

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Андрей Драгомирович Хлуткая
Должность: директор
Дата подписания: 14.08.2026 15:21:14
Уникальный программный ключ:
880f7c07c583b07b775f6604e630281b17c89f42

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и
ГОСУДАРСТВЕННОЙ СЛУЖБЫ при ПРЕЗИДЕНТЕ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

СЕВЕРО-ЗАПАДНЫЙ ИНСТИТУТ УПРАВЛЕНИЯ

Факультет среднего профессионального образования

УТВЕРЖДЕНА
решением цикловой (методической)
комиссии общепрофессиональных
дисциплин и профессиональных
модулей специальностей 09.02.07
Информационные системы и
программирование
Протокол от 31.10.2025 № 2

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ОП.04. Информационные технологии в профессиональной деятельности

Специальность – 09.02.11 Разработка и управление программным обеспечением

Профиль – на базе основного общего образования

Квалификация – программист

Форма обучения – очная

Год набора – 2026

Санкт-Петербург 2025 год

Автор-составитель: Вилков Владислав Евгеньевич, преподаватель ФСПО СЗИУ РАНХиГС.

СОДЕРЖАНИЕ

1. Общие положения	4
1.1. Область применения программы	4
1.2. Место дисциплины в структуре основной профессиональной образовательной программы	4
1.3. Цели и задачи учебной дисциплины	4
1.4. Планируемые результаты обучения по дисциплине	4
2. Структура и содержание дисциплины	11
2.1. Объем учебной дисциплины и виды работ	11
2.2. Тематический план и содержание дисциплины	11
2.3. Регламент распределения видов работ по дисциплине с ДОТ	15
3. Материалы текущего контроля успеваемости и промежуточной аттестации обучающихся	16
3.1. Формы и методы текущего контроля успеваемости обучающихся и промежуточной аттестации.....	16
3.2. Оценочные средства текущего контроля успеваемости обучающихся	17
3.3. Оценочные средства промежуточной аттестации обучающихся	20
4. Методические указания для обучающихся по освоению дисциплины	22
5. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»	22
6. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы	25

1 Общие положения

1.1 Область применения программы

Рабочая программа учебной дисциплины «Информационные технологии в профессиональной деятельности» является частью основной профессиональной образовательной программы по специальности СПО 09.02.11 «Разработка и управление программным обеспечением».

1.2 Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Информационные технологии в профессиональной деятельности» является частью профессиональной подготовки, входит в общепрофессиональный цикл дисциплин. Базируется на таких дисциплинах, как «Информатика», «Операционные системы и среды», «Основы алгоритмизации и программирования» является основополагающей для такой дисциплины, как «Разработка программных модулей».

Дисциплина изучается на 2 курсе в 3 семестре.

1.3 Цели и задачи учебной дисциплины

Цель дисциплины «Информационные технологии в профессиональной деятельности»: формирование компетенций в области применения информационных технологий для решения профессиональных задач, освоение современных цифровых инструментов и развитие навыков их эффективного использования в профессиональной деятельности.

Задачи дисциплины:

- сформировать целостное представление об информационных технологиях как инструментарию профессиональной деятельности: их роли, месте в бизнес-процессах и базовых принципах применения;
- освоить методы поиска, анализа и систематизации профессиональной информации с использованием цифровых ресурсов, включая работу с номенклатурой информационных источников и современными средствами информатизации;
- приобрести практические навыки работы с актуальным программным обеспечением: от базовых офисных инструментов до специализированных цифровых средств автоматизации профессиональной деятельности;

- освоить технологии проектирования и управления базами данных: разработку объектов БД (таблицы, индексы, ограничения), оптимизацию запросов, создание хранимых процедур и триггеров на основе принципов реляционной модели и языка SQL;
- развить компетенции в разработке программного обеспечения: создание модулей на различных языках программирования с применением паттернов проектирования, структур данных и методов оптимизации кода (включая многопоточность, кэширование, управление памятью);
- освоить основы веб-разработки: проектирование клиентской и серверной частей веб-приложений, использование языков разметки, соблюдение стандартов кодирования, вёрстка страниц и разработка пользовательских интерфейсов с учётом технологий клиент-серверного взаимодействия (REST API, SOAP);
- сформировать навыки профессионального саморазвития: определение траекторий карьерного роста, применение современной ИТ-терминологии, оценка жизнеспособности проектных идей, а также освоение базовых знаний в области предпринимательской деятельности, правовой и финансовой грамотности;
- изучить этапы жизненного цикла ИТ-проектов: от планирования и реализации до мониторинга результатов, включая основы проектной деятельности и критерии оценки эффективности решений.

1.4 Планируемые результаты обучения по дисциплине

Перечень компетенций

Код ОК, ПК	Уметь	Знать	Навыки
ОК.02	– определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации; – применять средства информационных технологий для решения профессиональных задач;	– номенклатура информационных источников, применяемых в профессиональной деятельности; – современные средства и устройства информатизации, порядок их применения; – программное обеспечение в профессиональной деятельности, в том	-

	– использовать современное программное обеспечение в профессиональной деятельности.	числе цифровые средства.	
ОК.03	– определять траектории профессионального развития и самообразования. – применять современную научную профессиональную терминологию. – оценивать жизнеспособность проектной идеи.	– возможные траектории профессионального развития и самообразования; – основы предпринимательской деятельности, правовой и финансовой грамотности; – основные этапы разработки и реализации проекта.	-
ПК 1.2	– разрабатывать объекты баз данных (таблицы, индексы, ограничения); – оптимизировать запросы к базе данных для повышения производительности; – разрабатывать хранимые процедуры и триггеры.	– основы реляционной модели данных; – язык SQL и его основные команды; – принципы нормализации баз данных.	– работы с различными объектами базы данных; – оптимизации запросов.
ПК 2.2	– разрабатывать модули программного обеспечения на различных языках программирования; – применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей – проводить анализ и мониторинг производительности приложений	– язык программирования, основные конструкции, синтаксис – паттерны проектирования – структуры данных – принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP – работа с инструментальным программным обеспечением – методы оптимизации кода и алгоритмов	– создание модулей программного обеспечения на различных языках программирования – отладки и тестирования разработанных модулей

		– эффективные алгоритмы и структуры данных для повышения производительности – многопоточность в программных модулях – методы оптимизации сетевых протоколов для ускорения обмена данными – кэширование данных – управление памятью – техники повышения производительности программного обеспечения	
ПК 3.2	– разрабатывать программный код клиентской и серверной части веб-приложений; – использовать язык разметки страниц веб-приложения – оформлять код программы в соответствии со стандартом кодирования	– языки программирования и разметки для веб-разработки; – принципы работы объектной модели веб-приложений. – технологии клиент-серверного взаимодействия.	– выполнения верстки страниц; – разработки интерфейса пользователя.

В результате освоения учебной дисциплины студент должен:

иметь практический опыт	– работы с современными средствами информатизации и программным обеспечением в профессиональной деятельности; – поиска, анализа и систематизации профессиональной информации с использованием цифровых ресурсов; – разработки и оптимизации объектов баз данных; – создания и отладки программных модулей на различных языках программирования; – применения паттернов проектирования и структур данных; – разработки клиентских и серверных частей веб-приложений; – использования систем контроля версий и средств совместной разработки; – работы с облачными сервисами и инструментами разработчика; – обеспечения информационной безопасности при разработке программных продуктов.
уметь	– определять задачи для поиска информации, планировать процесс поиска и выбирать необходимые источники информации; – применять средства информационных технологий для решения профессиональных задач; – использовать современное программное обеспечение в профессиональной деятельности; – разрабатывать объекты баз данных (таблицы, индексы, ограничения);

	– оптимизировать запросы к базе данных с целью повышения производительности; – разрабатывать хранимые процедуры и триггеры; – создавать модули программного обеспечения на различных языках программирования; – отлаживать и тестировать разработанные программные модули; – применять паттерны проектирования при разработке программного обеспечения; – разрабатывать клиентскую и серверную части веб-приложений; – использовать языки разметки и программирования в процессе веб-разработки; – оформлять программный код в соответствии с установленными стандартами кодирования.
знать	– номенклатуру информационных источников, применяемых в профессиональной деятельности; – современные средства и устройства информатизации, порядок их применения; – программное обеспечение, используемое в профессиональной деятельности, включая цифровые средства; – основы реляционной модели данных; – язык SQL и его основные команды; – принципы нормализации баз данных; – языки программирования и технологии, применяемые для реализации программных модулей; – паттерны проектирования и структуры данных; – методы оптимизации кода и алгоритмов; – языки программирования и разметки, используемые в веб-разработке; – принципы работы объектной модели веб-приложений; – технологии клиент-серверного взаимодействия.

2 Структура и содержание дисциплины

2.1 Объем учебной дисциплины и виды работ

Виды учебной работы	Объем учебной работы, час.
Учебная нагрузка обучающихся всего, в том числе:	32
лекции	8
практические занятия	22
курсовая работа	-
самостоятельная работа обучающихся	-
консультации	2
промежуточная аттестация	-
Форма промежуточной аттестации	Зачёт с оценкой

2.2 Тематический план и содержание дисциплины

№ п/п	Наименование тем (разделов)	Содержание тем (разделов)	Распределение часов			Форми- руемые компетен- ции	Формы текущего контроля
1	Раздел 1. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ		Л	Пр	СР		
2	Тема 1.1. Искусственный интеллект как инструмент программиста	Содержание учебного материала ИИ и LLM: зачем они нужны разработчику. Обзор популярных ИИ-инструментов (GitHub Copilot, ChatGPT, Codeium). ИИ и написание кода: кейсы и ограничения. Использование ИИ для генерации тестов, SQL-запросов. Промпт-инжиниринг: формулировка запросов. Ревью кода с ИИ: плюсы и минусы. Генерация документации к проекту. ИИ в CI/CD пайплайнах (оптимизация шагов). ChatOps: использование ботов в командной разработке. Этические аспекты и ответственность при работе с ИИ. Практические занятия: Подключение и использование ChatGPT для генерации кода Генерация автотестов на Python по описанию задачи Написание SQL-запросов через Copilot Рефакторинг кода с объяснением шагов Генерация комментариев к функциям и классам Сравнение работы нескольких ИИ-инструментов Создание readme-файла проекта через ИИ Написание GitHub Action с подсказками Copilot Превращение баг-репорта в список задач Разработка промптов для сложных запросов	2	4	-	ОК.02, ПК.1.2, ПК.2.2	Т, ПЗ, О
3	Тема 1.2. Git и Markdown в командной разработке	Содержание учебного материала Контроль версий: зачем нужен Git. Git: базовые команды, концепция веток. Ветки, мержи, pull request и конфликты. GitHub/GitLab: интерфейс, CI, багтрекеры. Markdown: синтаксис, структура, назначение. Документирование API в Markdown.	2	6	-	ОК.02, ПК.2.2, ПК.3.2	Т, ПЗ,О

		README.md как витрина проекта. Использование GitHub Pages и Wiki. Рецензирование кода через pull request. Практика оформления задач и описаний. Практические занятия: Создание и клонирование репозитория Ведение истории коммитов и работа с ветками Конфликт и его разрешение Настройка CI в GitHub Actions Создание красивого README.md Использование маркдауна для changelog Описание API-интерфейса в markdown Работа с pull request и ревью кода Создание и публикация проекта на GitHub Pages Создание вики-проекта и структуры документации					
4	Тема 1.3. Облачные сервисы и инструменты разработчика	Содержание учебного материала Основы работы с облаками: IaaS, PaaS, SaaS. Яндекс Облако / VK Cloud / Selectel: обзор и интерфейс. Хранилище, вычисления, базы данных в облаке. Развёртывание приложения на облачном сервере. Terraform / IaC: автоматизация инфраструктуры. GitLab CI/CD + облако. Облачные IDE (Replit, GitHub Codespaces). S3-хранилище и автоматизация бэкапов. Логирование и мониторинг в облаке. Безопасность облачных сред. Практические занятия: Регистрация и запуск виртуальной машины в Яндекс.Облаке Развёртывание Python-приложения на облачном сервере Использование S3-хранилища для логов Настройка CI/CD-пайплайна для загрузки файлов Подключение к облачной базе данных Использование облачной IDE для командного проекта Создание YAML-манифеста Terraform Настройка доступа к bucket'у	2	6	-	ОК.03, ПК.2.2, ПК.3.2	Т, ПЗ, О

		Интеграция с логами и алертами Аудит безопасности облачного проекта					
5	Тема 1.4. Цифровые инструменты и экосистема разработчика	Содержание учебного материала IDE, расширения, сборщики: VS Code, JetBrains. Bash и командная строка как инструмент. Утилиты curl, wget, ping, telnet. Форматы данных: JSON, YAML, XML. Конфигурационные файлы и шаблоны. DevTools в браузере и веб-отладка. Task-менеджеры и трекеры: Trello, YouTrack. Работа с docker-образами. Инструменты тестирования API: Postman. Автоматизация повседневных задач. Практические занятия: Работа в VS Code: настройка расширений Написание bash-скрипта для автоматизации Отправка API-запроса через curl и Postman Разбор JSON-структуры и валидация Написание dockerfile и сборка образа Использование DevTools для анализа сайта Создание задачи и доски в Trello Отладка API на реальном сервисе Настройка git hooks и lint-автоматизации Создание шаблона конфига в YAML	1	4	-	ОК.02, ПК.2.2, ПК.3.2	Т, ПЗ, О
6	Тема 1.5. Кибербезопасность и цифровая гигиена ИТ-специалиста	Содержание учебного материала Угрозы в разработке: инъекции, XSS, MITM. Безопасные пароли, ключи, доступы. Работа с .env-файлами и секретами. Проверка зависимостей: Snyk, Dependabot. Шифрование, хеширование и токены. VPN, SSH и туннелирование. Анонимизация и защита данных. Правила цифровой гигиены и GDPR. Атаки на open-source проекты. Повседневная безопасность в DevOps.	1	2	-	ОК.02, ПК.1.2, ПК.2.2	Т, ПЗ, О

		Практические занятия: Настройка SSH-ключей и безопасного подключения Работа с .env-файлом в проекте Сканирование зависимостей с Snyk Пример XSS-атаки и защита от неё Хеширование строки и проверка целостности Шифрование данных с помощью openssl Работа с GitHub Secrets и CI Создание VPN-соединения Формирование чек-листа цифровой гигиены Анализ утечек и проверка паролей					
		Итого часов:	8	22	-		

2.3. Регламент распределения видов работ по дисциплине с ДОТ

Данная дисциплина реализуется с применением дистанционных образовательных технологий (ДОТ). Распределение видов учебной работы, форматов текущего контроля представлены в Таблице 2.3.

Таблица 2.3. — Распределение видов учебной работы и текущей аттестации

Вид учебной работы	Формат проведения
Лекционные занятия	Частично с применением ДОТ
Практические занятия	Частично с применением ДОТ
Текущий контроль	Частично с применением ДОТ
Промежуточная аттестация	Контактная аудиторная работа
Формы текущего контроля	Формат проведения
Тестирование	Частично с применением ДОТ
Опрос	Контактная аудиторная работа
Практические задания	Частично с применением ДОТ

Доступ к системе дистанционных образовательных программ осуществляется каждым обучающимся самостоятельно с любого устройства на портале: <https://szu-de.ganepa.ru>, в соответствии с их индивидуальным паролем и логином к личному кабинету/профилю.

Текущий контроль, проводимый в системе дистанционного обучения, оцениваются как в системе дистанционного обучения, так и преподавателем вне системы.

Доступ к материалам лекций предоставляется в течение всего семестра по мере прохождения освоения программы. Доступ к каждому виду работ и количество попыток на выполнение задания предоставляется ограниченное время согласно регламенту дисциплины, опубликованному в системе дистанционного обучения. Преподаватель оценивает выполненные обучающимися работы не позднее 14 рабочих дней после окончания срока выполнения.

3 Материалы текущего контроля успеваемости и промежуточной аттестации обучающихся

3.1 Формы и методы текущего контроля успеваемости и промежуточной аттестации обучающихся

Формы текущего контроля успеваемости:

Опрос (О) метод проведения устный или письменный, позволяет выявить правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, степень развития логического мышления.

Тестирование(Т) письменный метод проверки с предлагаемыми вариантами ответов.

Практическое задание (ПЗ) используется для закрепления теоретических знаний и отработки навыков и умений, способности применять знания при решении конкретных задач, используется практический метод проведения.

Критерии оценивания текущего контроля и промежуточной аттестации:

Оценки «отлично» заслуживает студент, обнаруживший глубокое знание материала, умение свободно выполнять задания, понимающий взаимосвязь основных понятий темы;

Оценки «хорошо» заслуживает студент, обнаруживший полное знание материала; успешно выполняющий предусмотренные задания; и допустивший незначительные ошибки: неточность фактов, стилистические ошибки;

Оценки «удовлетворительно» заслуживает студент, обнаруживший знания основного материала в объеме, необходимом для дальнейшего изучения дисциплины. Справляющийся с выполнением заданий; допустивший погрешности в ответе, но обладающий необходимыми знаниями для их устранения под руководством преподавателя;

Оценки «неудовлетворительно» заслуживает студент, обнаруживший существенные пробелы в знании основного материала; не справляющийся с выполнением заданий, допустивший серьезные погрешности в ответах, нуждающийся в повторении основных разделов курса под руководством преподавателя.

Критерии оценивания тестирования.

Оценки «отлично» заслуживает студент, если он ответил правильно на 90-100% вопросов теста;

Оценки «хорошо» заслуживает студент, если он ответил правильно на часть вопросов 75%-89%;

Оценки «удовлетворительно» заслуживает студент, если он правильно ответил часть вопросов 50%-74%;

Оценки «неудовлетворительно» заслуживает студент, если он правильно ответил менее чем на 50% вопросов или не представлен тест на проверку

Формы текущего контроля

№ п/п	Название темы	Формы текущего контроля успеваемости
1	Тема 1.1. Искусственный интеллект как инструмент программиста	Т, ПЗ, О
2	Тема 1.2. Git и Markdown в командной разработке	Т, ПЗ, О
3	Тема 1.3. Облачные сервисы и инструменты разработчика	Т, ПЗ, О
4	Тема 1.4. Цифровые инструменты и экосистема разработчика	Т, ПЗ, О
5	Тема 1.5. Кибербезопасность и цифровая гигиена ИТ-специалиста	Т, ПЗ, О

Примечание. В столбце «Форма текущего контроля успеваемости, промежуточной аттестации» перечисляются все используемые в учебном процессе по данной дисциплине формы контроля освоения материала. (Т – тестирование; ПЗ – практическое задание, О – опрос).

3.2 Оценочные средства текущего контроля успеваемости обучающихся

Примеры заданий для устного опроса

Вопросы для устного опроса.

Тема 1.1. Искусственный интеллект как инструмент программиста

1. Перечислите 3-4 конкретных сценария, в которых программист может эффективно использовать LLM-инструменты (например, GitHub Copilot или ChatGPT) в повседневной работе, и кратко поясните пользу каждого сценария.
2. Сравните GitHub Copilot и Codeium: назовите два сходства и два различия между этими инструментами, касающиеся их функционала и особенностей применения в разработке.
3. Опишите два типичных ограничения ИИ при написании кода, которые могут привести к ошибкам или неоптимальному решению; приведите по одному примеру для каждого ограничения.
4. Объясните, как можно использовать ИИ для ревью кода: перечислите два преимущества такого подхода и один существенный риск, требующий внимательного контроля со стороны разработчика.
5. Сформулируйте чёткий промпт для генерации автотеста на Python к функции, которая принимает список чисел и возвращает их среднее арифметическое; включите в промпт требования к покрытию кейсов (как минимум три разных сценария) и стилю кода.

Примеры типовых заданий для практических заданий

Тема 1.2. Git и Markdown в командной разработке.

1. Работа с GitHub Secrets и CI

Задание: создайте репозиторий на GitHub и настройте GitHub Actions для сборки проекта. Используйте GitHub Secrets для хранения конфиденциальных данных (например, API-ключей).

Требования:

- создайте рабочий процесс (workflow) с использованием секретов;
- убедитесь, что секреты не отображаются в логах CI.

Тема 1.5. Кибербезопасность и цифровая гигиена ИТ-специалиста

1. Создание VPN-соединения

Задание: настройте VPN-соединение (например, с использованием OpenVPN или WireGuard). Опишите шаги процесса и объясните, как VPN защищает данные при передаче.

Требования:

- используйте локальный сервер или облачный сервис для развёртывания VPN;
- проверьте подключение через VPN, убедившись в шифровании трафика.

Тема 1.5. Кибербезопасность и цифровая гигиена ИТ-специалиста

1. Формирование чек-листа цифровой гигиены

Задание: составьте чек-лист из 10–15 пунктов, который поможет сотрудникам соблюдать цифровую гигиену. Включите в него правила работы с паролями, электронной почтой, социальными сетями и другими аспектами.

Требования:

- разделите чек-лист на категории (например, «Пароли», «Электронная почта», «Социальные сети»);
- предложите меры по внедрению чек-листа в компании.

Примеры тестовых заданий

Часть 1. Задания с выбором одного правильного ответа

1. Что такое Git?

а) Система контроля версий

- b) Язык программирования
- c) Текстовый редактор
- d) Операционная система

Часть 2. Задания на установление соответствия

2. Соотнесите инструменты разработки с их назначением:

Инструмент:

- 1. VS Code
- 2. Git
- 3. GitHub
- 4. Docker

Назначение:

- A. Система контроля версий
- B. Облачный сервис хранения
- C. Среда разработки
- D. Система контейнеризации

Часть 3. Задания на последовательность действий

3. Установите правильную последовательность этапов работы с Git:

- a) Создание репозитория
- b) Добавление файлов
- c) Коммит изменений
- d) Создание ветки
- e) Мердж веток

Часть 4. Ситуационные задачи

4. Опишите процесс настройки CI/CD пайплайна для автоматического тестирования и деплоя веб-приложения.

3.3 Оценочные средства по дисциплине для промежуточной аттестации

Вопросы для подготовки к устному зачету с оценкой:

- 1. Что такое искусственный интеллект и как он применяется в работе программиста?
- 2. Какие современные ИИ-инструменты существуют для разработчиков?
- 3. Как использовать Copilot для ускорения процесса написания кода?

4. В чем заключаются ограничения применения ИИ при генерации кода?
5. Как правильно формулировать запросы к ИИ-системам для получения оптимального результата?
6. Какие техники улучшения качества ответов ИИ существуют?
7. Что такое система контроля версий Git и для чего она нужна?
8. Основные команды Git и их практическое применение
9. Как работать с ветками в Git и разрешать конфликты слияния?
10. Что такое pull request и как им правильно пользоваться?
11. Синтаксис и основные возможности Markdown
12. Как правильно оформлять README.md файл проекта?
13. Принципы документирования API в Markdown
14. Что такое GitHub Pages и как его использовать?
15. Основные модели облачных услуг (IaaS, PaaS, SaaS)
16. Сравнение популярных облачных платформ
17. Как развернуть и настроить приложение в облачной среде?
18. Принципы работы с облачными хранилищами данных
19. Что такое DevOps и как настроить CI/CD в облаке?
20. Как организовать мониторинг и логирование в облачной среде?
21. Настройка и использование VS Code для разработки
22. Работа с командной строкой и bash-скриптами
23. Форматы данных: JSON, YAML, XML - особенности и применение
24. Основы работы с Docker для контейнеризации приложений
25. Инструменты для тестирования API и их применение
26. Как использовать task-менеджеры в процессе разработки?
27. Основные угрозы информационной безопасности в разработке
28. Защита от XSS-атак и SQL-инъекций
29. Как безопасно работать с секретами и ключами в проекте?
30. Принципы цифровой гигиены разработчика
31. Что такое VPN и как его настроить для безопасной работы?
32. Как правильно работать с .env файлами?
33. Инструменты для сканирования зависимостей проекта
34. Принципы хеширования и шифрования данных
35. Как настроить безопасное подключение через SSH?
36. Что такое Git hooks и как их использовать?

37. Принципы работы с GitHub Secrets
38. Как организовать автоматизацию тестирования в CI/CD?
39. Основы работы с Postman для тестирования API
40. Как настроить систему контроля доступа в проекте?
41. Принципы работы с Dockerfile и создание образов
42. Как организовать версионирование в Git?
43. Что такое Git rebase и когда его использовать?
44. Принципы работы с GitHub Actions
45. Как организовать совместную работу над проектом в Git?
46. Основы работы с облачными базами данных
47. Как организовать бэкапирование в облачной среде?
48. Принципы работы с контейнерами в облаке
49. Как организовать масштабирование приложений в облаке?
50. Современные тенденции в области информационной безопасности разработки

4. Методические указания для обучающихся по освоению дисциплины

Приступая к изучению дисциплины «Информационные технологии в профессиональной деятельности», студент должен ознакомиться с содержанием данной «Рабочей учебной программы дисциплины» с тем, чтобы иметь четкое представление о своей работе.

В первую очередь необходимо уяснить цель и задачи изучаемой дисциплины, оценить объем материала, познакомиться с предложенной и подобрать основную и дополнительную литературу, выявить наиболее важные проблемы, стоящие по вопросам изучаемой дисциплины.

Выполнение заданий осуществляется в соответствии с учебным планом и программой. Они должны выполняться в соответствии с методическими рекомендациями, выданными преподавателем, и представлены в установленные преподавателем сроки.

Работая с учебниками и учебными пособиями, целесообразно законспектировать тот материал, который не сообщался студентам на лекциях.

На занятиях лекционного и практического характера студентам для работы требуется тетрадь для записи лекций и заданий.

Для успешного овладения программой дисциплины необходимо выполнять следующие требования:

- посещать все лекционные и практические занятия;

- все рассматриваемые на лекциях и практических занятиях темы и вопросы обязательно фиксировать в тетради;
- в случае пропуска занятий по каким-либо причинам необходимо обязательно самостоятельно изучать соответствующий материал в Moodle, фиксируя записи в тетради, а также выполнять практические задания.

Подготовка к зачету с оценкой осуществляется по представленным в списке основной и дополнительной литературе. Рекомендуемые литература и интернет-ресурсы будут полезны при выполнении практических заданий и для подготовки к тестированиям.

Методические рекомендации по составлению конспекта

Конспект — сложный способ изложения содержания книги или статьи в логической последовательности. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта. Выделите главное, составьте план, представляющий собой перечень заголовков, подзаголовков, вопросов, последовательно раскрываемых затем в конспекте.

Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

При оформлении конспекта необходимо стремиться к емкости каждого предложения.

Методические рекомендации по составлению опорного конспекта

Опорный конспект — вид внеаудиторной самостоятельной работы студента по созданию краткой информационной структуры, обобщающей и отражающей суть материала лекции, темы учебника.

Опорный конспект — это наилучшая форма подготовки к ответу на вопросы.

Основная цель опорного конспекта — облегчить запоминание. Этапы составления опорного конспекта:

1. Изучить материалы темы, выбрать главное и второстепенное;
2. Установить логическую связь между элементами темы;
3. Представить характеристику элементов в краткой форме;
4. Выбрать опорные сигналы для акцентирования главной информации и отобразить в структуре работы.

Методические рекомендации по прохождению тестирования

Тестирование — это исследовательский метод, который позволяет выявить уровень знаний, умений и навыков, способностей, а также их соответствие определенным нормам усвоения, путем выполнения испытуемым ряда специальных заданий.

Следует понимать, что тестовые задания могут быть представлены в различных формах:

- задания закрытой формы, в которых обучающийся выбирает один или несколько правильных ответов из заданного набора:
- задания на дополнение (открытые задания), требующие самостоятельного получения ответов:
- задания на установления соответствия (с множественным выбором), выполнение которых связано с выявлением соответствия между элементами нескольких множеств:
- задания на установление правильной последовательности, в которых от учащегося требует указать порядок действий или процессов и другие. Этапы подготовки к тестированию:

1. Внимательно прочитайте материал по конспекту, составленному на учебном занятии. Прочитайте тот же материал по учебнику, учебному пособию.

2. Постарайтесь разобраться с непонятным, в частности новыми терминами и конструкциями.

3. Ответьте на контрольные вопросы для самопроверки, имеющиеся в учебнике, конспекте и т. д.

4. Кратко перескажите содержание изученного материала «своими словами».

5. Выучите определения основных понятий, условные обозначения, формулы и конструкции.

Подготовка к практическим занятиям

В ходе подготовки к практическим занятиям необходимо изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях, ознакомиться с программным обеспечением. Следует дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой.

Заканчивать подготовку следует закреплением материала с использованием соответствующих программных продуктов.

Все практические задания, предусмотренные рабочей программой, представлены в фонде оценочных средств по дисциплине.

Критерии оценивания выполненных практических работ:

- правильность выполнения работы (отсутствие фактических, логических и других ошибок);

- полнота выполнения работы;
- своевременность выполнения;
- правильность оформления отчета.

За задания, выполненные позже установленного срока или с нарушениями требований к оформлению, оценка на балл снижается.

Порядок организации самостоятельной работы студентов

Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Целью самостоятельной работы студентов является: овладение практическими знаниями, профессиональными умениями и навыками деятельности по специальности, опытом творческой, исследовательской деятельности.

Самостоятельная работа студентов предполагает:

- самостоятельный поиск ответов и необходимой информации в рамках изучаемых тем;
- выполнение заданий для самостоятельной работы, в том числе тестов;
- изучение теоретического и лекционного материала, а также основной и дополнительной литературы при подготовке к практическим занятиям.

5. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»

Основная литература

1. Зубова, Е. Д. Информационные технологии в профессиональной деятельности : учебное пособие для СПО / Е. Д. Зубова. — 4-е изд., стер. — Санкт-Петербург : Лань, 2025. — 212 с. — ISBN 978-5-507-52598-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/455726> (дата обращения: 12.11.2025). — Режим доступа: для авториз. пользователей

2. Ловцов, В. А. Информационные технологии в профессиональной деятельности : учебно-методическое пособие / В. А. Ловцов. — Тамбов : ТГУ им. Г.Р.Державина, 2025. — 118 с. — ISBN 978-5-00078-900-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/504499> (дата обращения: 12.11.2025). — Режим доступа: для авториз. пользователей

3. Федотов, Г. В. Информационные технологии в профессиональной деятельности / Г. В. Федотов. — Санкт-Петербург : Лань, 2024. — 136 с. — ISBN 978-5-507-48044-9. —

Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/362834> (дата обращения: 12.11.2025). — Режим доступа: для авториз. Пользователей

Дополнительная литература

1. Официальный сайт GitHub: — документация и ресурсы по Git и GitHub
2. Microsoft Developer Network: — техническая документация и учебные материалы
3. Oracle Technology Network: — ресурсы по базам данных и облачным технологиям
4. Docker Hub: — репозиторий Docker-образов и документация
5. Stack Overflow: — сообщество разработчиков
6. GitHub Education: — образовательные ресурсы
7. Microsoft Learn: — обучающие материалы по технологиям Microsoft
8. Google Developers: — ресурсы по разработке Google
9. AWS Documentation: — документация Amazon Web Services
10. Yandex Cloud Documentation: — документация Yandex Cloud

Интернет-ресурсы

1. Официальный сайт GitHub: — документация и ресурсы по Git и GitHub
2. Microsoft Developer Network: — техническая документация и учебные материалы
3. Oracle Technology Network: — ресурсы по базам данных и облачным технологиям
4. Docker Hub: — репозиторий Docker-образов и документация
5. Stack Overflow: — сообщество разработчиков
6. GitHub Education: — образовательные ресурсы
7. Microsoft Learn: — обучающие материалы по технологиям Microsoft
8. Google Developers: — ресурсы по разработке Google
9. AWS Documentation: — документация Amazon Web Services
10. Yandex Cloud Documentation: — документация Yandex Cloud

Нормативно-техническая документация:

1. ГОСТ Р 51898-2002 — Основные положения по безопасности информационных технологий
2. ГОСТ Р ИСО/МЭК 27001-2021 — Системы менеджмента информационной безопасности
3. ГОСТ Р 59795-2021 — Требования к содержанию документов
4. Федеральный закон № 152-ФЗ — О персональных данных
5. Федеральный закон № 187-ФЗ — О безопасности критической информационной инфраструктуры
6. PCI DSS — Стандарт безопасности данных платежных карт
7. GDPR — Общий регламент по защите данных
8. ISO/IEC 27001:2013 — Системы менеджмента информационной безопасности
9. ISO/IEC 27002:2013 — Практические правила управления информационной безопасностью
10. RFC 5322 — Стандарт формата электронных сообщений

6 Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Для реализации дисциплины необходимы:

Лаборатория информационных технологий и операционных систем:

- рабочие станции (персональные компьютеры) с характеристиками не ниже: процессор — Intel Core i3 (или аналогичный AMD), ОЗУ — 16 ГБ, SSD — не менее 256 ГБ;
- проекционное оборудование (проектор/интерактивная доска) для демонстрации материалов;
- сетевое подключение со скоростью не менее 100 Мбит/с.

Программное обеспечение:

Операционная система (РЕД ОС 8.0 или аналог)

ПО для просмотра документов в формате PDF (Atril или аналог)

ПО для архивации (Engrampa или аналог)

ПО офисный пакет (Программный пакет Р7-Офис. Профессиональный (десктопная версия), Программный пакет LibreOffice или аналоги)

ПО веб-браузер (Яндекс Браузер, Chromium, Google Chrome или аналоги)

ПО редактор диаграмм (P7-Графика, draw.io или аналоги)

ПО Системы контроля версий (Git, GitKraken или аналоги)

Программная платформа (.NET, Java SE Development Kit, Anaconda3 или аналоги)

ПО среда разработки (JetBrains Rider, Microsoft Visual Studio Professional, PyCharm Professional Edition, IntelliJ IDEA Ultimate, JetBrains WebStorm, Eclipse IDE for Java или аналоги)

Среда для разработки графических интерфейсов (Kivy Designer, Qt Designer или аналоги)

Текстовый редактор (Sublime Text, Visual Studio Code или аналоги)

Клиент для работы с API (Postman или аналог)

ПО СУБД (JetBrains DataGrip, DBeaver Community, PgAdmin, MySQL Workbench или аналоги)

Электронно-библиотечные системы (ЭБС):

1. Электронно-библиотечная система «IPR SMART» (СПО) — URL: <https://www.iprbookshop.ru/>;
2. Электронно-библиотечная система «Лань» — URL: <https://e.lanbook.com/>;
3. Электронно-библиотечная система «iBooks» — URL: <https://ibooks.ru/>;
4. Электронно-библиотечная система «ZNANIUM» — URL: <https://znanium.ru/>.