

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«КУЗБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Т. Ф. ГОРБАЧЕВА»
Филиал КузГТУ в г. Белово

Кафедра инженерно-экономическая

МДК. 03.01 Моделирование и анализ программного обеспечения
Методические рекомендации
по выполнению самостоятельных работ
для специальности
09.02.07 «Информационные системы и программирование»

Составитель: Витвицкий М.Н.
Рассмотрены и утверждены на
заседании кафедры
Протокол № 4 от 06.12.2025 г.
Рекомендовано учебно-
методической комиссией
специальностей СПО в качестве
электронного издания для
использования в учебном процессе
Протокол № 4 от 11.12.2025 г.

Белово 2025

СОДЕРЖАНИЕ

ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	1
ПЛАНИРОВАНИЕ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....	4
КОНТРОЛЬ РЕЗУЛЬТАТОВ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	6
МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ	7
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	20

ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа обучающихся может рассматриваться как организационная форма обучения, обеспечивающих управление учебной деятельностью или деятельность обучающихся по освоению общих и профессиональных компетенций, знаний и умений учебной и научной деятельности без посторонней помощи.

В учебном процессе выделяют два вида самостоятельной работы: аудиторная, внеаудиторная.

Аудиторная самостоятельная работа по учебной дисциплине и профессиональному модулю выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется учащимся по заданию преподавателя, но без его непосредственного участия.

Самостоятельная работа обучающихся проводится с целью:

систематизации и закрепления полученных теоретических знаний и практических умений студентов;

углубления и расширения теоретических знаний;

формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;

развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности и организованности;

формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

развития исследовательских умений;

формирования общих и профессиональных компетенций.

ПЛАНИРОВАНИЕ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Преподавателем учебной дисциплины эмпирически определяются затраты времени на самостоятельное выполнение конкретного содержания учебного задания: на основании наблюдений за выполнением учащимися аудиторной самостоятельной работы, опроса студентов о затратах времени на то или иное задание, хронометража собственных затрат на решение той или иной задачи с внесением поправочного коэффициента из расчета уровня знаний и умений обучающихся.

При разработке рабочей программы по учебной дисциплине или профессиональному модулю при планировании содержания внеаудиторной самостоятельной работы преподавателей устанавливается содержание и объем теоретической учебной информации или практических заданий, которые выносятся на внеаудиторную самостоятельную работу, определяются формы и методы контроля результатов.

Содержание внеаудиторной самостоятельной работы определяется в соответствии с рекомендуемыми видами заданий согласно программе учебной дисциплины профессионального модуля.

Видами заданий для внеаудиторной самостоятельной работы могут быть:

- для овладения знаниями:** компетентностно-ориентированные задание, чтение текста (учебника, первоисточника, дополнительной литературы); составление плана текста; графическое изображение структуры текста; конспектирование текста; реферирование текста; выписки из текста; работа со словарями и справочниками, ознакомление с нормативными документами; учебно-исследовательская работа; использование аудио- и видеозаписей, компьютерной техники и Интернета и др.;

- для закрепления и систематизации знаний:** компетентностно-ориентированное задание, работа с конспектом лекции (обработка текста); повторная работа над учебным материалом (учебника, первоисточника, дополнительной литературы, аудио- и видеозаписей); составление плана и тезисов ответа; составление таблиц для систематизации учебного материала;

изучение нормативных материалов; ответы на контрольные вопросы; аналитическая обработка текста (аннотирование, рецензирование, реферирование, контент-анализ и др.); подготовка сообщений к выступлению на семинаре, конференции; подготовка рефератов, докладов; составление библиографии, тематических кроссвордов; тестирование и др.;

•для формирования компетенций: компетентностно-ориентированное задание, решение задач и упражнений по образцу; решение вариативных задач и упражнений; выполнение чертежей, схем; выполнение расчетно-графических работ; решение ситуационных педагогических задач; подготовка к деловым играм; проектирование и моделирование разных видов и компонентов профессиональной деятельности; подготовка курсовых работ; опытно-экспериментальная работа; упражнения на тренажере; упражнения спортивно-оздоровительного характера; рефлексивный анализ профессиональных умений с использованием аудио- и видеотехники и др.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать специфику специальности, изучаемой дисциплины, индивидуальные особенности студента.

При предъявлении видов заданий на внеаудиторную самостоятельную работу рекомендуется использовать дифференцированный подход к студентам. Перед выполнением студентами внеаудиторной самостоятельной работы преподаватель проводит инструктаж по выполнению задания, который включает цель задания, его содержание, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. В процессе инструктажа преподаватель предупреждает обучающихся о возможных типичных ошибках, встречающихся при выполнении задания. Инструктаж проводится преподавателем за счет объема времени, отведенного на изучение дисциплины.

Самостоятельная работа может осуществляться индивидуально или группами обучающихся в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности уровня умений обучающихся.

Отчет по самостоятельной работе обучающихся предоставляется в электронном виде.

КОНТРОЛЬ РЕЗУЛЬТАТОВ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Контроль результатов внеаудиторной самостоятельной работы студентов может осуществляться в пределах времени, отведенного на обязательные учебные занятия по дисциплине и внеаудиторную самостоятельную работу обучающихся по дисциплине, может проходить в письменной, устной или смешанной форме, с представлением продукта деятельности учащегося.

В качестве форм и методов контроля внеаудиторной самостоятельной работы обучающихся могут быть использованы, *зачеты, тестирование, самоотчеты, контрольные работы, защита творческих работ и др., которые могут осуществляться на учебном занятии или вне его (например, оценки за реферат).*

Критериями оценки результатов внеаудиторной самостоятельной работы обучающегося являются:

- уровень освоения учащимся учебного материала;
- умение обучающегося использовать теоретические знания при выполнении практических задач;
- сформированность общих и профессиональных компетенций;
- обоснованность и четкость изложения ответа;
- оформление материала в соответствии с требованиями.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

РЕФЕРАТ

Реферат (от латинского – сообщаю) – краткое изложение в письменном виде содержания научного труда (трудов), литературы по теме. Это самостоятельная научно-исследовательская работа, где раскрывается суть исследуемой проблемы, изложение материала носит проблемно-тематический характер, показываются различные точки зрения, а также собственные взгляды на проблему. Содержание реферата должно быть логичным.

Критерии оценки реферата:

- соответствие теме;
- глубина проработки материала;
- правильность и полнота использования источников;
- оформление реферата.

ДОКЛАД

Доклад – вид самостоятельной работы обучающихся, используется в учебных и внеклассных занятиях, способствует формированию навыков исследовательской работы, расширяет познавательные интересы, приучает практически мыслить. При написании доклада по заданной теме следует составить план, подобрать основные источники. Работая с источниками, попытаться систематизировать полученные сведения, сделать выводы и обобщения. В настоящее время в учебных заведениях доклады содержательно практически ничем не отличаются от рефератов. Структура и оформление доклада такое же, как в реферате.

Критерии оценки доклада:

- соответствие теме;
- глубина проработки материала;
- правильность и полнота использования источников;
- оформление доклада.

Самостоятельная работа состоит из 2 заданий:

1. Теоретическое задание (реферат, доклад на 10 стр. А4);
2. Практическое задание (выдается преподавателем индивидуально согласно перечню).

Оформление работы

На титульном листе посередине его записывается вид работы, ниже на 10 мм – её название строчными буквами, справа в нижнем углу – фамилия автора разработки, группа. В нижней части титульного листа посередине указывается год написания разработки.

При наборе рекомендуется использовать основные системные гарнитуры шрифта TimesNewRoman. Текст набирается с соблюдением следующих правил: не допускаются ручной набор нумерации в главах и абзацах (только автонумерация); два и более пробела между символами. При наборе должны различаться тире и дефисы; маркеры и другие знаки должны быть сохранены аналогичными на протяжении всего материала. Между инициалами и после них (перед фамилией) ставится неразрывный пробел.

Размеры полей «обычное»: верхнее 1 см, левое 2 см, нижнее 1 см, правое 1 см. Нумерация страниц – внизу «по центру» шрифтом 12 пт. гарнитуры шрифта TimesNewRoman, нумерация страниц записи сквозная, причем начинается простановка номеров со страницы «Содержание», с учетом всех впереди стоящих страниц, на которых номера не проставляются.

Темы самостоятельной работы

№ раздела (темы)	Вопросы, выносимые на самостоятельное изучение	Количество часов
Тема 3.1.1. Задачи и методы моделирования и анализа программных продуктов.	Самостоятельная работа №1 «Сравнительный анализ офисных пакетов»	4
	Самостоятельная работа №2 «Сравнительный анализ браузеров»	4
Тема 3.1.2 Организация ревьюирования. Инstrumentальные средства ревьюирования.	Самостоятельная работа 3. Анализ структуры заданного программного модуля	6

Описание выполнения задания

Самостоятельная работа 1. «Сравнительный анализ офисных пакетов»

Задание 1. Критерии сравнения и матрица функций.

Сформируйте список критериев (не менее 20), сгруппировав их по разделам:

Совместимость: форматы (DOCX, XLSX, PPTX, ODT/ODS/ODP, PDF), корректность импорта/экспорта, макросы.

Функциональность: текст, таблицы, презентации, макросы/VBA, формулы, сводные таблицы, диаграммы, шаблоны, стили, средства рецензирования.

Совместная работа: одновременное редактирование, комментарии, отслеживание изменений, история версий.

Интеграции: облака, почта, календари, коннекторы (Teams/Slack/Zoom), BI, CMS.

Кроссплатформенность: Windows/macOS/Linux, iOS/Android, веб.

Производительность: время открытия/сохранения, размер файлов, потребление памяти/CPU.

Безопасность и управление: шифрование, DLP, права доступа, SSO, офлайн-доступ, политики.

Стоимость и лицензирование: подписка/перпетуальная, ТСО, условия для образования/НКО.

Удобство: интерфейс, локализация, доступность (вкл. скринридеры), офлайн-возможности.

Поддержка и экосистема: документация, комьюнити, плагины/надстройки.

Составьте матрицу сравнения для 3–5 пакетов (например, Microsoft 365, Google Workspace, LibreOffice, WPS Office, OnlyOffice), отметьте наличие/качество по шкале 0–3 и краткие примечания.

Задание 2. Практический тест на совместимость документов

Подготовьте 3 эталонных файла:

Документ: сложные стили, оглавление, сноски, поля, комментарии, отслеживание изменений, таблицы, изображения, формулы.

Таблица: сводные, сложные формулы (в т.ч. массивы), условное форматирование, диаграммы, проверка данных.

Презентация: нестандартные шрифты, анимации, встроенное видео, диаграммы, мастер-слайды.

Откройте/сохраните каждый файл в выбранных пакетах, затем вновь откройте в «эталоне».

Оцените:

Степень сохранения форматирования (0–5).

Сохранность функций (макросы, сводные, анимации) (0–5).

Дельта размера файла и заметные артефакты.

Сформируйте выводы о практической совместимости.

Задание 3. Тест производительности и ресурсоемкости.

Методика:

Замерьте время открытия/сохранения/экспорта в PDF для каждого эталонного файла (3 прогона, усреднить).

Измерьте потребление RAM/CPU во время работы (5 минут активного редактирования).

Тест устойчивости: длительная сессия 1 час с многозадачностью (переключение окон, прокрутка, поиск/замена).

Результаты сведите в таблицу, укажите конфигурацию ПК/сети и версии ПО.

Задание 4. Сценарии совместной работы

Смоделируйте 3 сценария:

Совместное редактирование документа с комментариями и режимом правок (3 участника).

Таблица с одновременным вводом, защитой диапазонов, уведомлениями о правках.

Презентация с задачами для соавторов и историей версий.

Оцените:

Удобство (по шагам), конфликты версий, задержки синхронизации.

Гранулярность прав (только просмотр/комментирование/редактирование, защита).

Экспорт итогов (PDF, ссылки, встраивание), ревью-воркфлоу.

Задание 5. Анализ интеграций и расширяемости

Проверьте:

Поддерживаемые облачные хранилища и SSO (Google, Microsoft, Apple, AD/LDAP, OAuth).

Надстройки/плагины: наличие магазинов, API/SDK, макросы (VBA/JS), автоматизация (Apps Script, Power Automate).

Интеграции с коммуникациями и задачами (Teams, Slack, Notion, Trello, Jira).

Составьте карту интеграций для каждого пакета и примеры 2–3 рабочих автоматизаций.

Задание 6. Юзабилити-аудит.

Выполните типовые задачи и оцените:

Обучаемость: время на поиск функций (стили, таблица, диаграмма, комментарии, гиперссылки).

Согласованность интерфейса между приложениями (Docs/Sheets/Slides vs Word/Excel/PowerPoint и т.д.).

Доступность: навигация с клавиатуры, читабельность, поддержка скринридеров, контрастность.

Итог: перечень проблем/находок и 5 предложений по улучшениям.

Задание 7. Экономический расчет ТСО.

Для каждого пакета рассчитайте:

Стоимость владения на 1 и 3 года для 20 пользователей: лицензии, хранилище, надстройки, обучение, поддержка.

Риски и скрытые издержки: миграция, простой из-за несовместимости, безопасность/инциденты.

Вывод: сравнительная диаграмма и окупаемость с учетом сценариев использования.

Задание 8. Политики безопасности и соответствия.

Сопоставьте:

Шифрование данных на клиенте/сервере, управление ключами, DLP.

Сертификаты и соответствие (ISO 27001, SOC 2, GDPR, HIPAA при наличии).

Управление устройствами, контроль доступа, аудит, eDiscovery, хранение версий/журналов.

Подготовьте чек-лист требований вашей организации и отметьте соответствие.

Задание 9. Кейс миграции.

Исходные данные: компания 50 человек, 30 ТБ документов, активное использование макросов в Excel, мобильная работа.

Задание:

Предложите целевой пакет и план миграции (этапы, инструменты, пилот, обучение).

Оцените риски (макросы, совместимость, права доступа, ИБ) и меры их снижения.

KPI успеха: скорость, ошибок конвертации, удовлетворенность пользователей.

Задание 10. Итоговый обработка этапов выполнения анализа.

Сформируйте:

Резюме с баллами по критериям, сильные/слабые стороны.

Рекомендации по выбору для разных сценариев: образование, SMB, предприятие, онлайн-среда, бюджетное решение.

Приложения: матрицы, замеры, чек-листы, кейсы.

Критерии оценки преподавателем:

Полнота критериев и корректность методики.

Достоверность экспериментов (повторяемость, фиксация версий).

Аргументация выводов и прозрачность расчетов.

Самостоятельная работа №2 «Сравнительный анализ браузеров»

Задание 1. Критерии сравнения и матрица функций.

Сформируйте список критериев (не менее 20), сгруппировав по разделам:

Совместимость веб-стандартов: HTML5/CSS3, ES202x, WebAssembly, WebGL/WebGPU, Service Workers, PWAs.

Производительность: cold/warm start, время загрузки страниц, рендеринг, JS-бенчмарки, потребление RAM/CPU, энергопотребление.

Безопасность: изоляция сайтов (Site Isolation), песочницы, список блокировки вредоносных сайтов, HTTPS-Only, обновления, контейнеры/профили, анти-фишинг/анти-трекер.

Конфиденциальность: блокировка трекеров и сторонних куки, fingerprinting-защита, профили приватного просмотра, режимы DNS over HTTPS/QUIC, встроенный VPN/Proxy.

Функциональность: синхронизация (закладки, пароли, вкладки, история), менеджер паролей, чтение PDF, переводы, чтение вслух, коллекции/онотки, таб-группы.

Совместная работа и экосистема: аккаунты/профили, шэкинг вкладок/сессий, облачные сервисы, отправка страниц/скриншоты.

Расширяемость: поддержка WebExtensions/Manifest V3, магазины дополнений, совместимость с Chrome Web Store.

Удобство: UI/UX, жесты, кастомизация панели, боковые панели, рабочие пространства, быстродействие на слабых ПК.

Кроссплатформенность: Windows/macOS/Linux, iOS/Android, синхронизация между устройствами.

Администрирование: политики для организаций, GPO/MDM, контроль обновлений, отчётность.

Ресурсоёмкость и фоновая активность: автосон вкладок, троттлинг, экономия батареи.

Специфика: встроенный VPN/AdBlock, крипто-кошельки, ИИ-фици, чтение новостей.

Составьте матрицу сравнения (0–3: отсутствует/базово/хорошо/отлично) с примечаниями.

Задание 2. Методика и подготовка тестов.

Подготовьте тестовый набор:

Набор страниц: новостные сайты, SPA на React/Vue, WebGL/WebGPU-демо, видео 4K YouTube, PWA (например, Twitter/X, Notion), страницы с большим числом изображений.

Профили: чистый профиль и «реальный» профиль (с 10–20 расширениями/закладками/историей).

Сценарии: холодный старт, восстановление сессии 20 вкладок, параллельная работа с 10 вкладками, видео-конференция (Meet/Teams), онлайн-редактор (Docs/Office).

Зафиксируйте версии браузеров и ОС, характеристики устройства и сети.

Задание 3. Тест производительности.

Проведите измерения (по 3 прогона, усреднить):

Время старта (cold/warm).

Время загрузки и интерактивности страниц (Largest Contentful Paint, Time to Interactive — можно через встроенные инструменты производительности).

JS-производительность (например, Speedometer/JetStream/ARES-6 — по возможности).

Рендеринг 2D/3D (WebGL/WebGPU демо, FPS).

Память и CPU: при 10/20/50 вкладках в покое и под нагрузкой; экспорт графиков.

Энергопотребление/долговечность батареи (на ноутбуке, 60 минут сценария).

Сведите результаты в таблицы/графики, укажите методику измерений.

Задание 4. Совместимость и корректность отображения.

Проверка веб-стандартов:

Запустите тесты совместимости (например, Can I use — вручную по ключевым фичам; дополнительно: демо с Web Components, CSS Grid/Subgrid, HDR-видео).

Оцените корректность отрисовки сложных макетов, анимаций, шрифтов.

PWA и онлайн:

Установка PWA, работа онлайн, пуш-уведомления, фоновые синки.

Итог: карта совместимости и проблемные кейсы.

Задание 5. Безопасность и приватность (практика).

Проверьте:

Настройки приватности: блокировка сторонних куки, очистка данных, анти-трекер, изоляция сайтов.

DNS over HTTPS/QUIC, HTTPS-Only, предупреждения о смешанном контенте.

Режимы контейнеров/профилей (например, Firefox Multi-Account Containers).

Поведение «Инкогнито/Приватного» режима: что сохраняется/нет.

Выполните тест на трекинг/отпечаток браузера (например, через популярные демо-сайты) и сравните результаты.

Оцените частоту и механизм автообновлений.

Задание 6. Расширения и совместимость с экосистемами.

Сравните:

Поддержку WebExtensions/Manifest V3, доступ к API.

Доступность каталогов расширений, совместимость с Chrome Web Store.

Топ-5 критичных расширений (блокировщик рекламы, менеджер паролей, переводчик, заметки, devtools) — наличие и стабильность.

Протестируйте 5–10 расширений на стабильность и влияние на

производительность.

Задание 7. UX/юзабилити-аудит.

Выполните типовые задачи и измерьте:

Создание/перемещение/поиск вкладок, группировка и восстановление закрытых.

Работа с закладками и читательским режимом.

Скриншоты/запись страницы, перевод, чтение PDF.

Удобство настроек: поиск параметров, профили, разрешения сайтов.

Оцените доступность: навигация с клавиатуры, масштабирование, контрастность, чтение экранными дикторами.

Итог: список проблем/находок и 5 предложений улучшений.

Задание 8. Сценарии синхронизации и многопрофильности.

Проверьте:

Синхронизацию закладок/паролей/вкладок/истории/расширений между устройствами (ПК+мобильный).

Разделение рабочих/личных профилей, гостевой режим, передача вкладок между устройствами.

Оцените скорость, конфликты, контроль над данными (сквозное шифрование, где доступно).

Задание 9. Администрирование и корпоративное применение.

Изучите:

Политики управления (GPO/MDM), шаблоны админ-настроек, контроль обновлений и расширений.

Инструменты развертывания и портативные сборки.

Логи/аудит, изоляция сайтов для корпоративных доменов, режимы совместимости с устаревшими движками (например, IE-режим в Edge).

Сформируйте чек-лист требований вашей организации и отметьте соответствие браузеров.

Задание 10. Экономический и операционный анализ.

Рассчитайте ТСО для 1 и 3 лет на 50 рабочих мест:

Стоимость поддержки/обучения, простой из-за несовместимости, стоимость расширений (платные), риски безопасности.

Время, затрачиваемое ИТ на администрирование и инциденты.

Сформируйте диаграмму сравнения и выводы о целесообразности.

Задание 11. Кейс-задание «Выбор браузера для компании».

Исходные данные: 200 сотрудников, 20 критичных веб-сервисов (включая старую ERP на IE-совместимом движке), строгие требования ИБ, ноутбуки на Windows/macOS, часть — с ARM.

Задание:

Предложите целевой набор браузеров (возможно, 2) и политику их применения.

План развертывания, миграции профилей/закладок, ограничения расширений, обновления.

Риски (совместимость, плагины, политика куки/SSO) и меры снижения.

KPI: стабильность, процент успешных сценариев, число инцидентов, удовлетворённость пользователей.

Критерии оценки преподавателем:

Полнота критериев и корректность методики.

Достоверность экспериментов (повторяемость, фиксация версий).

Аргументация выводов и прозрачность расчетов.

Самостоятельная работа №3 «Анализ структуры заданного программного модуля»

Задание 1: Анализ модуля аутентификации (Python/Псевдокод.

Дан модуль `auth.py`, отвечающий за вход пользователя в систему. Оценить связность (cohesion) и зацепление (coupling), выявить структурные проблемы.

```
python
# Фрагмент модуля auth.py
def login(username, password):
    user = db_query("SELECT * FROM users WHERE username = ?",
(username,)) # Прямой запрос к БД
    if not user:
        return "Пользователь не найден"
    if hash_password(password) != user['password_hash']: # Хеширование
внутри функции
        return "Неверный пароль"
    session['user_id'] = user['id'] # Прямая работа с сессией
    log_activity(f"User {username} logged in") # Логирование внутри
    return "Успешный вход"
def hash_password(password): # Вспомогательная функция внутри модуля
    return hashlib.sha256(password.encode()).hexdigest()
```

```

1. Разбейте модуль на компоненты.

Выделите основные ответственности (например: проверка данных, работа с БД, хеширование, управление сессией, логирование).

2. Оцените связность (Cohesion).

Является ли модуль высокосвязным? Все ли функции относятся к одной задаче?

Ответьте: Какие функции нарушают принцип единственной ответственности (SRP)? Почему?

3. Оцените зацепление (Coupling).

С какими внешними системами/модулями взаимодействует `login()` (БД, сессия, логгер)? Является ли зацепление сильным (tight) или слабым (loose)? Обоснуйте.

4. Предложите рефакторинг. Как можно выделить ответственности в отдельные модули/классы? Предложите новую структуру (например: `UserRepository` для БД, `PasswordHasher`, `SessionManager`, `ActivityLogger`).

5. Напишите псевдокод улучшенной функции `login()`, использующей зависимости через параметры или внедрение.

Критерии оценки.

Точность выделения ответственостей.

Глубина анализа нарушения SRP и сильного зацепления.

Практичность предложений по рефакторингу.

Корректность псевдокода после изменений.

**Задание 2: Анализ цикломатической сложности модуля обработки заказов (Java/Псевдокод).**

Дан метод `processOrder(Order order)` в модуле `OrderService.java`.

Рассчитать цикломатическую сложность, выявить сложные для тестирования участки.

```java

// Фрагмент OrderService.java

public String processOrder(Order order) {

 if (order == null || order.getItems().isEmpty()) { // Условие 1

 throw new InvalidOrderException("Пустой заказ");

}

 for (Item item : order.getItems()) { // Цикл 1

 if (item.getStockCount() <= 0) { // Условие 2 (внутри цикла)

 if (!inventory.restockItem(item.getId(), 10)) { // Условие 3

(вложенное)

 throw new InventoryException("Не удалось пополнить " +
 item.getName());

 }

}

 inventory.reserveItem(item.getId(), 1); // Действие

}

 if (paymentGateway.charge(order.getTotal(), order.getPaymentToken())) { // Условие 4

```

        order.setStatus(Status.PAID);
        notificationService.sendEmail(order.getUserEmail(), "Заказ оплачен"); // Зависимость
    }
    return "Заказ успешно обработан";
} else {
    order.setStatus(Status.PAYMENT_FAILED);
    return "Ошибка оплаты";
}
```

```

1. Рассчитайте цикломатическую сложность (СС).

Используйте формулу: ' $CC = E - N + 2P$ ', где 'E' - рёбра графа потока управления, 'N' - узлы, 'P' - компоненты связности (обычно 1 для метода).

Или подсчитайте: \* '1' (начало) + количество условий ('if', 'for', 'while', 'case') + 1.

Укажите итоговое значение СС.

2. Оцените результат. Какое значение СС считается высоким (обычно  $>10$ )? Превышает ли метод этот порог? Какие последствия у высокой сложности (тестируемость, читаемость, поддержка)?

3. Выделите проблемные участки. Перечислите блоки кода (по условиям/циклам), которые сложнее всего протестировать изолированно. Почему?

4. Предложите упрощение. Как можно разбить метод на более мелкие (например, 'validateOrder()', 'reserveInventory()', 'processPayment()')? Предложите 1-2 конкретных изменения для снижения СС.

Критерии оценки.

Правильность расчета цикломатической сложности.

Понимание влияния высокой СС на качество кода.

Точность выявления сложных для тестирования фрагментов.

Эффективность предложений по декомпозиции.

**Задание 3: Анализ зависимостей и поиск "Божественного объекта" в модуле отчетов (C#/Псевдокод).**

Дан класс 'ReportGenerator.cs', формирующий различные отчеты.

Построить граф зависимостей, выявить нарушение принципов модульности.

```

```C#
// Фрагмент ReportGenerator.cs
public class ReportGenerator {
    private DatabaseConnector _dbConnector; // Зависимость 1: Работа с БД
    private ExcelExporter _excelExporter; // Зависимость 2: Экспорт в Excel
    private PdfRenderer _pdfRenderer; // Зависимость 3: Рендер PDF
    private EmailService _emailService; // Зависимость 4: Отправка email

    public void GenerateSalesReport(DateTime from, DateTime to, string format) {
        var data = _dbConnector.GetSalesData(from, to); // Исп. Зав.1
        if (format == "Excel") {
            var file = _excelExporter.Export(data, "Sales"); // Исп. Зав.2
        }
    }
}

```

```

        _emailService.Send("sales@company.com", "Отчет", file); // Исп.
Зав.4
    } else if (format == "PDF") {
        var file = _pdfRenderer.Render(data); // Исп. Зав.3
        _emailService.Send("sales@company.com", "Отчет", file); // Исп.
Зав.4
    }
}

public void GenerateInventoryReport(string format) {
    var data = _dbConnector.GetInventoryData(); // Исп. Зав.1
    // ... Аналогичная логика выбора формата и отправки ...
}
// ... Еще 5 методов для разных отчетов ...
}
...

```

1. Постройте граф зависимостей.

Изобразите схему (можно текстом/списком), показывающую, какие внешние сервисы ('DatabaseConnector', 'ExcelExporter' и т.д.) использует класс 'ReportGenerator'.

Укажите, сколько разных ответственостей реализует сам 'ReportGenerator' (получение данных, преобразование в формат, отправка).

2. Выявите "Божественный объект" (God Object).

Обладает ли класс 'ReportGenerator' слишком многими обязанностями?

Какие признаки "Божественного объекта" здесь присутствуют (размер, количество зависимостей, несвязанные функции)?

3. Проанализируйте устойчивость.

Как изменение в 'EmailService' (например, новый параметр в 'Send()') повлияет на 'ReportGenerator'?

Нарушает ли класс принцип открытости/закрытости (OCP)? Почему?

4. Предложите новую структуру.

- . Как разделить ответственности? (Например: 'ReportDataProvider', 'ReportFormatter' (с подклассами 'ExcelFormatter', 'PdfFormatter'), 'ReportSender').

- . Как 'ReportGenerator' должен взаимодействовать с этими новыми компонентами (использовать интерфейсы)?

Критерии оценки.

Точность построения графа зависимостей.

Глубина анализа признаков "Божественного объекта".

Понимание нарушения OCP и влияния изменений.

Практичность предложенной декомпозиции и снижения связанности.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Основная литература

1. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для среднего профессионального образования / Е. А. Черткова. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 146 с. — (Профессиональное образование). — ISBN 978-5-534-18094-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563828>.

Дополнительная литература

2. Казанский, А. А. Программирование на C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 181 с. — (Профессиональное образование). — ISBN 978-5-534-21380-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/569863>.

3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : Учебное пособие / Г. Н. Федорова. — Москва : КУРС : ИНФРА-М, 2022. — 336 с. (Среднее профессиональное образование). — ISBN 9785906818416. — Текст : непосредственный.

4. Гниденко, И. Г. Технология разработки программного обеспечения : учебник для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563151>.

Составитель
Витвицкий Максим Николаевич

Методические указания по выполнению самостоятельной работы
для студентов очной формы обучения
по направлению специальности
09.02.07 «Информационные системы и программирование»

Публикуется в авторской редакции