

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



СЕРТИФИКАТ

о. ректора

М.В. Румянцев М.В. Румянцев
07 « 08 2025 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

«Python в производстве: автоматизация и аналитика»

Красноярск 2025

I. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

I.1. Нормативная правовая основа Программы

Дополнительная профессиональная программа (программа профессиональной переподготовки) ИТ-профиля «Python в производстве: автоматизация и аналитика» (далее — Программа) разработана в соответствии с нормами Федерального закона РФ от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»; постановлением Правительства РФ от 11 октября 2023 г. № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»; приказом Минобрнауки России от 19 октября 2020 г. № 1316 «Об утверждении порядка разработки дополнительных профессиональных программ, содержащих сведения, составляющие государственную тайну, и дополнительных профессиональных программ в области информационной безопасности» (далее – приказ Минобрнауки России № 1316); методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов (утв. Минобрнауки России 22 января 2015 г. № ДЛ-1/05вн); с учетом требований приказа Минобрнауки России от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам», с изменениями, внесенными приказом Минобрнауки России от 15 ноября 2013 г. № 1244 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденный приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499»; приказа Министерства образования и науки РФ от 23 августа 2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»; паспорта федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика Российской Федерации»; постановления Правительства Российской Федерации от 13 мая 2021 г. № 729 «О мерах по реализации программы стратегического лидерства «Приоритет-2030» (в редакции постановления Правительства Российской Федерации от 14 марта 2022 г. № 357 «О внесении изменений в постановление Правительства Российской Федерации от 13 мая 2021 г. № 729»); приказа Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 28 февраля 2022 г. № 143 «Об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации» и признании утратившими силу некоторых приказов Министерства цифрового развития, связи и массовых коммуникаций

Российской Федерации об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации»; федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии (уровень бакалавриата), утвержденного приказом Минобрнауки России от 19 сентября 2017 г. № 926, (далее — ФГОС ВО), а также профессионального стандарта 06.001 «Программист», утвержденного приказом Министерства труда и социальной защиты РФ от 20.07.2022 № 424н.

Профессиональная переподготовка заинтересованных лиц (далее — Слушатели), осуществляемая в соответствии с Программой, имеющей отраслевую направленность «Обрабатывающая промышленность (металлургия)», проводится в ФГАОУ ВО «Сибирский федеральный университет» (далее — Университет) в соответствии с учебным планом в очно-заочной форме обучения.

Разделы, включенные в учебный план Программы, используются для последующей разработки календарного учебного графика, учебно-тематического плана, рабочих программ модулей (дисциплин), оценочных и методических материалов. Перечисленные документы разрабатываются Университетом самостоятельно, с учетом актуальных положений законодательства об образовании, законодательства в области информационных технологий и смежных областей знаний ФГОС ВО и профессионального стандарта 06.001 «Программист».

1.2. Цель программы

Целью ДПП ПП является формирование у слушателей, обучающихся по ОП ВО – программам бакалавриата и программам специалитета (начиная со 2 курса), программам магистратуры (начиная с 1 курса) по специальностям и направлениям подготовки, относящимся к обрабатывающей промышленности цифровых компетенций в области применения языков программирования для решения профессиональных задач, применения интегрированных сред разработки (IDE), а также приобретение по итогам прохождения ДПП ПП новой квалификации «Программист».

1.3 Требования к поступающим

К обучению по Программе допускаются обучающиеся по ОП ВО – программам бакалавриата и программам специалитета (начиная со 2 курса), программам магистратуры (начиная с 1 курса) по специальностям и направлениям подготовки, относящимся к обрабатывающей промышленности.

1.4. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и(или) уровней квалификации

1.4.1. Область профессиональной деятельности слушателя, прошедшего обучение по программе профессиональной переподготовки, в которой может осуществлять профессиональную деятельность: разработка компьютерного программного обеспечения.

Выпускники могут осуществлять профессиональную деятельность в других областях и(или) сферах профессиональной деятельности при условии соответствия уровня их образования и полученных компетенций требованиям к квалификации работника.

1.4.2. Объекты профессиональной деятельности: программное обеспечение, веб-приложения, системы анализа данных.

Виды профессиональной деятельности: разработка компьютерного программного обеспечения.

1.4.3. Уровень квалификации. В соответствии с приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. № 424н «Об утверждении Профессионального стандарта «Программист», дополнительная профессиональная программа профессиональной переподготовки «Python в производстве: автоматизация и аналитика» обеспечивает достижение шестого уровня квалификации.

1.4.4. Компетенции (трудовые функции) в соответствии с профессиональным стандартом (формирование новых или совершенствование имеющихся)

Программа разработана в соответствии с актуальными квалификационными требованиями, профессиональными стандартами специалистов. Виды профессиональной деятельности, трудовые функции, указанные в профессиональном стандарте 06.001 «Программист», представлены в таблицах 1–2.

Таблица 1

Характеристика новой квалификации, связанной с видом профессиональной деятельности и трудовыми функциями в соответствии с профессиональным стандартом 06.001 «Программист»

Трудовые действия	Трудовая функция	Обобщенная трудовая функция	Вид профессиональной деятельности
Составление формализованных описаний решений, поставленных задач в соответствии с требованиями технического задания или внутренних документов организации. Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или внутренних документов организации.	A/01.3 Формализация и алгоритмизация поставленных задач для разработки программного кода	А Разработка и отладка программного кода	
Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)	A/02.3 Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных		
Проверка работоспособности компьютерного программного обеспечения на основе разработанных тестовых наборов данных	B/03.4 Проверка работоспособности программного обеспечения	В Проверка работоспособности и рефакторинг кода программного обеспечения	Разработка программного обеспечения
Сборка программных модулей и компонентов в программный продукт Подключение программного продукта к компонентам внешней среды	C/02.5 Осуществление интеграции программных модулей и компонентов и проверки работоспособности выпусков программного продукта	С Интеграция программных модулей и компонентов и проверка работоспособности выпусков программного продукта	

Трудовые действия	Трудовая функция	Обобщенная трудовая функция	Вид профессиональной деятельности
Разработка, изменение архитектуры компьютерного программного обеспечения и ее согласование с системным аналитиком и архитектором программного обеспечения	D/03.6 Проектирование компьютерного программного обеспечения	D Разработка требований и проектирование программного обеспечения	

Таблица 2

Характеристика новой и развивающейся цифровой компетенции в ИТ-сфере, связанной с уровнем формирования и развития в результате освоения программы «Python в производстве: автоматизация и аналитика»

Наименование сферы	Тип компетенции	Наименование компетенции	Номер компетенции (ID)	БАЗОВЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ
Средства программной разработки	Компетенция применима в различных отраслях экономики	Применяет языки программирования для решения профессиональных задач	28	Применяет языки программирования (в т.ч. скрипты) для решения профессиональных задач под контролем более опытных специалистов
Средства программной разработки	Компетенция применима в различных отраслях экономики	Применяет интегрированные среды разработки (IDE)	31	Применяет IDE. Использует инструменты отладки и проверку синтаксиса под контролем опытных специалистов

Таблица 3

Структура образовательных результатов

ID и формулировка целевого уровня формирования компетенций	Промежуточные образовательные результаты		
	Опыт практической деятельности (ОПД)	Умения (У)	Знания (З)
28. Применяет языки программирования для решения профессиональных задач	<p>Разработка и внедрение скриптов на Python для автоматизации рутинных задач.</p> <p>Создание простых программ для обработки данных и выполнения аналитических расчетов.</p> <p>Моделирование технологических процессов с использованием Python.</p> <p>Разработка приложений с графическим интерфейсом для упрощения взаимодействия с пользователем.</p>	<p>Умение писать и отлаживать код на Python, используя различные структуры данных.</p> <p>Способность применять функции и лямбда-функции для решения задач.</p> <p>Умение читать и записывать данные в текстовые файлы, а также работать с форматами CSV и JSON.</p> <p>Способность использовать библиотеки для анализа данных (NumPy, Pandas) и визуализации (Matplotlib, Seaborn).</p>	<p>Знание основ синтаксиса Python, включая переменные, типы данных и управляющие конструкции.</p> <p>Понимание принципов работы с библиотеками Python для анализа данных и визуализации результатов.</p> <p>Знание основ разработки веб-приложений на Flask или Django.</p> <p>Понимание основ информационной безопасности и методов защиты от уязвимостей.</p>
31. Применяет интегрированные среды разработки (IDE)	<p>Настройка окружения разработки: Установка и настройка IDE (например, PyCharm, VS Code) для работы с проектами на Python.</p> <p>Использование инструментов отладки: Применение встроенных инструментов отладки в IDE для поиска и исправления ошибок в коде.</p>	<p>Умение эффективно использовать инструменты IDE для повышения производительности разработки.</p> <p>Способность настраивать среду разработки под конкретные задачи и требования проекта.</p> <p>Умение использовать инструменты автоматизации для упрощения</p>	<p>Знание популярных интегрированных сред разработки для Python (например, PyCharm, VS Code).</p> <p>Понимание принципов работы с виртуальными окружениями и управления зависимостями в проектах.</p> <p>Знание основ работы с системами контроля версий и их интеграции в IDE.</p>

	<p>Использование плагинов и расширений: Установка и использование дополнительных плагинов для повышения продуктивности разработки.</p>	<p>процесса разработки и развертывания приложений.</p>	
--	--	--	--

1.5. Требования к материально-техническому обеспечению, необходимому для реализации дополнительной профессиональной программы профессиональной переподготовки (требования к аудитории, компьютерному классу, программному обеспечению)

Обучение производится на платформе электронного обучения СФУ «e-Курсы» (<https://e.sfu-kras.ru/>). Используются сервисы вебинаров и видеоконференций.

При проведении лекций, практических занятий, самостоятельной работы слушателей и практики используется следующее оборудование: компьютер с наушниками или аудиоколонками, микрофоном и веб-камерой, высокоскоростное подключение к Интернет (не менее 5 Мбит/с).

Программное обеспечение (обновленное до последней версии): браузер Google Chrome, PyCharm, Java, текстовый редактор.

1.6. Особенности (принципы) построения дополнительной профессиональной программы профессиональной переподготовки

Особенности построения программы переподготовки «Python в производстве: автоматизация и аналитика»:

- в основу проектирования программы положен компетентностный подход;
- выполнение учебных заданий, требующих практического применения знаний и умений, полученных в ходе изучения логически связанных дисциплин;
- выполнение итоговых аттестационных работ по реальному заданию;
- использование информационных и коммуникационных технологий, в том числе современных систем технологической поддержки процесса обучения, обеспечивающих комфортные условия для обучающихся, преподавателей;
- применение электронных образовательных ресурсов (дистанционное, электронное, комбинированное обучение и пр.).

В поддержку дополнительной профессиональной программы профессиональной переподготовки разработан электронный курс на платформе <http://e.sfu-kras.ru>.

1.7. Особенности организации практики

Практика слушателей дополнительной профессиональной программы переподготовки «Python в производстве: автоматизация и аналитика» представляет собой вид учебной деятельности, непосредственно ориентированный на профессионально-практическую подготовку слушателей. Практика осуществляется в целях формирования и закрепления профессиональных умений и навыков, полученных в результате теоретической подготовки.

Сроки проведения практики устанавливаются графиком учебного процесса в объеме 36 часов в конце процесса обучения в соответствии с утвержденным в установленном порядке учебно-тематическим планом.

В рамках очно-заочной формы обучения на основе дистанционных технологий практика осуществляется в форме online практики.

1.8. Документ об образовании: диплом о переподготовке установленного образца.

УЧЕБНЫЙ ПЛАН
дополнительной профессиональной программы профессиональной переподготовки
«Python в производстве: автоматизация и аналитика»

Форма обучения – очно-заочная, с применением электронного обучения и дистанционных образовательных технологий.
Срок обучения – 256 часов.

№ п/п	Наименование дисциплин	Общая трудоем- кость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Формы Контроля
				Лекции	Лабора- торные работы	Практические и семинарские занятия		
1.	Базовые навыки работы с Python	50	28	10		18	22	Зачет
2.	Применение Python для моделирования технологических процессов	56	30	10		20	26	Зачет
3.	Разработка автоматизированных решений	90	50	10		40	40	Зачет
4.	Практика	36	18	-		18	18	Зачет
5.	Итоговая аттестация	24	8	-		8	16	Выполнение итоговой аттестационной работы
Итого		256	133	29		104	123	

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН
дополнительной профессиональной программы профессиональной переподготовки
«Python в производстве: автоматизация и аналитика»

Категория слушателей: лица, имеющие или получающие высшее образование.

Срок обучения: 256 часов.

Форма обучения: очно-заочная.

Режим занятий: 6 часов в неделю.

№ п/п	Наименование дисциплин	Общая трудоем- кость, ч	Всего контактн., ч	Контактные часы			CPC, ч
				Лекции	Лабора- торные работы	Практ. и семинарские занятия	
1	Базовые навыки работы с Python	50	28	10		18	22
1.1	Установка Python и настройка окружения	8	4	2		2	4
1.2	Основы синтаксиса: переменные, типы данных, операторы. Управляющие конструкции (условные операторы (if, else, elif), циклы (for, while))	10	6	2		4	4
1.3	Функции (определение и вызов функций, аргументы и возвращаемые значения, лямбда-функции)	12	6	2		4	6
1.4	Структуры данных (справки, кортежи, множества и словари, основные операции с коллекциями)	10	6	2		4	4
1.5	Файловый ввод-вывод (чтение и запись текстовых файлов, работа с CSV и JSON файлами)	10	6	2		4	4
2	Применение Python для моделирования технологических процессов	56	30	10		20	26
3.1	Введение в анализ данных	14	6	2		4	8
3.2	Обработка и анализ данных с использованием библиотек (NumPy, Pandas)	20	12	4		8	8
3.3	Библиотеки для визуализации (Matplotlib, Seaborn)	22	12	4		8	10
3.	Разработка автоматизированных решений	90	50	10		40	40

№ п/п	Наименование дисциплин	Общая трудоем- кость, ч	Всего контактн., ч	Контактные часы			CPC, ч
				Лекции	Лабора- торные работы	Практ. и семинарские занятия	
3.1	Разработка приложений. Основы графического интерфейса (Tkinter)	24	12	2		10	12
3.2	Работа с базами данных. Работа с фреймворком SQLAlchemy	28	14	4		10	14
3.3	Создание простых веб-приложений (ведение в Flask или Django для создания веб-приложений.	34	24	4		20	10
3.4	Основы информационной безопасности. Защита от распространенных уязвимостей.	4	0				4
4	Практика	36	18			18	18
5	Итоговая аттестация	24	8	-		8	16
Всего		256	133	29		104	123

Календарный учебный график
дополнительной профессиональной программы профессиональной переподготовки
«Python в производстве: автоматизация и аналитика»

Наименование модулей (курсов) Объем учебной нагрузки, ч.	Сентябрь												Октябрь			Ноябрь			декабрь			Январь			февраль			Март			Апрель			май			июнь						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
Базовые навыки работы с Python																																											
Применение Python для моделирования технологических процессов																			K	K																							
Разработка автоматизированных решений																																											
Практика																																											
Итоговая аттестация																																											

II. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Обучение по программе профессиональной переподготовки «Python в производстве: автоматизация и аналитика» реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде комплекса мини-видеолекций, записей занятий, текстовых материалов, презентаций, размещаемых в системе электронного обучения СФУ «е-Курсы» (<https://e.sfu-kras.ru>). Данные материалы сопровождаются заданиями и дискуссиями в чате программы. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации дисциплины

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя практические занятия, сочетающие в себе ответы на вопросы, связанные с материалом лекции, в формате дискуссий, а также групповую и индивидуальную работу. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи SaluteJazz.

Учебно-методическое и информационное обеспечение дисциплины

Программа может быть реализована как очно, так и заочно, в том числе с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, семинарские, активные и ситуативные методы обучения.

По программе разработан электронный учебно-методический комплекс (УМК) — электронный курс в системе электронного обучения СФУ «е-Курсы». Обучающиеся могут дополнить представленные материалы, подключая к учебной работе иные источники информации, освещающие обсуждаемые проблемы.

Содержание комплекта учебно-методических материалов

Учебно-методический комплекс содержит: систему навигации по программе (учебно-тематический план, интерактивный график работы по программе, сведения о результатах обучения, о преподавателе дисциплины, чат для объявлений и вопросов преподавателю), набор презентации к лекциям, набор ссылок на внешние образовательные ресурсы и инструменты, систему заданий с подробными инструкциями, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи, а также онлайн-площадки для взаимного обучения.

Виды и содержание самостоятельной работы

Выполнение самостоятельной работы слушателями предполагается в дистанционном режиме в рамках электронного курса, размещенного в системе электронного обучения СФУ. Самостоятельно слушателями изучаются представленные кейсы с лучшими практиками, дополнительные ссылки и материалы по темам курса, а также краткие резюмирующие материалы,

дополнительные инструкции в различных форматах (видео, скринкасты, подкасты, интерактивные справочники, текстовые пояснения).

Также слушатели самостоятельно проводят анализ и систематизацию материала в рамках выполнения практических заданий и решения ситуаций. Для оценки уровня усвоения изученного учебного материала, слушатели проходят контрольные тесты.

III. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

4.1. Учебно-методическое обеспечение, в т.ч. электронные ресурсы сети Интернет

Модуль 1. Базовые навыки работы с Python

1. Чернышев С. Основы программирования на Python: учеб. пособие для вузов. – М.: Litres, 2021.

2. Кучунова Е.В., Олейников Б.В., Чередниченко О.М. Программирование. Процедурное программирование. – Красноярск: СФУ, 2016.

3. Real Python Tutorials [Электронный ресурс]. – Режим доступа: <https://realpython.com/>.

4. Прикладное программирование на языке Python [Электронный ресурс]. – Режим доступа: <https://openedu.ru/course/urfu/PYAP/>.

5. Официальная документация [Электронный ресурс]. – Режим доступа: <https://docs.python.org>.

Модуль 2. Введение в анализ данных

1. Гэддис, Т. Начинаем программировать на PYTHON: пер. с англ. / Т. Гэддис. – 4-е изд. – СПб.: БХВ-Петербург, 2019.

2. Пател, Анкур. Прикладное машинное обучение без учителя с использованием Python: Пер. с англ. – СПб.: ООО «Диалектика», 2020. – 432 с.

3. Коэльо Луис Педро, Ричарт Вилли. Построение систем машинного обучения на языке Python. 2-е изд. / Пер. с англ. Слинкин А А. – М.: ДМК Пресс, 2016. – 302 с.

4. Real Python Tutorials [Электронный ресурс]. – Режим доступа: <https://realpython.com/>.

5. Прикладное программирование на языке Python [Электронный ресурс]. – Режим доступа: <https://openedu.ru/course/urfu/PYAP/>.

6. Официальная документация [Электронный ресурс]. – Режим доступа: <https://docs.python.org>.

7. Machine Learning in Python [Электронный ресурс]. – Режим доступа: <https://scikit-learn.org/stable/>.

Модуль 3. Разработка автоматизированных решений

1. Гниденко И.Г. Технологии и методы программирования: учеб. пособие для вузов / И.Г. Гниденко, Ф.Ф. Павлов, Д.Ю. Федоров. – М.: Издательство «Юрайт», 2021. – 235 с.

2. Real Python Tutorials [Электронный ресурс]. – Режим доступа: <https://realpython.com/>.

3. Прикладное программирование на языке Python [Электронный ресурс]. – Режим доступа: <https://openedu.ru/course/urfu/PYAP/>.
4. Официальная документация [Электронный ресурс]. – Режим доступа: <https://docs.python.org>.
5. Django documentation [Электронный ресурс]. – Режим доступа: <https://docs.djangoproject.com>.
6. Git Book [Электронный ресурс]. – Режим доступа: <https://git-scm.com/book/ru/v2>.

IV. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

2.1. Формы аттестации, оценочные материалы, методические материалы

Программа предусматривает проведение текущей и итоговой аттестации. Текущая аттестация слушателей проводится по дисциплинам на основе выполнения заданий в электронном обучающем курсе, а также с учетом результатов промежуточного ассесмента.

Методические материалы, необходимые для выполнения текущих заданий, представлены в соответствующих элементах электронного обучающего курса и включают описание задания, методические рекомендации по его выполнению, критерии оценивания.

2.2. Требования и содержание итоговой аттестации

После завершения обучения по Программе обучающиеся допускаются к итоговой аттестации. Аттестация проводится с участием представителей профильных индустриальных партнеров. Итоговая аттестация по программе включает выполнение итоговой аттестационной работы (ИАР) в форме проекта. Основная цель итоговой аттестационной работы— выполнить работу, демонстрирующую уровень подготовленности к самостоятельной профессиональной деятельности.

ИАР выполняется индивидуально или в группах по 2-4 человека. Слушатель предоставляет результат выполненной работы в формате PDF, оформленной и отвечающей требованиям к содержанию итоговой аттестационной работы. Список использованных источников литературы приводится в конце ИАР. Документ прикрепляется в организационный электронный курс программы профессиональной переподготовки. В итоговой аттестационной работе должны быть четко обозначены область и актуальность работы, постановка задачи, приведены результаты, полученные слушателем. Требования и содержание итоговой аттестации изложены в методических указания к выполнению ИАР и размещаются на платформе электронных курсов СФУ.

Выполнение итоговой аттестационной работы является обязательным.

По результатам выполнения ИАР аттестационная комиссия принимает решение о предоставлении слушателям по результатам освоения дополнительной профессиональной программы профессиональной переподготовки права заниматься профессиональной деятельностью в сфере разработки программного обеспечения и выдаче диплома о профессиональной переподготовке.

Примерные задания на выполнение ИАР.

Вариант 1: Моделирование технологического процесса (на примере процесса плавления металла)

Описание разработки:

Студенты должны разработать программу, моделирующую процесс плавления металла, учитывая такие параметры, как температура, время и состав сплава. Программа должна использовать математические модели для расчета времени плавления и визуализировать результаты с помощью графиков.

Компетенции, оцениваемые в задании:

1. Применяет интегрированные среды разработки (IDE)	
Показатели оценки	Оценка/норма
Использование интегрированной среды разработки	<p>Оценка: проект должен быть разработан в одной из интегрированных сред разработки (например, PyCharm, VSCode).</p> <p>Норма: код организован и структурирован с использованием возможностей IDE (например, управление зависимостями, использование виртуальных окружений).</p>
2. Применяет языки программирования для решения профессиональных задач	
Показатели оценки	Оценка/норма
Корректность выполнения программы	Оценка: программа корректно моделирует процесс плавления и выводит результаты.

	Норма: результаты должны соответствовать ожидаемым значениям по заданным параметрам.
Эффективность кода	<p>Оценка: код написан с использованием оптимальных алгоритмов.</p> <p>Норма: время выполнения программы не превышает установленные пределы (например, расчет результатов за не более чем 5 секунд).</p>
Документация кода	<p>Оценка: код содержит комментарии и документацию.</p> <p>Норма: каждая функция имеет строку документации (docstring).</p>
Обоснованность решений	<p>Оценка: все выбранные методы обоснованы.</p> <p>Норма: в конце проекта представлен отчет о выборе математических моделей.</p>

Вариант 2: Анализ данных производственного процесса

Описание разработки:

Создание приложения, которое загружает данные о производственном процессе из CSV файла, обрабатывает их с использованием Pandas и визуализирует результаты с помощью Matplotlib. Студенты должны проанализировать данные и предоставить выводы о производительности процесса.

Компетенции, оцениваемые в задании:

1. Применяет интегрированные среды разработки (IDE)	
Показатели оценки	Оценка/норма
Использование интегрированной среды разработки	<p>Оценка: проект должен быть разработан в одной из интегрированных сред разработки (например, PyCharm, VSCode).</p>

	<p>Норма: код организован и структурирован с использованием возможностей IDE (например, управление зависимостями, использование виртуальных окружений).</p>
Организация проекта	<p>Оценка: проект структурирован логически и удобно для навигации.</p> <p>Норма: наличие четкой структуры каталогов и файлов.</p>
2. Применяет языки программирования для решения профессиональных задач	
Показатели оценки	Оценка/норма
Корректность выполнения программы	<p>Оценка: программа корректно загружает и обрабатывает данные.</p> <p>Норма: результаты анализа должны соответствовать данным из CSV файла.</p>
Эффективность кода	<p>Оценка: код написан с использованием оптимальных методов обработки данных.</p> <p>Норма: время выполнения обработки данных не превышает 10 секунд для 1000 записей.</p>
Документация кода	<p>Оценка: код содержит комментарии и документацию.</p> <p>Норма: каждая функция имеет строку документации (docstring).</p>
Обоснованность решений	<p>Оценка: все методы анализа обоснованы.</p> <p>Норма: отчет должен содержать объяснение выбора методов анализа данных.</p>

Вариант 3: Разработка графического интерфейса для мониторинга процессов

Описание разработки:

Студенты должны создать приложение с графическим интерфейсом (GUI) на Tkinter или PyQt, которое позволяет пользователю вводить параметры технологического процесса и получать результаты в реальном времени.

Компетенции, оцениваемые в задании:

1. Применяет интегрированные среды разработки (IDE)	
Показатели оценки	Оценка/норма
Использование интегрированной среды разработки	<p>Оценка: проект должен быть разработан в одной из интегрированных сред разработки (например, PyCharm, VSCode).</p> <p>Норма: код организован и структурирован с использованием возможностей IDE (например, управление зависимостями, использование виртуальных окружений).</p>
Организация проекта	<p>Оценка: проект структурирован логически и удобно для навигации.</p> <p>Норма: наличие четкой структуры каталогов и файлов.</p>
2. Применяет языки программирования для решения профессиональных задач	
Показатели оценки	Оценка/норма
Корректность выполнения программы	<p>Оценка: программа корректно обрабатывает ввод пользователя и отображает результаты.</p> <p>Норма: все функции интерфейса работают без ошибок.</p>
Эффективность кода	<p>Оценка: код написан с использованием оптимальных подходов к обработке событий.</p> <p>Норма: время отклика интерфейса не превышает 1 секунды.</p>

Документация кода	Оценка: код содержит комментарии и документацию. Норма: каждая функция имеет строку документации (docstring).
Обоснованность решений	Оценка: все выбранные элементы интерфейса обоснованы. Норма: отчет должен содержать объяснение выбора элементов интерфейса.

Вариант 4: Создание веб-приложения для управления данными

Описание разработки:

Разработка простого веб-приложения с использованием Flask или Django, которое позволяет пользователям загружать данные о производственных процессах, сохранять их в базе данных и отображать результаты на веб-странице.

Компетенции, оцениваемые в задании:

1. Применяет интегрированные среды разработки (IDE)

Показатели оценки	Оценка/норма
Использование интегрированной среды разработки	Оценка: проект должен быть разработан в одной из интегрированных сред разработки (например, PyCharm, VSCode). Норма: код организован и структурирован с использованием возможностей IDE (например, управление зависимостями, использование виртуальных окружений).
Организация проекта	Оценка: проект структурирован логически и удобно для навигации. Норма: наличие четкой структуры каталогов и файлов.
2. Применяет языки программирования для решения профессиональных задач	
Показатели оценки	Оценка/норма

Корректность выполнения программы	Оценка: приложение корректно обрабатывает запросы и взаимодействует с базой данных. Норма: все функции CRUD работают без ошибок.
Эффективность кода	Оценка: код написан с использованием оптимальных подходов к работе с базами данных. Норма: время обработки запросов не превышает 2 секунд.
Документация кода	Оценка: код содержит комментарии и документацию. Норма: каждая функция имеет строку документации (docstring).
Обоснованность решений	Оценка: все методы работы с данными обоснованы. Норма: отчет должен содержать объяснение выбора технологий для разработки.

Вариант 5. Разработка базы данных для автоматизации металлургических технологий

Описание задания:

Необходимо разработать базу данных, которая будет использоваться для хранения информации о металлургических процессах, материалах, оборудовании и результатах испытаний. База данных должна обеспечивать возможность добавления, редактирования, удаления и поиска информации. Для реализации задания необходимо использовать язык Python и интегрированную среду разработки (IDE).

Компетенции, оцениваемые в задании:

1. Применяет интегрированные среды разработки (IDE)

Показатели оценки	Оценка/норма
Использование интегрированной среды разработки	Оценка: проект должен быть разработан в одной из

	<p>интегрированных сред разработки (например, PyCharm, VSCode).</p> <p>Норма: код организован и структурирован с использованием возможностей IDE (например, управление зависимостями, использование виртуальных окружений).</p>
Отладка и тестирование кода	<p>Оценка: код проходит отладку с использованием инструментов IDE.</p> <p>Норма: все ошибки исправлены, приложение работает корректно и без сбоев.</p>
2. Применяет языки программирования для решения профессиональных задач	
Показатели оценки	Оценка/норма
Корректность работы базы данных	<p>Оценка: база данных выполняет все заявленные функции (например, добавление, редактирование, удаление записей о материалах и процессах).</p> <p>Норма: все функции работают корректно и без ошибок, пользователь может легко взаимодействовать с базой данных.</p>
Эффективность кода	<p>Оценка: код написан с использованием оптимальных алгоритмов и структур данных.</p> <p>Норма: время выполнения операций (например, поиск материала) не превышает установленные пределы (например, менее 2 секунд на поиск).</p>
Документация кода	<p>Оценка: код должен содержать комментарии и документацию, объясняющую логику работы функций.</p> <p>Норма: Каждая функция имеет строку документации</p>

	(docstring), описывающую ее назначение и параметры.
Обоснованность решений	<p>Оценка: все выбранные методы и подходы должны быть обоснованы.</p> <p>Норма: в конце проекта должен быть представлен отчет, объясняющий выбор технологий и инструментов для разработки базы данных</p>

Эти темы могут быть адаптированы в зависимости от интересов студентов и специфики их обучения. Обучающиеся предоставляется право самостоятельно определить тему ИАР. Приветствуется выбор темы итогового проекта, связанный с профессиональной деятельностью студента по основному направлению подготовки или специальности.

Структура и содержание итоговой аттестационной работы

Задачи итоговой аттестационной работы должны иметь практическое значение. Во введении ИАР должна быть обоснована актуальность работы и сущность исследуемой проблемы, раскрыты цель, задачи, объект и предмет проектирования (разработки), методы выполнения проектирования (разработки). Как правило, ИАР состоит из 3 разделов (глав).

Глава 1 содержит обзор литературы по теме работы, в котором должны быть освещены различные точки зрения по затронутым в работе вопросам и обязательно сформулировано авторское отношение к ним, характеристика объекта и предмета разработки.

Глава 2 содержит характеристику методов выполнения работы, а также техническое задание на разработку программного продукта, включая перечень функций и возможностей, которые должен выполнять продукт, описание пользовательских сценариев и требований к интерфейсу, нефункциональные требования, этапы разработки и сроки их выполнения, описание методов и подходов к тестированию.

Глава 3 содержит характеристику результатов выполнения работы и их интерпретацию. В третьей главе студенту необходимо привести значимые участки программного кода, а также продемонстрировать визуально сценарии работы пользователя с программным продуктом.

В заключении формулируются конкретные выводы по работе и предложения по их реализации.

Этапы выполнения ИАР задаются графиком выполнения и контролируются руководителем. Примерный список основных этапов выполнения проекта представлен ниже:

1. Выбор темы ИАР. В начале обучения за студентом закрепляется руководитель итоговой работы, который прорабатывает тему. При

выборе темы итогового проекта берется во внимание вопросы, связанные с профессиональной деятельностью студента по основному направлению подготовки или специальности.

2. Формулировка цели и задач ИАР работы. Обоснование актуальности выбранного направления работы.
3. Поиск аналогичных решений, формирование концепции предлагаемых подходов, методов.
4. Выбор и/или проектирование программного, технического и иных видов обеспечений для достижения цели и решения задач ИАР.
5. Разработка программной реализации проекта, проведение практических испытаний и анализ результатов.
6. Формирование итоговой аттестационной работы.
7. Представление ИАР аттестационной комиссии.

Критерии оценивания итоговой аттестационной работы

Слушатель получает оценку «удовлетворительно», если:

1. Контроль со стороны преподавателя:
 - Разрабатывает программное обеспечение под контролем преподавателя, не проявляя инициативы в проектировании.
 - Выполняет задания по шаблонам или инструкциям без глубокого понимания процесса разработки.
2. Базовые навыки программирования:
 - Демонстрирует базовые навыки программирования на Python, но испытывает трудности с более сложными задачами.
 - Использует простые конструкции языка, не применяя более продвинутые возможности.
3. Использование интегрированной среды разработки:
 - Студент использует IDE, но демонстрирует ограниченные навыки работы с инструментами отладки и тестирования.
 - Проект имеет базовую структуру, однако не все файлы и каталоги организованы логически.
4. Корректность выполнения программы:
 - Программа выполняет основные функции, однако могут быть ошибки в расчетах или выводе результатов.
 - Результаты не всегда соответствуют ожидаемым значениям по заданным параметрам.
5. Эффективность кода:
 - Код работает, но может быть неэффективным и содержать излишние вычисления.
 - Код может содержать значительные недочеты, проблемы с читаемостью и структурой.
6. Документация кода:
 - Код не содержит комментариев.
 - Документация отсутствует или крайне ограничена.

7. Обоснованность решений:

- В отчете представлены общие соображения, но недостаточно обоснований для выбора конкретных методов.

Слушатель получает оценку «хорошо», если:

1. Самостоятельная разработка модулей:

- Разрабатывает отдельные модули или компоненты программы с минимальным контролем со стороны преподавателя.
- Умеет интегрировать свои модули в общую архитектуру проекта.

2. Использование интегрированной среды разработки:

- Студент уверенно использует IDE, применяет некоторые функции отладки и тестирования.
- Проект структурирован логично, но может быть небольшое количество недочетов в организации файлов.

3. Корректность выполнения программы:

- Программа корректно выполняет заявленный функционал, хотя могут быть незначительные неточности в выводе результатов.
- Результаты в большинстве случаев соответствуют ожидаемым значениям.

4. Эффективность кода:

- Код написан достаточно эффективно, однако есть возможности для оптимизации.
- Время выполнения программы не превышает 5 секунд.

5. Документация кода:

- Код содержит комментарии и документацию, но некоторые функции могут быть недостаточно описаны.

6. Обоснованность решений:

- В отчете представлены обоснования выбора технологий, однако некоторые аспекты требуют более глубокого анализа.

7. Работа в команде:

- Участвует в командной разработке, взаимодействует с другими участниками проекта, но требует некоторого контроля.

Слушатель получает оценку «Отлично», если

1. Экспертное применение Python:

- Применяет язык программирования Python на высоком уровне, демонстрируя глубокое понимание языка и его возможностей.
- Использует продвинутые конструкции языка, такие как генераторы, декораторы и контекстные менеджеры.

2. Системный подход к разработке:

- Контролирует весь цикл разработки программного обеспечения: от сбора требований до тестирования и развертывания.
- Самостоятельно проводит анализ требований, проектирует архитектуру приложения и реализует ее.

3. Использование интегрированной среды разработки:

- Студент демонстрирует отличные навыки работы с IDE, эффективно использует инструменты отладки и тестирования.
- Проект имеет четкую и логичную структуру с хорошо организованными файлами и каталогами.

4. Корректность выполнения программы:

- Программа полностью корректно выполняет заявленный функционал, результаты точно соответствуют ожидаемым значениям.

5. Эффективность кода:

- Код написан оптимально, использует эффективные алгоритмы и структуры данных.

6. Документация кода:

- Код содержит подробные комментарии.

7. Обоснованность решений:

- В отчете представлено глубокое обоснование выбора методов и подходов с учетом различных факторов и альтернативных подходов.

**РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Базовые навыки работы с Python»**

1. Аннотация

Основной целью данного модуля является формирование базовых навыков программирования на Python, которые станут основой для дальнейшего изучения более сложных аспектов языка и разработки программного обеспечения. По окончании модуля студенты смогут уверенно использовать Python для решения простых задач и будут готовы к более углубленному изучению языка и его библиотек.

Цели модуля:

- Ознакомить студентов с основами программирования на Python.
- Развить навыки написания простых программ с использованием базовых конструкций языка.
- Подготовить участников к более углубленному изучению программирования и разработке более сложных проектов.

Цель дисциплины (результаты обучения)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 1. Базовые навыки работы с Python (50 часов)			
1. Установка Python и настройка окружения (8 ч.)	Введение в Python Применение Python в юриспруденции. Установка Python: выбор версии и дистрибутива. Настройка среды разработки. Обзор популярных IDE и текстовых редакторов (PyCharm, VSCode, Jupyter Notebook). (2 ч.)	Установка Python, установка дополнительных инструментов (pip, virtualenv). Установка и настройка выбранной IDE. Настройка окружения для работы с Python. (2 ч.)	Изучение пакетного менеджера (4 ч.)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
2. Основы синтаксиса: переменные, типы данных, операторы. Управляющие конструкции (условные операторы (if, else, elif), циклы (for, while) (10 ч.)	<p>Переменные и их использование. Типы данных в Python.</p> <p>Операторы и выражения.</p> <p>Условные операторы (if, else, elif).</p> <p>Применение условных операторов в реальных задачах. Введение в циклы. (2 ч.)</p>	<p>Основы синтаксиса: переменные, типы данных, операторы. Исследование типов данных.</p> <p>Операторы и выражения (4 ч.)</p> <p>Практические задачи с комбинированием конструкций. (6 ч.)</p>	<p>Понятие исключительной ситуации, работа с исключениями (2 ч.)</p> <p>Документирование кода. Изучение стандартов документирования функций с помощью docstring. Написание документации для своих функций и модулей (2 ч.)</p>
4. Функции (определение и вызов функций, аргументы и возвращаемые значения, лямбда-функции) (12 ч.)	<p>Определение функций в Python. Вызов функций и область видимости переменных.</p> <p>Аргументы функций: позиционные и именованные. Значения по умолчанию для аргументов. Аргументы переменной длины: *args и **kwargs.</p> <p>Возвращаемые значения: возвращение нескольких значений и использование кортежей. (2 ч.)</p>	<p>Создание простых функций и их вызов. Работа с аргументами и значениями по умолчанию.</p> <p>Использование *args и **kwargs в функциях.</p> <p>Написание функций с несколькими возвращаемыми значениями.</p> <p>Применение лямбда-функций для упрощения код (4 ч.)</p>	<p>Изучение итераторов и декораторов.</p> <p>Лямбда-функции: определение и применение.</p> <p>Замыкания и функции высшего порядка (6 ч.)</p>
5. Файловый ввод-вывод (чтение и запись текстовых файлов,	Введение в файловый ввод-вывод: основные понятия. Чтение и запись текстовых	Чтение и запись текстовых файлов: практические задачи. Работа с	Изучение дополнительных библиотек для

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
работа с CSV и JSON файлами) (10 ч.)	файлов: методы и примеры. Работа с CSV файлами: чтение, запись и обработка данных. Работа с JSON файлами: структура, чтение и запись. Обработка ошибок при работе с файлами. (2 ч.)	CSV файлами: практические задачи и анализ данных. Работа с JSON файлами: практические задачи по сериализации и десериализации. (4 ч.)	работы с файлами (4 ч.)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиоколонками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, PyChamp, текстовый редактор.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

6. Чернышев С. Основы программирования на Python: учеб. пособие для вузов. – М.: Litres, 2021.
7. Кучунова Е.В., Олейников Б.В., Чередниченко О.М. Программирование. Процедурное программирование. – Красноярск: СФУ, 2016.

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины

1. Real Python Tutorials [Электронный ресурс]. – Режим доступа: <https://realpython.com/>.
2. Прикладное программирование на языке Python [Электронный ресурс]. – Режим доступа: <https://openedu.ru/course/urfu/PYAP/>.
3. Официальная документация [Электронный ресурс]. – Режим доступа: <https://docs.python.org>.
4. Django documentation [Электронный ресурс]. – Режим доступа: <https://docs.djangoproject.com>.
5. Machine Learning in Python [Электронный ресурс]. – Режим доступа: <https://scikit-learn.org/stable/>.
6. Git Book [Электронный ресурс]. – Режим доступа: <https://git-scm.com/book/ru/v2>.

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

тесты самоконтроля к лекциям 40 %;
практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100% от общего прогресса по курсу.

Примеры тестов для контроля знаний

Пример тестового задания по типу «Множественный выбор»

Вопрос: Основы синтаксиса

Какой из следующих операторов используется для проверки условия в Python?

- A) check
- B) if
- C) condition
- D) when

Вопрос: Структуры данных

Какой метод используется для добавления элемента в список в Python?

- A) append()
- B) add()
- C) insert()
- D) push()

Вопрос: Файловый ввод-вывод

Какой модуль в Python обычно используется для работы с CSV файлами?

- A) json
- B) csv
- C) file
- D) data

Типовое практическое задание (Тема: Основы работы в PyCharm)

Описание задания:

Создайте программу на Python, которая будет рассчитывать вес различных металлических изделий на основе их объема и плотности. Программа должна позволять пользователю вводить данные о металлах и выполнять расчеты.

Требования к заданию:

1. Создание структуры данных: используйте словарь для хранения информации о различных металлах и их плотности. Например:

```
metals = {
    «сталь»: 7850, кг/м3
    «алюминий»: 2700, кг/м3
    «меди»: 8960, кг/м3
    «железо»: 7874, кг/м3
}
```

2. Функции для расчетов: реализуйте функцию calculate_weight(volume, density), которая принимает объем (в кубических метрах) и плотность (в кг/м³) и возвращает вес (в кг).

3. Интерактивное меню: создайте текстовое меню, которое позволяет пользователю выполнять следующие действия:

1. Рассчитать вес металла
 2. Добавить новый металл
 3. Показать доступные металлы
 4. Выход
4. Обработка пользовательского ввода: при выборе пункта 1, программа должна запрашивать у пользователя:
- Название металла (проверяйте, существует ли металл в словаре).
 - Объем в кубических метрах.
 - Если металл не найден, выводите сообщение об ошибке и предлагайте повторный ввод.

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

РАБОЧАЯ ПРОГРАММА

дисциплины (модуля)

«Введение в анализ данных»

1. Аннотация

Модуль «Применение Python для моделирования технологических процессов» предназначен для изучения методов анализа и визуализации данных с использованием языка программирования Python. В рамках этого модуля студенты познакомятся с основами анализа данных, а также с ключевыми библиотеками, такими как NumPy и Pandas, которые позволяют эффективно обрабатывать и анализировать большие объемы данных. Кроме того, будет рассмотрено использование библиотек Matplotlib и Seaborn для визуализации результатов анализа, что является важным этапом в принятии обоснованных решений в технологических процессах. Модуль направлен на формирование навыков работы с данными и их представления в наглядной форме, что способствует более глубокому пониманию технологических процессов и улучшению качества принимаемых решений.

Цель данного модуля заключается в том, чтобы обучить студентов основам анализа и визуализации данных с использованием Python, что позволит им:

1. Освоить базовые концепции анализа данных и их значимость в контексте технологических процессов.
2. Научиться использовать библиотеки NumPy и Pandas для обработки и анализа данных, включая фильтрацию, агрегацию и манипуляцию данными.
3. Развить навыки визуализации данных с помощью Matplotlib и Seaborn, чтобы эффективно представлять результаты анализа и делать их доступными для интерпретации.
4. Применять полученные знания на практике для решения реальных задач, связанных с моделированием и оптимизацией технологических процессов в различных отраслях.

В результате изучения модуля студенты будут готовы к анализу данных в своих профессиональных областях, что повысит их конкурентоспособность на рынке труда.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 2. Применение Python для моделирования технологических процессов (56 часов)			

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Введение в анализ данных (14 ч.)	Основные термины и определения. Понятие данных (2 ч.)	Работа с данными, загрузка и предобработка данных. Генерация файла с данными, предобработка данных (4 ч.).	Поиск информации по теме анализ данных. (6 ч.)
Обработка и анализ данных с использованием библиотек (NumPy, Pandas) (20 ч)	Введение в библиотеки для анализа данных: NumPy и Pandas. Основы работы с Pandas: структуры данных и их использование. Фильтрация данных в Pandas: методы и примеры Агрегация данных в Pandas: группировка и сводные таблицы. (4 ч.)	Работа с библиотекой NumPy: практические задачи. Фильтрация данных в Pandas: практические задачи. Агрегация данных в Pandas: создание сводных таблиц. (8 ч.)	Изучение дополнительных возможностей Pandas (например, работа с временными рядами). (8 ч.)
Библиотеки для визуализации (Matplotlib, Seaborn) (22 ч.)	Визуализация данных с помощью Matplotlib и Seaborn (4 ч.)	Визуализация данных: построение графиков на основе обработанных данных (8 ч.)	Решение задач на фильтрацию и агрегацию данных с использованием реальных наборов данных (10 ч).

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения вебинаров применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиоколонками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Java, текстовый редактор.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине программы имеется электронный учебно-методический комплекс на платформе электронных курсов СФУ. УМК содержит: систему навигации по программе (учебно-тематический план, график работы по программе, сведения о результатах обучения, о преподавателях программы, форумы для объявлений и вопросов преподавателям), набор презентаций к лекциям, набор ссылок на внешние образовательные ресурсы и инструменты, систему заданий с подробными инструкциями, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

8. Гэддис, Т. Начинаем программировать на PYTHON: пер. с англ. / Т. Гэддис. – 4-е изд. – СПб.: БХВ-Петербург, 2019.
9. Пател, Анкур. Прикладное машинное обучение без учителя с использованием Python: Пер. с англ. – СПб.: ООО «Диалектика», 2020. – 432 с.

Дополнительная литература

1. Коэльо Луис Педро, Ричарт Вилли. Построение систем машинного обучения на языке Python. 2-е изд. / Пер. с англ. Слинкин А А. – М.: ДМК Пресс, 2016. – 302 с.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Real Python Tutorials [Электронный ресурс]. – Режим доступа: <https://realpython.com/>.
2. Прикладное программирование на языке Python [Электронный ресурс]. – Режим доступа: <https://openedu.ru/course/urfu/PYAP/>.

3. Официальная документация [Электронный ресурс]. – Режим доступа: <https://docs.python.org>.
4. Django documentation [Электронный ресурс]. – Режим доступа: <https://docs.djangoproject.com>.
5. Machine Learning in Python [Электронный ресурс]. – Режим доступа: <https://scikit-learn.org/stable/>.
6. Git Book [Электронный ресурс]. – Режим доступа: <https://git-scm.com/book/ru/v2>.

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

тесты самоконтроля к лекциям 40 %;
практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100% от общего прогресса по курсу.

Задания для самостоятельной работы

В самостоятельные работы входит изучение материала курса и закрепление заданий с практических уроков.

Практические задания

1. Генерация и предобработка данных для машинного обучения.
2. Реализация алгоритма регрессии на python.
3. Реализация метода кластеризации.

Задания для самостоятельной работы

В самостоятельные работы входит изучение материала курса и закрепление заданий с практических уроков.

Критерии оценивания заданий и/или контрольных вопросов

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

Примеры практических заданий

Задание 1. Реализация алгоритма регрессии на Python.

Цель задания: изучить и реализовать алгоритм регрессии.

Инструкция:

Шаг 1. Загрузка и предобработка данных (или генерация данных).

Шаг 2. Изучение алгоритма прогноза значений на основе восстановления функции с применением регрессии.

Шаг 3. Реализовать алгоритм на Python.

Шаг 4. Отобразить на графиках исходные данные и полученный результаты.

Обратную связь по выполненной работе слушатель получит напрямую. Некоторые работы будут прокомментированы на практическом задании.

Задание 2. Реализация алгоритма кластеризации на Python.

Цель задания: Реализовать несколько алгоритмов кластеризации на Python с использованием библиотеки scikit-learn.

Инструкция:

Шаг 1. Загрузка (или генерация) и предобработка данных.

Шаг 2. Выбрать несколько методов кластеризации из библиотеки scikit-learn.

Шаг 3. Написать скрипты для кластеризации данных с применением выбранных алгоритмов

Шаг 4. Отобразить результаты в графическом виде.

Обратную связь по выполненной работе слушатель получит напрямую. Некоторые работы будут прокомментированы на практическом задании.

Примеры тестовых заданий к лекциям

1. Назначение библиотеки matplotlib:

- a. визуализация данных;
- b. математические вычисления;
- c. построение моделей машинного обучения

2. Series — это:

- a. объект Pandas;
- b. отдельная библиотека;
- c. функция Pandas/

3. Какая функция возвращает количество уникальных элементов:

- a. nunique;
- b. unique;
- c. drop_duplicates/

4. Результатом прогноза в задаче регрессии является:

- a. вещественное число;
- b. метка класса/

5. Метод в scipy, который позволяет построить линейную регрессию:

- a. linregress;
- b. linear_regression;
- c. regression.

6. На каком множестве делается прогноз для оценки качества модели:

- a. на тестовом множестве;
- b. на обучающем множестве;
- c. на всем множестве.

РАБОЧАЯ ПРОГРАММА

дисциплины (модуля)

«Разработка автоматизированных решений»

1. Аннотация

Модуль «Разработка автоматизированных решений» предназначен для обучения студентов основам разработки программного обеспечения, включая создание приложений с графическим интерфейсом, работу с базами данных и разработку веб-приложений. В рамках данного модуля студенты познакомятся с популярными инструментами и технологиями, такими как Tkinter и PyQt для создания графических интерфейсов, а также с фреймворком SQLAlchemy для работы с базами данных. Модуль также включает введение в разработку веб-приложений с использованием Flask или Django, что позволяет студентам получить практические навыки создания полнофункциональных веб-сервисов. Кроме того, особое внимание уделяется основам информационной безопасности, включая защиту от распространенных уязвимостей, что является важным аспектом при разработке программных решений.

Цель данного модуля заключается в том, чтобы обучить студентов ключевым аспектам разработки автоматизированных решений, что позволит им:

1. Освоить принципы разработки приложений с графическим интерфейсом, используя такие инструменты, как Tkinter или PyQt.
2. Научиться работать с базами данных, применяя фреймворк SQLAlchemy для эффективного управления данными.
3. Развить навыки создания простых веб-приложений, используя Flask или Django, что позволит студентам создавать интерактивные и динамичные веб-сервисы.
4. Понять основы информационной безопасности и научиться защищать свои приложения от распространенных уязвимостей, что повысит надежность и безопасность разрабатываемых решений.

В результате изучения модуля студенты будут готовы к разработке современных автоматизированных решений в профессиональной сфере, что значительно расширит их профессиональные возможности и подготовит их к решению реальных задач в сфере информационных технологий.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 3. Разработка автоматизированных решений (90 часов)			
Разработка приложений. Основы	Введение в графические	Создание простого приложения с использованием Tkinter.	Изучение основ тестирования GUI-приложений (10 ч.)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
графического интерфейса (Tkinter) (34 ч.)	интерфейсы. Основы Tkinter. Управление компоновкой/ Событийное программирование/ Работа с меню и панелями инструментов (2 ч.)	Создать интерфейс с использованием различных менеджеров компоновки. Реализовать обработчики событий для различных виджетов (кнопки, текстовые поля). (10 ч.)	
Работа с базами данных. Работа с фреймворком SQLAlchemy (28.)	Введение в базы данных, реляционные и нереляционные базы данных. Основные компоненты ORM (Object-Relational Mapping) и его преимущества (2 ч.) Модели и схемы в SQLAlchemy. Взаимоотношения между моделями: один к одному, один ко многим, многие ко многим CRUD операции с использованием SQLAlchemy. Запросы и фильтрация данных Транзакции и управление сессиями (2 ч.)	Установка и настройка SQLAlchemy. Создание простого проекта и подключение к базе данных. Создание моделей и таблиц. Определение моделей данных и создание соответствующих таблиц в базе данных. Выполнение CRUD операций. Реализация функций для создания, чтения, обновления и удаления записей. Фильтрация и сортировка данных. Написание запросов для фильтрации и сортировки данных из таблиц. Работа с транзакциями. Реализация операций с использованием транзакций и управление сессиями (10 ч.)	Изучение документации SQLAlchemy. Ознакомление с официальной документацией и примерами использования Изучение дополнительных возможностей SQLAlchemy (14 ч.)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Создание простых веб-приложений (ведение в Flask или Django для создания веб-приложений. Применение веб-приложений в юридической практике) (34 ч)	Основные понятия и компоненты веб-разработки, архитектура клиент-сервер. Протокол HTTP. Методы HTTP (GET, POST, PUT, DELETE). Фреймворк Flask. Работа с формами в Flask. Введение в Django, основные принципы работы фреймворка, установка и настройка окружения. (4 ч)	Создание простого веб-приложения на Flask. Настройка окружения и создание базовой структуры проекта. Создание формы для ввода данных и обработка POST-запроса (4 ч.). Создание первого проекта на Django. Установка Django и создание базового проекта. Определение моделей и создание миграций. Создание представлений и маршрутов. Реализация формы для ввода данных пользователем. Работа с админкой Django, настройка админки для управления моделями (20 ч.)	Изучение документации Flask. Ознакомление с официальной документацией и примерами использования. Изучение документации Django. Ознакомление с официальной документацией и примерами использования. Изучение дополнительных возможностей Django. Исследование таких тем, как кэширование, сигнализация и обработка ошибок (10 ч.)
Основы информационной безопасности. Защита от распространенных уязвимостей. (4 ч)			Введение в информационную безопасность. Понятие и цели информационной безопасности. Основные принципы: конфиденциальность, целостность, доступность. Угрозы и уязвимости. Различия между угрозами и уязвимостями. Классификация угроз: физические, технические, организационные. Защита от распространенных уязвимостей (4 ч.)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения вебинаров применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиоколонками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Java, текстовый редактор.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине программы имеется электронный учебно-методический комплекс на платформе электронных курсов СФУ. УМК содержит: систему навигации по программе (учебно-тематический план, график работы по программе, сведения о результатах обучения, о преподавателях программы, форумы для объявлений и вопросов преподавателям), набор презентаций к лекциям, набор ссылок на внешние образовательные ресурсы и инструменты, систему заданий с подробными инструкциями, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

7. Гниденко И.Г. Технологии и методы программирования: учеб. пособие для вузов / И.Г. Гниденко, Ф.Ф. Павлов, Д.Ю. Федоров. – М.: Издательство «Юрайт», 2021. – 235 с.

Дополнительная литература

1. Гэддис, Т. Начинаем программировать на PYTHON: пер. с англ. / Т. Гэддис. – 4-е изд. – СПб.: БХВ-Петербург, 2019.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

7. Real Python Tutorials [Электронный ресурс]. – Режим доступа: <https://realpython.com/>.

8. Прикладное программирование на языке Python [Электронный ресурс]. – Режим доступа: <https://openedu.ru/course/urfu/PYAP/>.

9. Официальная документация [Электронный ресурс]. – Режим доступа: <https://docs.python.org>.

10. Django documentation [Электронный ресурс]. – Режим доступа: <https://docs.djangoproject.com>.

11. Git Book [Электронный ресурс]. – Режим доступа: <https://git-scm.com/book/ru/v2>.

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

тесты самоконтроля к лекциям 40 %;

практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100% от общего прогресса по курсу.

Задания для самостоятельной работы

В самостоятельные работы входит изучение материала курса и закрепление заданий с практических уроков.

Типовые практические задания (Задание: приложение для управления запасами)

Описание: Разработайте простое графическое приложение для управления запасами на складе. Приложение должно позволять пользователю добавлять, редактировать, удалять и просматривать информацию о различных товарах, их количестве и местоположении на складе. Данные должны храниться в базе данных с использованием SQLAlchemy.

Технологии: Python, Tkinter (или PyQt), SQLAlchemy.

Требования:

1. Графический интерфейс:

- Создайте главное окно приложения с заголовком "Управление запасами".

- Добавьте меню с пунктами "Добавить товар", "Редактировать товар", "Удалить товар" и "Просмотреть запасы".

2. База данных:

- Используйте SQLAlchemy для создания базы данных с таблицей products, которая должна содержать следующие поля:

- id (целое число, первичный ключ)
- name (строка, название товара)
- quantity (целое число, количество товара на складе)
- location (строка, местоположение на складе)

3. Функциональность:

- Добавление товара: создайте форму для ввода информации о новом товаре (название, количество, местоположение). После заполнения формы данные должны сохраняться в базе данных.

- Редактирование товара: реализуйте возможность редактирования информации о существующем товаре. Пользователь должен иметь возможность выбирать товар из списка и изменять его данные.

- Удаление товара: добавьте функциональность для удаления товара из базы данных. Пользователь должен иметь возможность выбирать товар из списка и удалять его.

- Просмотр запасов: реализуйте отображение всех запасов товаров в виде таблицы, где будут показаны название, количество и местоположение каждого товара.

4. Обработка ошибок:

- Добавьте обработку ошибок при вводе данных (например, проверка на наличие пустых полей или некорректных значений).

- Убедитесь, что приложение не выдает ошибок при работе с базой данных.

5. Дополнительные функции (опционально):

- Реализуйте возможность фильтрации запасов по названию товара.
- Добавьте возможность экспорта данных в CSV файл.

Ожидаемый результат: Рабочее приложение для управления запасами с графическим интерфейсом и функциональностью работы с базой данных. Приложение должно быть интуитивно понятным и удобным в использовании.

Критерии оценивания заданий и/или контрольных вопросов

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

Примеры тестовых заданий к лекциям

Вопрос 1: Разработка приложений. Основы графического интерфейса (Tkinter)

Какой метод в Tkinter используется для создания кнопки?

1. create_button()
2. Button()
3. add_button()
4. new_button()

Вопрос 2: Работа с базами данных. Работа с фреймворком SQLAlchemy

Какой метод используется для добавления нового объекта в сессию SQLAlchemy?

1. insert()
2. add()
3. append()
4. create()

Вопрос 3: Создание простых веб-приложений (введение в Flask или Django для создания веб-приложений)

Какой декоратор используется в Flask для определения маршрута (route) к функции?

1. @app.route()
2. @app.path()
3. @app.url()
4. @app.endpoint()

РАБОЧАЯ ПРОГРАММА ПРАКТИКИ

1. Аннотация

Основная цель данной практики — обучить студентов основам программирования на Python и показать, как этот язык может быть использован для автоматизации задач в промышленности. Участники научатся собирать, обрабатывать и анализировать данные, а также разрабатывать небольшие программы для автоматизации процессов.

Практика включает в себя теоретические занятия, практические работы и проектную деятельность. Участники будут работать над реальными кейсами, что позволит им применить полученные знания на практике. Основной задачей практики является закрепление в практической деятельности профессиональных компетенций, умений, навыков и знаний, полученных в ходе обучения.

Планируемые результаты:

По окончании практики слушатели будут способны составлять применять язык программирования Python для написания программного кода; осуществлять проверку работоспособности программного обеспечения на основе разработанных тестовых наборов данных; использовать при разработке программного обеспечения существующие типовые решения и шаблоны проектирования.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Практика (36 часов)			
1. Общие вопросы (ознакомление с предприятием)		Изучение общего технологического цикла производства программного обеспечения (4 ч.).	
2. Практическая часть		Решение практико- ориентированных задач. Обратная связь от экспертов (12 ч.)	Решение практико- ориентированных задач (18 ч.)

Содержание практики включает следующие этапы:

1. Знакомство с работодателями и презентация компаний: на этом этапе представители компаний расскажут о своих организациях, сферах деятельности и специфике задач, которые они решают в области ИТ. Участники смогут задать вопросы и узнать о реальных потребностях бизнеса.

2. Обсуждение реальных кейсов: представители работодателей представляют конкретные примеры задач, которые требуют автоматизации.

3. Формирование пула заданий. Наставники определяют актуальные задачи, которые могут быть автоматизированы с помощью Python. Это может включать в себя задачи по обработке данных, автоматизации и анализу данных.

4. Выполнение проектные заданий

Каждый участник выбирает одно или несколько заданий из сформированного пула, основываясь на своих интересах и навыках.

Участники работают над проектом, который включает в себя:

- Обработку и анализ собранных данных.
- Автоматизацию задач,
- Моделирование технологических процессов.

5. Проверка и тестирование: после завершения работы над проектами участники представляют свои результаты для проверки. Может использоваться система Yandex Context для оценки качества выполненных заданий. Тестирование проектов: участники тестируют свои программы на наличие ошибок и недочетов, а также проверяют их работоспособность в различных сценариях.

6. Обратная связь и доработка проектов: участники получают обратную связь от работодателей и наставников по своим проектам. Обсуждаются сильные стороны работы и области для улучшения. На основе полученных комментариев участники вносят изменения и улучшения в свои проекты, что позволяет повысить их качество и функциональность.

3. Условия реализации программы практики

Организационные и педагогические условия реализации программы

Обучение по программе практики реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Материал практических занятий представляется в виде синхронных занятий, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Практика проводится под руководством назначенного руководителя из числа профессорско-преподавательского состава Университета, а также руководителя из состава организации, структурных подразделениях организации, материально-техническое обеспечение которой соответствует профилю программы.

Учебно-методическое и информационное обеспечение

По данному модулю используется электронный УМК. УМК предполагает использование разных типов материалов, сопровождающих учебный процесс, включая информационные, обучающие и контролирующие. На платформе электронных курсов размещаются задания, приводится перечень необходимых для изучения материалов. Обучающиеся могут на протяжении прохождения практики обращаться к теоретической базе знаний.

4. Оценка качества освоения программы практики (формы аттестации, оценочные и методические материалы)

В качестве подтверждения прохождения практики на базе предприятий, организаций, учреждений, для зачета результатов обучения предъявляется дневник прохождения практики (Приложение 2) (*отчет в виде дневника прохождения практики*).

Программу составили:

Доцент, заведующий
кафедрой информационных систем
Института космических и информационных
технологий СФУ

А.В. Раскина

Ст. преподаватель кафедры прикладной информатики
Института космических и информационных
технологий СФУ

А.Н. Усов

Руководитель программы:

Доцент, заведующий
кафедрой информационных систем
Института космических и информационных
технологий СФУ

А.В. Раскина

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Наименование образовательной организации

Индивидуальный план слушателя, направляемого на практику

Фамилия, имя, отчество _____

Место работы и должность/статус _____

Название предприятия (организации), где проводится практика

Цель практики _____

Срок практики с «____» ____ 202 г. по «____» ____ 202 г.

План практики

№ п.п.	Перечень разрабатываемых (изучаемых) вопросов, виды работ	Количество часов	Форма отчета
1.			Дневник практики
2.			
3.			

СОГЛАСОВАНО

(должность ответственного)

(подпись)

*(расшифровка подписи) лица,
направляющего на практику)*

Наименование площадки

УТВЕРЖДАЮ
Руководитель площадки
_____ ФИО
«_____» 2022 г.
М.П.

**ДНЕВНИК
прохождения практики**

_____,
(фамилия, имя, отчество)
проходящего обучение в рамках дополнительной профессиональной программе
переподготовки «Python в производстве: автоматизация и аналитика»

Цель практики:

Руководители практики (от организации): _____
(должность) (ФИО)

1. Дневник

Дата	Выполняемая работа	Вопросы для консультантов и руководителей практики

2. Краткий отчет о практике

3. Заключение руководителя практики от принимающей организации

Руководитель практики

(подпись)

(расшифровка подписи)