

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



УТВЕРЖДАЮ

и.о. ректора

М.В. Румянцев М.В. Румянцев

« 07 » 08 2025 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

«Python для анализа данных в праве и социальных науках»

Красноярск 2025

I. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

I.1. Нормативная правовая основа Программы

Дополнительная профессиональная программа (программа профессиональной переподготовки) ИТ-профиля «Python для анализа данных в праве и социальных науках» (далее — Программа) разработана в соответствии с нормами Федерального закона РФ от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»; с 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 18 ноября 2013 г. № 679н.

Профессиональная переподготовка заинтересованных лиц (далее — Слушатели), осуществляемая в соответствии с Программой, имеющей отраслевую направленность «Социальная сфера (юриспруденция)», 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 дополнительной профессиональной программы (ДПП) 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 отрасли «Социальная сфера» 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 отрасли «Социальная сфера».

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 для анализа данных в праве и социальных науках» 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.

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.	А/01.3 Формализация и алгоритмизация поставленных задач для разработки программного кода	А Разработка и отладка программного кода	Разработка программного обеспечения
Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)	А/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	В/03.4 Проверка работоспособности программного обеспечения	В Проверка работоспособности и рефакторинг кода программного обеспечения	
Сборка программных модулей и компонентов в программный продукт Подключение программного продукта к компонентам внешней среды	С/02.5 Осуществление интеграции программных модулей и компонентов и проверки работоспособности выпусков программного продукта	С Интеграция программных модулей и компонентов и проверка работоспособности и 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	

Характеристика новой и развиваемой цифровой компетенции в ИТ-сфере, связанной с уровнем формирования и развития в результате освоения программы «Python для анализа данных в праве и социальных науках»

Наименование сферы	Тип компетенции	Наименование компетенции	Номер компетенции (ID)	БАЗОВЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ Способность проявляется под внешним контролем / при внешней постановке задачи/ обучающийся пользуется готовыми, рекомендованными продуктами
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 проверки синтаксиса под контролем опытных специалистов

Структура образовательных результатов

ID и формулировка целевого уровня формирования компетенций	Промежуточные образовательные результаты		
	Опыт практической деятельности (ОПД)	Умения (У)	Знания (З)
<p>28. Применяет языки программирования для решения профессиональных задач</p>	<p>Разработка скриптов: создание скриптов на Python для автоматизации задач в юриспруденции, таких как обработка документов и извлечение ключевых слов из текстов.</p> <p>Анализ данных: применение библиотек Python (например, Pandas, NumPy) для обработки и анализа данных судебной практики, включая фильтрацию и агрегацию информации.</p> <p>Веб-скрейпинг: использование библиотеки BeautifulSoup для сбора данных с юридических сайтов, включая автоматизацию сбора информации о судебных делах или нормативных актах.</p> <p>Создание веб-приложений: разработка простых веб-приложений с использованием Flask или Django для предоставления доступа к юридическим данным и инструментам анализа.</p> <p>Разработка телеграмм-ботов: Создание ботов для Telegram, которые могут</p>	<p>Программирование на Python: умение писать код на Python для решения конкретных задач в юриспруденции.</p> <p>Работа с библиотеками: умение использовать библиотеки для анализа данных (Pandas, NumPy), веб-скрейпинга (BeautifulSoup) и создания веб-приложений (Flask, Django).</p> <p>Автоматизация процессов: умение разрабатывать скрипты для автоматизации обработки документов и анализа данных.</p> <p>Работа с базами данных: умение взаимодействовать с базами данных с помощью SQLAlchemy для хранения и извлечения юридической информации.</p> <p>Создание отчетов и визуализация данных: умение представлять результаты анализа в виде отчетов и графиков.</p>	<p>Основы программирования: Знание синтаксиса Python, структур данных и основных операторов.</p> <p>Методы анализа данных: Знание методов фильтрации, агрегации и визуализации данных в контексте юридической практики.</p> <p>Технологии веб-скрейпинга: Знание принципов работы с HTML и основ веб-скрейпинга.</p> <p>Основы работы с фреймворками: Знание основ разработки веб-приложений на Flask или Django и принципов работы с API.</p> <p>Информационная безопасность: Знание основных принципов защиты приложений от уязвимостей.</p>

	предоставлять юридическую информацию или автоматизировать взаимодействие с клиентами.		
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>Использование плагинов и расширений: Установка и использование дополнительных плагинов для повышения продуктивности разработки.</p>	<p>Навигация по IDE: умение эффективно использовать интерфейс IDE для написания кода, выполнения команд и навигации по проекту.</p> <p>Отладка кода: умение использовать инструменты отладки для анализа и исправления ошибок в программах.</p> <p>Оптимизация рабочего процесса: умение настраивать IDE под свои нужды, используя плагины и расширения.</p>	<p>Функции IDE: Знание основных функций интегрированных сред разработки и их возможностей для повышения продуктивности.</p> <p>Основы отладки: Знание принципов отладки кода и методов поиска ошибок.</p>

I

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 «е-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 аудиокolonками, 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 для анализа данных в праве и социальных науках»:

- 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 для анализа данных в праве и социальных науках» 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 16 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 для анализа данных в праве и социальных науках»

Форма обучения – очно-заочная.

Срок обучения – 256 часов.

№ п/п	Наименование дисциплин	Общая трудоемкость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Формы контроля
				Лекции	Лабораторные работы	Практические и семинарские занятия		
1.	Введение в программирование	50	28	10		18	22	Зачет
2.	Работа с данными	56	30	10		20	26	Зачет
3.	Применение Python в юриспруденции	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 для анализа данных в праве и социальных науках»

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 трудоемкость, ч	Всего контактн., ч	Контактные часы			СРС, ч
				Лекции	Лабораторные работы	Практ. и семинарские занятия	
1	Введение в программирование	50	28	10		18	22
2.1	Установка Python и настройка окружения	12	6	2		4	6
2.2	Основы синтаксиса: переменные, типы данных, операторы	12	6	2		4	6
2.3	Управляющие конструкции (условные операторы (if, else, elif), циклы (for, while))	10	6	2		4	4
2.4	Функции (определение и вызов функций, аргументы и возвращаемые значения, лямбда-функции)	16	10	4		6	6
2	Работа с данными	56	30	10		20	26
2.1	Структуры данных (списки, кортежи, множества и словари, основные операции с коллекциями)	10	6	2		4	4
2.3	Файловый ввод-вывод (чтение и запись текстовых файлов, работа с CSV и JSON файлами)	10	6	2		4	4
2.4	Обработка данных (основы работы с библиотеками для анализа данных, фильтрация и агрегация данных)	30	12	4		8	18
3	Применение Python в юриспруденции	90	50	10		40	40
3.1	Автоматизация рутинных задач (создание скриптов для автоматизации обработки документов, работа с текстами: извлечение ключевых слов, анализ документов)	13	7	1		6	6

№ п/п	Наименование дисциплин	Общая трудоем- кость, ч	Всего контактн., ч	Контактные часы			СРС, ч
				Лекции	Лабора- торные работы	Практ. и семинарские занятия	
3.2	Работа с базами данных. Работа с фреймворком SQLAlchemy	14	8	2		6	6
3.3	Юридическая аналитика (применение Python для анализа судебной практики, визуализация данных)	13	7	1		6	6
3.4	Веб-скрейпинг (основы веб-скрейпинга с использованием библиотеки BeautifulSoup. сбор данных с юридических сайтов)	14	8	2		6	6
3.5	Создание простых веб-приложений (ведение в Flask или Django для создания веб-приложений. Применение веб-приложений в юридической практике)	16	10	2		8	6
2.6	Разработка телеграмм ботов. Работа с фреймворками PyTelegramBot и Aiogram	16	10	2		8	6
2.7	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

II. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Обучение по программе профессиональной переподготовки «Python для анализа данных в праве и социальных науках» 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. Введение в программирование

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. Работа с данными

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. Применение Python в юриспруденции

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 указаниях 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.

Задание 1. Анализ данных о судебных делах с использованием Python и парсинг данных с сайтов судов.

Описание задания:

Вам необходимо разработать программу на языке Python, которая будет:

1. Парсить данные о судебных делах с одного или нескольких сайтов судов.
2. Сохранять полученные данные в формате CSV.
3. Выполнять анализ собранных данных, включая фильтрацию по различным критериям (например, по статусу дела, судье или дате).
4. Генерировать отчет о результатах анализа и сохранять его в виде текстового файла.

Компетенции, оцениваемые в задании

1. Применяет интегрированные среды разработки (IDE)

Показатели оценки:

- Критерий 1: Использование интегрированной среды разработки
 - Оценка: проект должен быть разработан в интегрированной среде разработки (например, PyCharm, VSCode).
 - Норма: код организован и структурирован, с использованием возможностей IDE (например, виртуальные окружения, управление зависимостями).
- Критерий 2: Отладка и тестирование кода
 - Оценка: код проходит отладку с использованием инструментов IDE.
 - Норма: все ошибки исправлены, программа работает корректно и без сбоев.
- 2. Применяет языки программирования для решения профессиональных задач

Показатели оценки:

- Критерий 1: Корректность выполнения программы
 - Оценка: программа выполняет все заявленные функции (парсинг данных, анализ, фильтрация).
 - Норма: результаты анализа должны соответствовать данным, полученным с сайтов.
- Критерий 2: Эффективность кода
 - Оценка: код написан с использованием оптимальных алгоритмов и структур данных.
 - Норма: время выполнения программы не превышает установленные пределы (например, парсинг и анализ 1000 записей за не более чем 10 секунд).
- Критерий 3: Документация кода

- Оценка: код должен содержать комментарии и документацию, объясняющую логику работы функций.
- Норма: каждая функция имеет строку документации (docstring), описывающую ее назначение и параметры.
- Критерий 4: Обоснованность решений
 - Оценка: все выбранные методы и подходы должны быть обоснованы.
 - Норма: в конце проекта должен быть представлен отчет, объясняющий выбор инструментов для парсинга и анализа данных.

Задание 2. Разработка веб-приложения для управления судебными делами

Описание задания:

Вам необходимо разработать веб-приложение на языке Python, которое будет использоваться для управления судебными делами. Приложение должно включать функционал для добавления, редактирования, удаления и поиска дел, а также отображения информации о каждом деле. Пользовательский интерфейс должен быть интуитивно понятным и удобным для юристов.

Компетенции, оцениваемые в задании

1. Применяет интегрированные среды разработки (IDE)

Показатели оценки:

- Критерий 1: Использование интегрированной среды разработки
 - Оценка: проект должен быть разработан в интегрированной среде разработки (например, PyCharm, VSCode).
 - Норма: код организован и структурирован с использованием возможностей IDE (например, управление зависимостями, использование виртуальных окружений)
- Критерий 2: Отладка и тестирование кода
 - Оценка: код проходит отладку с использованием инструментов IDE.
 - Норма: все ошибки исправлены, приложение работает корректно и без сбоев.
- Критерий 3: Использование систем контроля верси
 - Оценка: код размещен в системе контроля версий (например, Git).
 - Норма: коммиты должны содержать понятные сообщения, отражающие изменения.

2. Применяет языки программирования для решения профессиональных задач

Показатели оценки:

- Критерий 1: Корректность выполнения приложения
 - Оценка: приложение выполняет все заявленные функции (добавление, редактирование, удаление и поиск дел).
 - Норма: все функции работают корректно и без ошибок.
- Критерий 2: Эффективность кода
 - Оценка: код написан с использованием оптимальных алгоритмов и структур данных.

- Норма: Время выполнения операций (например, поиск дела) не превышает установленные пределы (например, менее 2 секунд на поиск).
- Критерий 3: Документация кода
 - Оценка: код должен содержать комментарии и документацию, объясняющую логику работы функций.
 - Норма: каждая функция имеет строку документации (docstring), описывающую ее назначение и параметры.
- Критерий 4: Обоснованность решений
 - Оценка: все выбранные методы и подходы должны быть обоснованы.

- Норма: в конце проекта должен быть представлен отчет, объясняющий выбор технологий и инструментов для разработки приложения.

Задание 3. Разработка Telegram-бота для юридической практики

Описание задания:

Необходимо разработать Telegram-бота, который будет использоваться в юридической практике для автоматизации различных процессов, таких как управление делами, консультации клиентов, предоставление информации о законодательстве и т.д. Бот должен иметь интуитивно понятный интерфейс и обеспечивать высокую степень взаимодействия с пользователями.

Компетенции, оцениваемые в задании

1. Применяет интегрированные среды разработки (IDE)

Показатели оценки:

- Критерий 1: 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).
- Критерий 2: Отладка и тестирование кода
 - Оценка: код проходит отладку с использованием инструментов IDE.
 - Норма: все ошибки исправлены, приложение работает корректно и без сбоев.

2. Применяет языки программирования для решения профессиональных задач

Показатели оценки:

- Критерий 1: Корректность выполнения приложения
 - Оценка: бот выполняет все заявленные функции (например, добавление, редактирование, удаление дел, поиск информации).
 - Норма: все функции работают корректно и без ошибок, пользователь может легко взаимодействовать с ботом.
- Критерий 2: Эффективность кода
 - Оценка: код написан с использованием оптимальных алгоритмов и структур данных.
 - Норма: время выполнения операций (например, поиск дела) не превышает установленные пределы (например, менее 2 секунд на поиск).

- Критерий 3: Документация кода
 - Оценка: код должен содержать комментарии и документацию, объясняющую логику работы функций.
- Критерий 4: Обоснованность решений
 - Оценка: все выбранные методы и подходы должны быть обоснованы.
 - Норма: в конце проекта должен быть представлен отчет, объясняющий выбор технологий и инструментов для разработки бота.

Задание 4. Разработка базы данных для юридической практики

Описание задания:

Необходимо разработать базу данных, которая будет использоваться в юридической практике для хранения информации о клиентах, делах, документах и других юридических данных. 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:

- Критерий 1: 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).
- Критерий 2: Отладка и тестирование кода
 - Оценка: код проходит отладку с использованием инструментов IDE.
 - Норма: все ошибки исправлены, приложение работает корректно и без сбоев.

2. Применяет языки программирования для решения профессиональных задач

Показатели оценки:

- Критерий 1: Корректность работы базы данных
 - Оценка: база данных выполняет все заявленные функции (например, добавление, редактирование, удаление записей о клиентах и делах).
 - Норма: все функции работают корректно и без ошибок, пользователь может легко взаимодействовать с базой данных.
- Критерий 2: Эффективность кода
 - Оценка: код написан с использованием оптимальных алгоритмов и структур данных.
 - Норма: время выполнения операций (например, поиск клиента) не превышает установленные пределы (например, менее 2 секунд на поиск).
- Критерий 3: Документация кода
 - Оценка: код должен содержать комментарии и документацию, объясняющую логику работы функций.

- Норма: каждая функция имеет строку документации (docstring), описывающую ее назначение и параметры.
- Критерий 4: Обоснованность решений
 - Оценка: все выбранные методы и подходы должны быть обоснованы.
 - Норма: В конце проекта должен быть представлен отчет, объясняющий выбор технологий и инструментов для разработки базы данных.

Задание 5. Создание скриптов для автоматизации обработки документов и анализа текстов в юридической практике

Описание задания:

Вам необходимо разработать скрипты на языке Python, которые будут автоматизировать обработку юридических документов. Скрипты должны включать функции для извлечения ключевых слов, анализа содержания документов и формирования отчетов. Вы также должны использовать интегрированную среду разработки (IDE) для написания и отладки вашего кода.

Компетенции, оцениваемые в задании

1. Применяет интегрированные среды разработки (IDE)

Показатели оценки:

- Критерий 1: Использование IDE
 - Оценка: код должен быть написан в одной из интегрированных сред разработки (например, PyCharm, VSCode).
 - Норма: проект организован с использованием возможностей IDE (например, управление зависимостями, использование виртуальных окружений).
- Критерий 2: Отладка и тестирование кода
 - Оценка: код проходит отладку с использованием инструментов IDE.
 - Норма: все ошибки исправлены, и приложение работает корректно и без сбоев.
- Критерий 3: Структурирование проекта
 - Оценка: проект должен быть структурирован с использованием модулей и пакетов.
 - Норма: код должен быть организован логически, с четким разделением на функции и модули.

2. Применяет языки программирования для решения профессиональных задач.

Показатели оценки:

- Критерий 1: Корректность работы скриптов
 - Оценка: скрипты должны выполнять все заявленные функции (например, извлечение ключевых слов, анализ текста).
 - Норма: все функции работают корректно и без ошибок, результаты анализа соответствуют ожиданиям.
- Критерий 2: Эффективность кода
 - Оценка: код написан с использованием оптимальных алгоритмов и структур данных.
 - Норма: время выполнения операций (например, анализ документа) не превышает установленные пределы (например, менее 3 секунд на документ).

- Критерий 3: Документация кода
 - Оценка: код должен содержать комментарии и документацию, объясняющую логику работы функций.
 - Норма: каждая функция имеет строку документации (docstring), описывающую ее назначение и параметры.
- Критерий 4: Обоснованность решений
 - Оценка: все выбранные методы и подходы должны быть обоснованы.
 - Норма: в конце проекта должен быть представлен отчет, объясняющий выбор технологий и инструментов для автоматизации обработки документов.

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

Оценка «отлично»

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. Качество кода:
 - Пишет чистый, читаемый и поддерживаемый код.

Оценка «хорошо»

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. Основы проектирования:
 - Понимает основные принципы проектирования и архитектуры программного обеспечения.
 - Может составить базовую документацию для своих модулей и функций.
3. Тестирование:
 - Проводит базовое тестирование своего кода, включая написание простых тестов.
 - Умеет выявлять и исправлять ошибки в своем коде.
4. Код и документация:
 - Пишет код, который в основном соответствует стандартам качества, но могут быть небольшие недочеты.

5. Работа в команде:

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

Оценка «удовлетворительно»

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. Ограниченное тестирование:

- Проводит минимальное тестирование, часто не выявляя существующие ошибки.

4. Качество кода:

- Код может содержать значительные недочеты, проблемы с читаемостью и структурой.
- Документация отсутствует или крайне ограничена.

5. Зависимость от наставничества:

- Требуется постоянной помощи и направлений от преподавателя для выполнения задач.
- Не проявляет инициативы в поиске решений или улучшений в проекте.

III. ОСНОВНОЕ СОДЕРЖАНИЕ ПРОГРАММЫ

РАБОЧАЯ ПРОГРАММА дисциплины (модуля) «Введение в программирование»

1. Аннотация

Модуль "Введение в программирование" предназначен для изучения основ программирования с использованием языка Python. В ходе обучения студенты познакомятся с ключевыми концепциями и инструментами, необходимыми для разработки простых программ. Участники научатся устанавливать интерпретатор Python на различных операционных системах, а также настраивать рабочее окружение для комфортной разработки. Рассмотрим популярные редакторы кода и интегрированные среды разработки (IDE). В этом разделе будут рассмотрены базовые элементы языка: переменные, типы данных

(числа, строки, списки, кортежи, множества и словари) и операторы (арифметические, логические, сравнительные). Обучающиеся научатся правильно объявлять переменные и выполнять основные операции с данными. Студенты изучат условные операторы (if, else, elif), позволяющие реализовать ветвление в программах, а также циклы (for, while), которые обеспечивают повторение действий. Практические задания помогут закрепить навыки работы с этими конструкциями. Участники узнают о важности функций в программировании: как их определять и вызывать, как передавать аргументы и получать возвращаемые значения. Также будет рассмотрено использование лямбда-функций для создания анонимных функций. Этот раздел поможет студентам организовывать код и повышать его читаемость.

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. Введение в программирование (50 часов)			
1. Установка Python и настройка окружения (12 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. (4 ч.)	Изучение пакетного менеджера (6 ч.)
2. Основы синтаксиса: переменные, типы данных, операторы (12 ч.)	Переменные и их использование. Типы данных в Python. Операторы и выражения. (2 ч.)	Основы синтаксиса: переменные, типы данных, операторы. Исследование типов данных.	Понятие исключительной ситуации, работа с исключениями (4 ч.)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
		Операторы и выражения (4 ч.)	
3. Управляющие конструкции (условные операторы (if, else, elif), циклы (for, while) (10 ч.)	Условные операторы (if, else, elif). Применение условных операторов в реальных задачах. Введение в циклы..Синтаксис цикла for и примеры использования.Синтаксис цикла while и примеры использования. Как использовать break и continue. Примеры использования условных операторов внутри циклов. Использование циклов для обработки данных. (2 ч.)	Практические задачи с комбинированием конструкций. (6 ч.)	Документирование кода. Изучение стандартов документирования функций с помощью docstring. Написание документации для своих функций и модулей (6 ч.)
4. Функции (определение и вызов функций, аргументы и возвращаемые значения, лямбда-функции) (16 ч.)	Определение функций в Python. Вызов функций и область видимости переменных. Аргументы функций: позиционные и именованные. Значения по умолчанию для аргументов. Аргументы переменной длины: *args и **kwargs. Возвращаемые значения: возвращение нескольких значений и использование кортежей. (4 ч.)	Создание простых функций и их вызов. Работа с аргументами и значениями по умолчанию. Использование *args и **kwargs в функциях. Написание функций с несколькими возвращаемыми значениями. Применение лямбда-функций для упрощения код (6 ч.)	Изучение итераторов и декораторов. Лямбда-функции: определение и применение. Замыкания и функции высшего порядка (6 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 аудиокolonками, 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 1: Какой из следующих типов данных используется для хранения текстовой информации в Python?

- A) int
- B) float
- C) str
- D) list

Вопрос 2: Какой оператор используется для сравнения двух значений на равенство в Python?

- A) =
- B) ==
- C) !=

- D) <>

Пример тестового задания по типу «Верно/Неверно»

Вопрос 1: В Python переменные могут начинаться с цифры.

Ответ: Неверно

Вопрос 2: Функция print() используется для вывода данных на экран.

Ответ: Верно

Типовое практическое задание (Тема: Основы работы в PyCharm)

Задание: Создание простого скрипта для расчета стоимости юридических услуг

Шаг 1. Создайте новый проект в PyCharm.

- Откройте PyCharm и создайте новый проект, назвав его "LegalServicesCalculator".

Шаг 2. Создайте новый файл Python.

- В папке проекта создайте новый файл с именем calculator.py.

Шаг 3. Напишите программу, которая будет запрашивать у пользователя следующие данные:

- Часовая ставка (в рублях).
- Количество часов работы.

Шаг 4. Рассчитайте общую стоимость услуг и выведите результат на экран.

- Используйте функцию input() для получения данных от пользователя.
- Преобразуйте введенные данные в нужные типы (например, float для часовой ставки и int для количества часов).

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) «Работа с данными»

1. Аннотация

Модуль «Работа с данными» предназначен для изучения основных структур данных в Python, а также методов их обработки и взаимодействия с файловой системой. В рамках этого модуля студенты познакомятся с различными типами коллекций, научатся эффективно работать с текстовыми и структурированными файлами, а также осvoят базовые приемы анализа данных с использованием популярных библиотек.

Цели модуля:

По завершении модуля студенты смогут:

- Эффективно использовать различные структуры данных для хранения и обработки информации.
- Осуществлять ввод и вывод данных из файлов в различных форматах.
- Применять базовые методы анализа данных для извлечения полезной информации из наборов данных.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 2. Работа с данными (56 часов)			
Структуры данных (списки, кортежи, множества и словари, основные операции с коллекциями) (10 ч.)	Введение в структуры данных: списки, кортежи, множества и словари. Списки: создание, доступ к элементам, методы и операции. Кортежи: особенности и применение. Словари: создание, доступ к элементам, методы и операции. Основные операции с коллекциями: перебор, фильтрация и сортировка (2 ч.)	Работа со списками: создание и модификация. Применение кортежей в задачах. Использование множеств для удаления дубликатов и операций над множествами. Работа со словарями: создание, изменение и доступ к данным. Основные операции с коллекциями: практика на перебор, фильтрацию и сортировку (4 ч.)	Изучение дополнительных методов работы со структурами данных. Решение задач на применение структур данных (4 ч.)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Файловый ввод-вывод (чтение и запись текстовых файлов, работа с CSV и JSON файлами) (10 ч.)	Введение в файловый ввод-вывод: основные понятия. Чтение и запись текстовых 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.)	Чтение и запись текстовых файлов: практические задачи. Работа с CSV файлами: практические задачи и анализ данных. Работа с JSON файлами: практические задачи по сериализации и десериализации. (4 ч.)	Изучение дополнительных библиотек для работы с файлами (4 ч.)
Обработка данных (основы работы с библиотеками для анализа данных, фильтрация и агрегация данных) (30 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. Визуализация данных с помощью Matplotlib и Seaborn (2 ч.)	Работа с библиотекой NumPy: практические задачи. Фильтрация данных в Pandas: практические задачи. Агрегация данных в Pandas: создание сводных таблиц. Визуализация данных: построение графиков на основе обработанных данных (8 ч.)	Изучение дополнительных возможностей Pandas (например, работа с временными рядами). Решение задач на фильтрацию и агрегацию данных с использованием реальных наборов данных (8 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. Дронов В.А. Django. Практика создания Web-сайтов на Python: пособие / В.А. Дронов. – СПб.: Издательство «БХВ- Петербург», 2016.

8. 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.

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

Типовые практические задания (Тема: работа с данными)

Цель задания: научиться обрабатывать и анализировать данные, связанные с арбитражными делами, используя язык программирования Python. Это задание направлено на развитие навыков работы с файлами, регулярными выражениями, а также на умение проводить анализ данных.

Описание задания: — написать программу на Python, которая выполняет следующие действия:

Шаг 1. Создание файла с данными

Создайте текстовый файл с именем `arbitration_cases.txt`, содержащий информацию о нескольких арбитражных делах в следующем формате:

Дело №12345, 2023-01-15, ИП "Иванов И.И.", ООО "Ромашка", 100000, выиграно

Дело №67890, 2023-02-20, ИП "Петров П.П.", ООО "Лотос", 150000, проиграно

Дело №54321, 2023-03-10, ИП "Сидоров С.С.", ООО "Тюльпан", 200000, выиграно

Дело №98765, 2023-04-05, ИП "Смирнова Е.Е.", ООО "Незабудка", 120000, проиграно

Каждая строка содержит следующие данные:

- Номер дела
- Дата подачи дела
- Истец (ФИО)
- Ответчик (название компании)
- Сумма иска
- Результат дела (выиграно/проиграно)

Шаг 2. Чтение данных из файла

Откройте созданный файл и прочитайте его содержимое. Выведите каждую строку на экран.

Шаг 3. Обработка данных

Используйте регулярные выражения для извлечения информации о каждом арбитражном деле. Создайте список словарей, где каждый словарь представляет собой арбитражное дело с ключами: `case_number`, `date`, `claimant`, `defendant`, `claim_amount`, `result`.

Шаг 4. Анализ данных

Подсчитайте общее количество дел в файле. Определите количество выигранных и проигранных дел. Найдите дело с максимальной суммой иска и выведите его данные. Рассчитайте среднюю сумму иска по всем делам. Создайте отдельный список для выигранных дел и проигранных дел.

Шаг 5. Запись результатов в новый файл

Создайте новый текстовый файл с именем `arbitration_analysis.txt` и запишите в него результаты анализа:

Общее количество дел: X

Количество выигранных дел: Y

Количество проигранных дел: Z

Дело с максимальной суммой иска: <данные о деле>

Средняя сумма иска: A

Дополнительные требования:

1. Обработка ошибок: обработайте возможные исключения при работе с файлами (например, файл может не существовать). Также проверьте корректность формата данных перед их обработкой.

2. Форматирование даты: используйте модуль `datetime` для работы с датами и преобразуйте даты из строкового формата в объекты `datetime`

3. Улучшение вывода: при выводе результатов анализа форматируйте числа (например, сумму иска) в виде валюты.

4. Функциональное программирование: разделите код на функции для улучшения читаемости и повторного использования кода.

Пример вывода на экран:

Дело №12345, 2023-01-15, ИП "Иванов И.И.", ООО "Ромашка", 100000,
выиграно

Дело №67890, 2023-02-20, ИП "Петров П.П.", ООО "Лотос", 150000, проиграно

Дело №54321, 2023-03-10, ИП "Сидоров С.С.", ООО "Тюльпан", 200000,
выиграно

Дело №98765, 2023-04-05, ИП "Смирнова Е.Е.", ООО "Незабудка", 120000,
проиграно

Общее количество дел: 4

Количество выигранных дел: 2

Количество проигранных дел: 2

Дело с максимальной суммой иска: Дело №54321, 2023-03-10, ИП "Сидоров С.С.", ООО "Тюльпан", 200000, выиграно

Средняя сумма иска: 142500.00

Пример содержимого файла `arbitration_analysis.txt`:

Общее количество дел: 4

Количество выигранных дел: 2

Количество проигранных дел: 2

Дело с максимальной суммой иска: Дело №54321, 2023-03-10, ИП "Сидоров С.С.", ООО "Тюльпан", 200000, выиграно

Средняя сумма иска: 142500.00

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: Какой из следующих типов данных в Python является изменяемым?

- A) Кортеж
- B) Список
- C) Строка
- D) Множество

Пример тестового задания по типу «Верно/Неверно»

Утверждение: в Python словари могут содержать только строки в качестве ключей.

- Верно
- Неверно

Правильный ответ: неверно (Ключами могут быть любые неизменяемые типы данных, такие как строки, числа и кортежи.)

РАБОЧАЯ ПРОГРАММА

дисциплины (модуля)

«Применение Python в юриспруденции»

1. Аннотация

Модуль «Применение Python в юриспруденции» предназначен для изучения основ программирования на языке Python и его применения в юридической практике. В рамках данного курса слушатели познакомятся с основными библиотеками и инструментами, которые позволяют автоматизировать рутинные процессы, анализировать юридические данные и улучшать эффективность работы юристов.

Цели модуля: по завершении модуля студенты:

1. научатся создавать скрипты для автоматизации обработки документов, включая извлечение ключевых слов и анализ текстов, освоят работу с фреймворком SQLAlchemy, который позволяет взаимодействовать с различными базами данных;
2. познакомятся с основами веб-скрейпинга с использованием библиотеки BeautifulSoup, что позволит собирать данные с юридических сайтов и использовать их для анализа или разработки новых сервисов;
3. изучат основы разработки веб-приложений с использованием Flask или Django, что поможет им создавать интерактивные платформы для работы с юридическими данными и услугами;
4. рассмотрят работу с фреймворками PyTelegramBot и AioGram для создания телеграмм-ботов, которые могут помочь в автоматизации общения с клиентами и предоставлении юридической информации;
5. рассмотрят важные аспекты защиты от распространенных уязвимостей, что является критически важным для работы с юридическими данными и обеспечения конфиденциальности клиентов.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 3. Применение Python в юриспруденции (90 часов)			
1. Автоматизация рутинных задач (создание скриптов для автоматизации обработки документов, работа с текстами: извлечение ключевых слов,	Основы работы с Python для обработки текстов. Библиотеки для работы с документами (например, PyPDF2, docx). Анализ текстов и извлечение	Установка и настройка окружения для разработки. Создание скрипта для извлечения текста из документов. Использование регулярных выражений для поиска ключевых слов	Регулярные выражения для извлечения информации. (6 ч.)

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
анализ документов) (13 ч.)	ключевых слов. Практические примеры автоматизации в юриспруденции (1 ч.)	Разработка скрипта для анализа юридических документов (6 ч.)	
Работа с базами данных. Работа с фреймворком SQLAlchemy (14.)	<p>Введение в базы данных, реляционные и нереляционные базы данных. Основные компоненты ORM (Object-Relational Mapping) и его преимущества</p> <p>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.)</p>	<p>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 (6 ч.)</p>	<p>Изучение документации SQLAlchemy. Ознакомление с официальной документацией и примерами использования Изучение дополнительных возможностей SQLAlchemy (6 ч.)</p>
Юридическая аналитика (применение Python для анализа судебной практики, визуализация данных) (13 ч.)	<p>Введение в юридическую аналитику и ее значение. Основы работы с данными в Python (pandas, NumPy). Методы анализа судебной практики. Визуализация данных с помощью Matplotlib и Seaborn. Примеры успешного применения</p>	<p>Установка необходимых библиотек и настройка окружения. Загрузка и предварительная обработка данных судебной практики. Применение методов анализа данных на примерах. Создание визуализаций для представления результатов анализа (6 ч.)</p>	<p>Чтение литературы по юридической аналитике и методам анализа данных. Выполнение домашних заданий по анализу данных и визуализации. (6 ч.)</p>

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
	аналитики в юриспруденции (1 ч.)		
Веб-скрейпинг (основы веб-скрейпинга с использованием библиотеки BeautifulSoup. сбор данных с юридических сайтов) (14 ч.)	Введение в веб-скрейпинг. Основы работы с библиотекой BeautifulSoup. Сбор данных с юридических сайтов: анализ структуры HTML. Работа с запросами и библиотекой requests (2 ч.)	Установка и настройка окружения для веб-скрейпинга. Практическое применение BeautifulSoup для сбора данных. Сбор данных с конкретного юридического сайта. Обработка и анализ собранных данных (6 ч.)	Чтение документации по BeautifulSoup и requests. Выполнение домашних заданий по сбору данных с различных сайтов (6 ч.)
Создание простых веб-приложений (ведение в Flask или Django для создания веб-приложений. Применение веб-приложений в юридической практике) (16 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. (2 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 (4 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 (6 ч.)
Разработка телеграмм ботов. Работа с фреймворками PyTelegramBot и Aiogram (16 ч.)	Введение в Telegram-ботов. Основные возможности и применение. Обзор фреймворков для разработки ботов. Сравнение	Создание простого Telegram-бота с PyTelegramBot. Создание простого Telegram-бота с Aiogram. Обработка команд и сообщений. Создание обработчиков для различных команд и	Изучение документации по PyTelegramBot и Aiogram. Ознакомление с официальной документацией и

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
	PyTelegramBot и Aiogram: особенности и преимущества (2 ч)	текстовых сообщений. Работа с клавиатурами и кнопками. Реализация inline-кнопок и кастомных клавиатур. Управление состояниями. Создание многоступенчатого диалога с использованием состояний. Интеграция с внешним API. Создание бота, который предоставляет информацию из внешнего API (например, погода) (6 ч.)	примерами использования (8 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

9. Дронов В.А. Django. Практика создания Web-сайтов на Python: пособие / В.А. Дронов. – СПб.: Издательство «БХВ- Петербург», 2016.

10. 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

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

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

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

12. 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.

Критерии оценивания заданий и/или контрольных вопросов

Типовые практические задания

Задание 1: Веб-приложение для управления юридическими документами

Описание: Разработайте веб-приложение для управления юридическими документами и делами с возможностью визуализации данных и отслеживания статуса дел.

Технологии: Python, Flask, SQLAlchemy.

Требования:

1. Создайте веб-интерфейс с использованием Flask для загрузки и управления юридическими документами.

2. Реализуйте систему аутентификации пользователей для обеспечения безопасности доступа к документам.

3. Используйте SQLAlchemy для работы с базой данных, где будут храниться данные о документах и делах.

4. Реализуйте функции добавления, редактирования и удаления записей о юридических делах, включая информацию о статусе, сроках и связанных документах.

5. Создайте страницу с графиками и диаграммами для визуализации статистики по делам (например, количество открытых и закрытых дел) с использованием Chart.js.

6. Обеспечьте возможность фильтрации данных по категориям дел, статусам и датам.

Ожидаемый результат: Функциональное веб-приложение для управления юридическими документами с удобным интерфейсом и визуализацией данных.

Дополнительно: Добавьте функцию экспорта данных о делах и документах в CSV или PDF формат для дальнейшего использования.

Задание 2: Парсер новостного сайта

Задание: Парсинг новостей в области юриспруденции

Описание: разработайте скрипт для парсинга новостного юридического сайта и сохранения информации о статьях в базу данных.

Технологии: Python, requests, BeautifulSoup, SQLite

Требования:

1. Используйте библиотеку requests для отправки HTTP-запросов к юридическому новостному сайту.

2. Примените BeautifulSoup для парсинга HTML-контента страниц сайта.

3. Извлекайте следующую информацию о каждой статье:

- Заголовок
- Дата публикации
- Автор
- Краткое содержание
- Тематика (например, уголовное право, гражданское право и т.д.)

4. Сохраняйте полученные данные в базу данных SQLite, структурируя таблицы по темам и датам.

5. Реализуйте обработку ошибок при неудачных запросах или парсинге, чтобы скрипт мог продолжать работу при возникновении проблем.

6. Добавьте возможность парсинга нескольких страниц новостей, чтобы собрать более полную информацию.

Ожидаемый результат: скрипт, который автоматически собирает информацию о юридических новостях с заданного сайта и сохраняет её в структурированном виде в базе данных.

Дополнительно: реализуйте функцию поиска по сохранённым новостям с использованием ключевых слов и фильтрацию по тематике и дате публикации.

Задание 3: Телеграм-бот для изучения юридических терминов

Описание: создайте Телеграм-бота, который помогает пользователям изучать юридические термины и понятия через систему флеш-карточек.

Технологии: Python, python-telegram-bot, SQLite

Требования:

1. Используйте библиотеку python-telegram-bot для создания бота.
2. Реализуйте функцию добавления новых юридических терминов и их определений в базу данных.
3. Создайте систему показа случайных терминов пользователю для проверки знаний.
4. Реализуйте отслеживание прогресса пользователя (количество правильных и неправильных ответов).
5. Добавьте команду для просмотра статистики изучения терминов.
6. Реализуйте напоминания о необходимости повторения терминов.

Ожидаемый результат: функциональный Телеграм-бот, помогающий пользователям эффективно изучать юридические термины и понятия.

Дополнительно: добавьте возможность изучения терминов с использованием изображений (например, схемы) или аудио произношения (например, произношение термина).

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

1. Какой из языков используется для разработки front-end?
 - a. JavaScript.
 - b. Java.
 - c. Python.
2. Какой слой в паттерне MVC отвечает за доступ к данным?
 - a. Model.
 - b. View.
 - c. Controller.
3. Какой слой в паттерне MVC отвечает за отображение данных?
 - a. View.
 - b. Model.
 - c. Controller.
4. В какой папке в проекте Django хранятся файлы html-страниц?
 - a. Templates.
 - b. Static.
 - c. Html.
5. Выберите верное утверждение:
 - a. В проектах Django используется встроенная ORM.

- b. Для работы с СУБД в Django требуется использовать стороннюю ORM.
 - c. В Django для работы с БД используются только SQL-запросы.
6. Где необходимо указать связь url и функций Python?
- a. url.py correct.
 - b. settings.py.
 - c. Слой Template.
 - d. Слой View.
7. Выберите верное утверждение:
- a. Проект Django состоит из приложений.
 - b. Приложение Django состоит из проектов.

РАБОЧАЯ ПРОГРАММА ПРАКТИКА

1. Аннотация

Основная цель данной практики — обучить юристов основам программирования на Python и показать, как этот язык может быть использован для автоматизации рутинных задач в юриспруденции. 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 из юридической практики, 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 и 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 (16 часов)			
1. Общие вопросы (ознакомление с предприятием)		Изучение общего технологического цикла производства программного обеспечения (4 ч.).	
2. Практическая часть		Решение практико-ориентированных задач (12 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 представят конкретные примеры задач, которые требуют автоматизации. Это может включать в себя обработку документов, анализ данных или создание отчетов.

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

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:

- Сбор данных из различных источников (например, веб-сайтов или баз данных).
- Обработку и анализ собранных данных.
- Автоматизацию рутинных задач (например, генерацию отчетов или отправку уведомлений).

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 для анализа данных в праве и социальных науках»

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)