный оператор и match-case. Отрабатывается обработка исключений: try / except / else / finally, raise и создание кастомных исключений, а также первое знакомство с функциями.	Тестирование, практическая работа
6	Модуль 2.5 Основные элементы программирования: функции	2,6	Цель: сформировать навык проектирования и использования функций. Краткое содержание: слушатели осваивают объявление функций: def, return, docstring и аргументы по умолчанию, *args, **kwargs, позиционные и именованные параметры (/ и * в сигнатуре). Рассматриваются lambda-функции, map, filter, reduce, области видимости и замыкания. Отрабатывается создание декораторов, functools.wraps и @property.	Тестирование, практическая работа
7	Модуль 2.6 Итераторы, генераторы и циклы. ООП	2,7	Цель: научить эффективно перебирать данные и применять объектно-ориентированный подход. Краткое содержание: слушатели изучают протокол итератора (iter/next), циклы for и while: range, enumerate, break/continue/else, walrus-оператор, генераторы (yield), генераторные выражения и включения (list/dict/set comprehensions). Рассматриваются основы ООП: классы, __init__, атрибуты, @property, @staticmethod, @classmethod, наследование, super(), MRO, полиморфизм и dunder-методы.	Тестирование, практическая работа

8	Модуль 2.7 Итоговый проект	10	Цель: объединить все освоенные навыки программирования на Python в одном законченном проекте. Краткое содержание: слушатель выполняет комплексный итоговый проект, в котором самостоятельно проектирует и реализует программу на Python. Участник продумывает структуру проекта, применяет типы данных, коллекции, функции, обработку исключений, итераторы и классы, при необходимости использует работу с файлами (CSV, JSON) и регулярные выражения, оформляет код по PEP 8, готовит презентацию результата и проводит финальную проверку работы программы.	Итоговый проект по курсу «Создание Игры»
---	-------------------------------	----	--	---

РАЗДЕЛ 2. СОДЕРЖАНИЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

2.1. Учебно-тематический план

№ п/п	Название раздела, темы	Всего часов	Теоретических (видеолекция + материалы)	Практических (семинар, тест, ДЗ)	Формы аттестации
2	МОДУЛЬ 2.1 Введение — основы Python	2	0,4	1,6	Тест, дз
3	МОДУЛЬ 2.2 Типы данных	2,7	0,7	2	Тест, дз
4	МОДУЛЬ 2.3 Коллекции: список, кортеж, словарь, множества	2,5	0,5	2	Тест, дз
5	МОДУЛЬ 2.4 Операторы и управление потоком. Функции	2,7	0,7	2	Тест, дз
6	МОДУЛЬ 2.5 Основные элементы программирования: функции	2,6	0,6	2	Тест, дз
7	МОДУЛЬ 2.6 Итераторы, генераторы и циклы. ООП	2,7	0,7	2	Тест, дз
8	МОДУЛЬ 2.7 Итоговый проект	10	1	9	Итоговый проект

9	МОДУЛЬ 3.1 Дополнительный материал	100	100	0	Самоконтроль
	Всего:	25,2			

2.2. Календарный учебный график

Неделя	Форма занятия	Контактная работа (ч.)	Самостоятельная работа (ч.)	Раздел	Форма контроля
1	Самостоятельная	2,1	2,6	Модули 2.1-2.2. Введение — основы Python. Типы данных	Тест, дз
2	Самостоятельная	0,7	2	Модуль 2.3 Коллекции: список, кортеж, словарь, множества	Тест, дз
3	Самостоятельная	0,7	2	Модуль 2.4 Операторы и управление потоком. Функции	Тест, дз
4	Самостоятельная	0,6	2	Модуль 2.5 Основные элементы программирования: функции	Тест, дз
5	Самостоятельная	0,7	2	Модуль 2.6 Итераторы, генераторы и циклы. ООП	Тест, дз
6	Самостоятельная	0,5	9,5	Модуль 2.7 Итоговый проект: комбинированный проект на выбор. Всего: 6 проектов.	Итоговый проект

2.3. Рабочая программа

Модуль 2.1. Практическая видеолекция посвящена знакомству с языком Python, его философией и возможностями применения. Слушатели установят Python, настроят рабочую среду, познакомятся с Jupyter Notebook, VS Code, PyCharm и Google Colab, изучат стандарт оформления кода PEP 8 и напишут свои первые программы.

Самостоятельная работа: тест, практическая работа (2 ак. ч.).

Модуль 2.2. Практическая видеолекция посвящена основным типам данных Python.

Слушатели научатся работать с числами, строками, логическими значениями и датами, освоят преобразование типов, форматирование строк и базовые операции с данными.

Самостоятельная работа: тест, практическая работа (2,7 ак. ч.).

Модуль 2.3. Практическая видеолекция посвящена встроенным коллекциям Python.

Слушатели освоят работу со списками, кортежами, словарями и множествами, научатся выбирать подходящие структуры данных, выполнять поиск, сортировку, сравнение и обработку информации.

Самостоятельная работа: тест, практическая работа (2,5 ак. ч.).

Модуль 2.4. Практическая видеолекция посвящена операторам, управляющим

конструкциям и обработке ошибок. Слушатели изучат условия, циклы, различные виды операторов, научатся управлять логикой выполнения программы и обрабатывать исключительные ситуации.

Самостоятельная работа: тест, практическая работа (2,7 ак. ч.).

Модуль 2.5. Практическая видеолекция посвящена созданию функций и повторному

использованию кода. Слушатели освоят различные способы передачи аргументов, функциональный стиль программирования, области видимости, замыкания и декораторы для построения более гибких программ.

Самостоятельная работа: тест, практическая работа (2,6 ак. ч.).

Модуль 2.6. Практическая видеолекция посвящена итераторам, генераторам, работе с

циклами, файлами и основам объектно-ориентированного программирования. Слушатели научатся эффективно обрабатывать данные, использовать стандартные библиотеки Python, создавать собственные классы и применять принципы наследования и полиморфизма.

Самостоятельная работа: тест, практическая работа (2,7 ак. ч.).

Модуль 2.7. Практическая видеолекция посвящена разработке итогового проекта.

Слушатели объединят все изученные инструменты Python, познакомятся с регулярными выражениями, особенностями работы памяти и передачей объектов, а также реализуют и презентуют собственную программу как завершённый практический проект.

Самостоятельная работа: тест, практическая работа, итоговый проект (10 ак. ч.).

РАЗДЕЛ 3. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ПРОГРАММЕ

3.1. Формы аттестации

Текущий контроль: Тестирование после каждой лекционной темы (проходной балл – 75%), проверка домашних заданий.

Итоговая аттестация: Защита итогового практического проекта. Оценка «зачтено» / «не зачтено». Для получения сертификата необходимо успешно пройти все тесты и получить «зачтено» по итоговому проекту.

3.2. Контрольно-оценочные средства промежуточного и итогового контроля

Типовые тестовые вопросы:

Формат – выбор одного правильного ответа из нескольких.

Пример:

1. Какая из встроенных коллекций Python является неизменяемой?

- A. Кортеж (tuple)
- B. Список (list)
- C. Словарь (dict)
- D. Множество (set)

2. Для чего используется конструкция try / except?

- A. Для перехвата и обработки ошибок во время выполнения программы
- B. Для ускорения работы циклов
- C. Для объявления новых переменных
- D. Для форматирования строк

3. Что возвращает функция enumerate() при переборе списка?

- A. Пары «индекс — элемент»
- B. Только элементы в обратном порядке
- C. Сумму всех элементов
- D. Копию списка без дублей

4. Зачем нужны виртуальные окружения в Python?

- A. Чтобы изолировать зависимости разных проектов друг от друга
- B. Чтобы код работал быстрее

- C. Чтобы запускать Python без установки
- D. Чтобы менять цвет интерфейса редактора

Пример итогового задания:

Итоговый практический проект на курсе:

КОНСОЛЬНЫЙ ОРГАНАЙЗЕР ЗАДАЧ

(Базовый уровень)

Для кого подходит

Тем, кто хочет создать свою первую законченную программу на Python.

Входные данные

- список типовых задач пользователя;
- требования к функциям программы: добавление, просмотр, отметка о выполнении и удаление задач;
- понятный ожидаемый результат.

Что нужно сделать

Подготовить:

1. описание задачи и структуры программы;
2. функции для добавления, просмотра, изменения и удаления задач;
3. саму программу на Python с обработкой ошибок ввода;
4. сохранение задач в файл (CSV или JSON) и их загрузку при запуске;
5. короткое пояснение, как устроена программа и какие темы курса в ней применены;
6. при желании улучшенную версию: классы, декораторы, генераторы.

Примеры тем

- органайзер задач или список покупок;
- дневник привычек со статистикой;
- телефонная книга с поиском;
- учет личных расходов;
- викторина с подсчетом очков;
- конвертер валют и единиц измерения;
- генератор паролей;
- текстовая игра с сохранением прогресса.

Этот проект хорошо показывает, что уже после первого курса по Python можно создать полезную законченную программу, применив типы данных, коллекции, функции, циклы и работу с файлами.

Критерии оценки итогового проекта:

Полнота выполнения этапов анализа (0-20 баллов), корректность применения методов (0-20 баллов), качество оформления и презентации результатов (0-15 баллов), обоснованность выводов (0-15 баллов). Проходной балл – 48 из 80 (60%).

РАЗДЕЛ 4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММ

4.1. Требования к материально-техническому обеспечению

При реализации данной образовательной программы созданы условия для функционирования электронной информационно-образовательной среды, включающей в себя электронные информационные ресурсы, электронные образовательные ресурсы, совокупность информационных технологий, телекоммуникационных технологий, соответствующих технологических средств, которые обеспечивают освоение слушателями образовательной программы в полном объеме независимо от места нахождения слушателей.

Формирование информационной среды осуществляется с помощью программной системы дистанционного обучения.

В условиях реализации образовательной программы в дистанционном формате с использованием дистанционных образовательных технологий слушатели сами обеспечивают создание необходимых материально-технических условий для осуществления учебной деятельности.

Необходимым минимальным условием использования дистанционных образовательных технологий является наличие интернет-браузера и подключения к сети Интернет. На компьютере также должен быть установлен комплект соответствующего программного обеспечения. Для работы с использованием аудиоканала, в том числе видеоконференций, необходимо наличие веб-камеры, микрофона и динамиков (наушников).

Слушатель должен иметь возможность использовать канал связи с пропускной способностью не ниже: 512 Кбит/с, для более комфортной связи рекомендовано 1 Мбит/с. Требования к скорости доступа в сеть Интернет носят рекомендательный характер и должны соблюдаться в целях беспрепятственного и своевременного освоения образовательной программы.

Для эффективного внедрения дистанционных образовательных технологий и использования электронных образовательных ресурсов имеется качественный доступ педагогических работников и слушателей к информационно-телекоммуникационной сети Интернет (далее - сеть Интернет) с использованием установленных программно-технических средств для слушателей и педагогических работников на скорости не ниже 512 Кбит/с; в труднодоступных районах, подключаемых к сети Интернет с использованием спутниковых каналов связи, скорость прямого канала должна быть не ниже 512 Кбит/с, обратного - не ниже 128 Кбит/с; обеспечен порт доступа в сеть Интернет со скоростью не

ниже 10 Мбит/с и возможностью установления не менее 20 одновременных сессий по 512 Кбит/с.

Услуга подключения к сети Интернет должна предоставляться в режиме 24 часа в сутки 7 дней в неделю без учета объемов потребляемого трафика за исключением перерывов для проведения необходимых ремонтных и профилактических работ при обеспечении совокупной доступности услуг не менее 99,5% в месяц.

Требования к скорости доступа в сеть Интернет носят рекомендательный характер и должны соблюдаться в целях беспрепятственного и своевременного освоения образовательной программы.

Для использования дистанционных образовательных технологий каждому слушателю и преподавателю предоставляется свободный доступ к средствам информационных и коммуникационных технологий.

В наличии необходимое программное обеспечение, серверное оборудование, обеспечивающее функционирование электронной информационно-образовательной среды, и высокоскоростной канал доступа к электронной информационно-образовательной среде:

1. Офисная мебель – стол и стул (рабочее место педагогического работника)
2. Демонстрационная доска (1 шт.)
3. Ноутбук (процессор с тактовой частотой от 2,0 ГГц, оперативная память — не менее 8 ГБ, твердотельный накопитель (SSD) — не менее 256 ГБ) (1 шт.)
4. Оборудование для подключения к сети интернет - роутер Wi-Fi (1 шт.)
5. Наушники – гарнитура с микрофоном (1 шт.)
6. Установлено программное обеспечение, необходимое для осуществления учебного процесса общего назначения (Антивирус KasperskyFree для Windows, Браузер Яндекс.Браузер, Яндекс.Документы).

Обучение проводится посредством электронной образовательной среды (платформы) bestudy.ru.

Платформа bestudy.ru. позволяет:

- размещать обучающие материалы и задания;
- загружать слушателям выполненные письменные, фото, видео задания, а также вопросы в адрес педагога;
- проводить вебинары педагогами, а также в процессе их проведения задавать вопросы в форме текстовых сообщений;
- осуществлять контроль прогресса изучения учебных материалов, количество выполненных слушателями заданий, а также проверять выполненные ими задания.

Требования к ПК слушателя:

ОС: Windows 7+, macOS 10.12+, Linux

ОЗУ: 4 ГБ (рекомендуется 8 ГБ)

Интернет: 2 Мбит/с минимум

Веб-камера и микрофон

Необходимое ПО (все бесплатное):

Python 3.8+ (через Anaconda Distribution)

JupyterNotebook

Веб-браузер (Chrome, Firefox, Safari)

4.2. Информационно-методическое обеспечение обучения

Рекомендованная литература:

- **Марк Лутц** — *Изучаем Python* (Learning Python)
Фундаментальное руководство по языку Python: типы данных, коллекции, функции, модули, работа с файлами и объектно-ориентированное программирование. Подходит в качестве основной книги по всему курсу.
- **Эрик Мэтиз** — *Изучаем Python. Программирование игр, визуализация данных, веб-приложения* (Python Crash Course)
Практическое пособие для начинающих, позволяющее закрепить основы Python на реальных примерах и проектах.
- **Официальная документация Python** (Python Documentation, Python Tutorial)
Основной источник информации по синтаксису языка, стандартной библиотеке и современным возможностям Python. Рекомендуется использовать при изучении всех тем программы.
- **PEP 8** — **Style Guide for Python Code**
Официальный стандарт оформления кода Python. Содержит рекомендации по стилю программирования, именованию объектов и структуре проектов.
- **Дэн Бейдер** — *Python Tricks. Приёмы эффективного Python* (Python Tricks)
Практические рекомендации по написанию понятного, лаконичного и современного Python-кода.
- **Лучано Рамальо** — *Python. К вершинам мастерства* (Fluent Python)
Углублённое руководство по внутреннему устройству Python: модели данных, итераторы, генераторы, функции высшего порядка, декораторы и объектно-ориентированное программирование.
- **Эл Свейгарт** — *Автоматизация рутинных задач с помощью Python* (Automate the Boring Stuff with Python)
Практическое руководство по автоматизации работы с файлами, таблицами, текстом, регулярными выражениями и другими повседневными задачами.
- **Real Python**
Современная образовательная платформа с большим количеством практических руководств по Python, стандартной библиотеке, ООП, работе с файлами, декораторам, генераторам и другим темам курса.
- **Документация стандартной библиотеки Python** (*collections, itertools, functools, pathlib, os, re, datetime, csv, json*)
Рекомендуется для изучения встроенных модулей, используемых при работе с коллекциями, файлами, регулярными выражениями, обработкой данных и автоматизацией.

4.3. Кадровое обеспечение реализации программы

Квалификация руководящих и научно-педагогических работников организации соответствует квалификационным характеристикам, установленным действующим законодательством Российской Федерации в области образования.

Требования к квалификации преподавателей:

- ✓ наличие высшего образования и/или опыта профессиональной деятельности в сфере разработки программного обеспечения, ИТ или цифровой трансформации;
- ✓ стаж работы в образовательной организации не менее 1 года либо наличие учёной степени (кандидата или доктора наук) без предъявления требований к стажу работы;
- ✓ регулярное повышение квалификации (не реже одного раза в три года), в том числе через программы по применению дистанционных образовательных технологий (ДОТ) и электронного обучения;
- ✓ владение современными методами преподавания, включая разработку и использование электронных образовательных ресурсов.

Преподаватели, участвующие в реализации программы с применением дистанционных технологий, проходят повышение квалификации в области информационных и образовательных технологий не реже одного раза в 5 лет. Для поддержания актуальности профессиональных компетенций организуются внутренние семинары, конференции и стажировки.