

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Соловьев Дмитрий Александрович  
Должность: ректор ФГБОУ ВО Вавиловский университет  
Дата подписания: 26.05.2026 15:46:10  
Уникальный программный ключ:  
528682d78e671e566ab07601c11a2172f735a12



**Министерство сельского хозяйства Российской Федерации**  
**Федеральное государственное бюджетное образовательное**  
**учреждение высшего образования**  
**«Саратовский государственный университет генетики,**  
**биотехнологии и инженерии имени Н.И. Вавилова»**  
**Финансово-технологический колледж**

## **ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

### **для проверки сформированности компетенций**

Междисциплинарный курс	<b>Моделирование и анализ программного обеспечения</b>
Профессиональный модуль	<b>ПМ.03 Ревьюирование программных продуктов</b>
Специальность	<b>09.02.07 Информационные системы и программирование</b>
Квалификация выпускника	<b>Специалист по информационным системам</b>
Нормативный срок обучения	<b>2 года 10 месяцев</b> <b>(на базе среднего общего образования)</b>
Форма обучения	<b>Очная</b>

**Разработчики:** преподаватель Зыбина О.В.  (подпись)

**Саратов 2024**

## ОГЛАВЛЕНИЕ

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы .....	3
2. Сценарии выполнения заданий.....	4
3. Система оценивания выполнения заданий.....	5
4. Описание дополнительных материалов и оборудования, необходимых для выполнения заданий.....	5
5. Задания для проверки уровня сформированности компетенций с указанием типа заданий (с ключами к оцениванию заданий).....	6

## 1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы (ОП)

В результате изучения профессионального модуля 03.01 «Моделирование и анализ программного обеспечения» (профессиональный цикл дисциплин) обучающиеся, в соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование, приказом Министерства образования и науки РФ от 9 декабря 2016 года N 1547 (квалификация – специалист по информационным системам), формируют следующие компетенции, указанные в таблице:

Код компетенции	Наименование компетенции	Этапы формирования компетенции в процессе освоения ОП (семестр)
ОК 01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.	3
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.	3
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках.	3
ПК 3.1	Осуществлять ревьюирование программного кода в соответствии с технической документацией	3
ПК 3.2	Выполнять процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям.	3
ПК 3.3	Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.	3
ПК 3.4	Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.	3

## 2. Сценарии выполнения заданий

№ п/п	Тип задания	Последовательность действий при выполнении задания
<b>1. Задания закрытого типа</b>		
1.1	Задание закрытого типа на установление последовательности	<ol style="list-style-type: none"> <li>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов)</li> <li>2. Внимательно прочитать предложенные варианты ответа)</li> <li>3. Построить верную последовательность из предложенных элементов)</li> <li>4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности без пробелов и знаков препинания (например, БВА или 135).</li> </ol>
1.2	Задание закрытого типа на установление соответствия	<ol style="list-style-type: none"> <li>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов)</li> <li>2. Внимательно прочитать оба списка: список 1 - вопросы, утверждения, факты, понятия и т.д.; список 2 - утверждения, свойства объектов и т.д.</li> <li>3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов)</li> <li>4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А-1 или Б-4).</li> </ol>
<b>2. Задания открытого типа</b>		
2.1	Задание открытого типа с кратким ответом	<ol style="list-style-type: none"> <li>1. Внимательно прочитать текст задания и понять суть вопроса)</li> <li>2. Продумать краткий ответ.</li> <li>3. Записать ответ в виде слова, словосочетания или числа)</li> <li>4. В случае расчетной задачи, записать ответ в виде числа)</li> </ol>
2.2	Задание открытого типа с развернутым ответом	<ol style="list-style-type: none"> <li>1. Внимательно прочитать текст задания и понять суть вопроса)</li> <li>2. Продумать логику и полноту ответа)</li> <li>3. Записать ответ, используя четкие компактные формулировки.</li> <li>4. В случае расчетной задачи, записать решение и ответ.</li> </ol>
<b>3. Задания комбинированного типа</b>		
3.1	Задание комбинированного типа с выбором одного верного ответа из предложенных и обоснованием выбора	<ol style="list-style-type: none"> <li>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов)</li> <li>2. Внимательно прочитать предложенные варианты ответа)</li> <li>3. Выбрать один ответ, наиболее верный.</li> <li>4. Записать только номер (или букву) выбранного варианта ответа)</li> <li>5. Записать аргументы, обосновывающие выбор ответа)</li> </ol>
3.2	Задание комбинированного типа с выбором нескольких	<ol style="list-style-type: none"> <li>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько из</li> </ol>

№ п/п	Тип задания	Последовательность действий при выполнении задания
	верных ответов из предложенных обоснованием выбора	предложенных вариантовВ) 2. Внимательно прочитать предложенные варианты ответаА) 3. Выбрать несколько ответов, наиболее верных. 4. Записать только номера (или буквы) выбранных вариантов ответаА) 5. Записать аргументы, обосновывающие выбор ответаВ)

### 3. Система оценивания выполнения заданий

№ п/п	Указания по оцениванию	Характеристика правильности ответа
<b>1. Задания закрытого типа</b>		
1.1	Задание закрытого типа на установление последовательности считается верным, если правильно указана вся последовательность цифр.	«верно» / «неверно»
1.2	Задание закрытого типа на установление соответствия считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого).	«верно» / «неверно»
<b>2. Задания открытого типа</b>		
2.1	Задание открытого типа с кратким ответом оценивается по следующим критериям: 1) Правильность ответа (отсутствие фактических и грамматических ошибок). 2). Сопоставимость с эталонным ответом в случае расчетной задачи.	«верно» / «неверно»
2.2	Задание открытого типа с развернутым ответом оценивается по следующим критериям. 1) Правильность ответа (отсутствие фактических ошибок). 2) Полнота ответа (раскрытие объема используемых понятий). 3) Обоснованность ответа (наличие аргументов). 4) Логика изложения ответа (грамотная последовательность излагаемого материала). 5. Сопоставимость с эталонным ответом.	«верно» / «неверно»
<b>3. Задания комбинированного типа</b>		
3.1	Задание комбинированного типа с выбором одного верного ответа из предложенных с обоснованием выбора ответа считается верным, если правильно указана цифра (буква) и приведены корректные аргументы, используемые при выборе ответа	«верно» / «неверно»
3.2	Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных с обоснованием выбора ответов считается верным, если правильно указаны цифры (буквы) и приведены корректные аргументы, используемые при выборе ответаА)	«верно» / «неверно»

### 4. Описание дополнительных материалов и оборудования, необходимых для выполнения заданий

Для выполнения заданий не дополнительные материалы и оборудование.

**5. Задания для проверки уровня сформированности компетенций с указанием типа заданий  
(с ключами к оцениванию заданий)**

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
<b>3 семестр</b>			
<b>ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам</b>			
1	<p><i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность цифр слева направо:</i></p> <p>Прочитайте текст и установите последовательность состояний файла при работе с системой контроля версий Git от момента создания до публикации в удаленном репозитории. Запишите соответствующую последовательность цифр слева направо:</p> <p>1) Committed (Зафиксирован в локальном репозитории) 2) 3) Modified (Изменен, но не добавлен в индекс) 4) Pushed (Отправлен в удаленный репозиторий) 5) Staged (Добавлен в индекс/область подготовленных файлов)</p>	Задание закрытого типа на установление последовательности	2413
2	<p><i>Прочитайте текст и установите соответствие:</i></p> <p>Установите соответствие между моделью ветвления в Git и её ключевой характеристикой.</p> <p>А) GitFlow Б) GitHub Flow В) Trunk-Based Development</p>	Задание закрытого типа на установление соответствия	А-2 Б-1 В-3

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
	<p>1) Простая модель, основанная на Pull Requests и короткоживущих feature-ветках. Программа, выполняющаяся в фоновом режиме без интерфейса</p> <p>2) Использует долгоживущие ветки develop и master, а также вспомогательные release/ и hotfix/.Центральная часть ОС, управляющая ресурсами</p> <p>3) Все разработчики коммитят в одну главную ветку (main/trunk), используя короткие lived-ветки.</p>		
3	<p><i>Прочитайте текст, выберите один правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа:</i></p> <p>Кто такой разработчик (программист)?</p> <p>1) Специалист, который тестирует готовое программное обеспечение и ищет в нём ошибки.</p> <p>2) Сотрудник, который проектирует дизайн интерфейсов и рисует макеты сайтов.</p> <p>3) Специалист, который пишет код, создавая алгоритмы и логику работы программ.</p> <p>4) Человек, который настраивает серверы и обеспечивает безопасность сети.</p>	<p>Задание комбинированного типа с выбором одного верного ответа из предложенных и обоснованием выбора</p>	<p>3</p> <p>Разработчик (программист) — это специалист, который занимается непосредственно написанием исходного кода, созданием архитектуры и логики программного обеспечения.</p>
4	<p>Прочитайте текст и запишите ответ в виде термина:</p> <p>Какие две основные операции в Git позволяют «сохранить» изменения: одна — локально, другая — на удалённом сервере?</p>	<p>Задания открытого типа с кратким ответом</p>	<p>Локально — git commit, на удалённый сервер — git push.</p>

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
5	<p><i>Прочитайте текст и запишите развернутый обоснованный ответ:</i></p> <p>Что является основной целью дизассемблирования в процессе обратного проектирования?</p>	Задание открытого типа с развернутым ответом	<p>Преобразование машинного кода (бинарного исполняемого файла) в читаемый ассемблерный код для последующего анализа логики программы.</p> <p>Пояснение: Это первый ключевой шаг, позволяющий понять структуру программы при отсутствии исходного кода.</p>
<b>3 семестр</b>			
<b>ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.</b>			
6	<p><i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность цифр слева направо:</i></p> <p>Расположите этапы сравнительного анализа в правильном порядке:</p> <ol style="list-style-type: none"> <li>1) Представление результатов (отчёт, таблица, рекомендация)</li> <li>2) Сбор и оценка данных по каждому продукту</li> <li>3) Выбор критериев сравнения (функции, цена, поддержка)</li> <li>4) Определение целей анализа (зачем сравниваем?)</li> </ol>	Задание закрытого типа на установление последовательности	4321

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
7	<p><i>Прочитайте текст и установите соответствие:</i></p> <p>Установите соответствие между характеристикой браузера и инструментом/методом её проверки:</p> <p>А) Производительность  Б) Безопасность  В) Совместимость</p> <p>1) Проверка наличия и частоты обновлений, режима изоляции процессов (Sandbox), списков уязвимостей (CVE)  2) Использование встроенных DevTools (вкладка Network, Lighthouse) или специализированных бенчмарков (Speedometer). Проверка наличия и частоты обновлений, режима изоляции процессов (Sandbox), списков уязвимостей (CVE).  3) Тестирование отображения на сайтах-«аквариумах» (например, CanIUse) или на внутренних корпоративных порталах.</p>	<p>Задание закрытого типа на установление соответствия</p>	<p>А-2  Б-1  В-3</p>
8	<p><i>Прочитайте текст, выберите все правильные варианты ответов и запишите аргументы, обосновывающие выбор ответа:</i></p> <p>Какие инструменты и методы относятся к типовым средствам анализа программных проектов?</p> <p>1) Статический анализатор кода (SonarQube)  2) Code review («пристальный взгляд»)  3) Система контроля версий (Git)  4) Диаграмма Ганта  5) Профилировщик (VisualVM)  6) А/В-тестирование кнопок</p>	<p>Задания комбинированного типа с выбором нескольких верных ответов из предложенных и обоснованием выбора</p>	<p>1, 2, 3, 4,5</p> <p><b>Обоснование:</b> Автоматическая проверка кода без запуска (статический анализ).  Ручная проверка кода экспертом (экспертный анализ).  Анализ истории изменений, поиск ошибок (git blame).  Анализ сроков, этапов и зависимостей (проектный анализ).  Анализ производительности запущенного приложения (динамический анализ).</p>

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
9	<p><i>Прочитайте текст и запишите ответ в виде термина:</i></p> <p>Мощный отладчик, поставляемый с Eclipse CDT, является стандартом де-факто для отладки нативных C/C++ приложений в Linux</p>	Задания открытого типа с кратким ответом	GDB (GNU Debugger), интегрированный через интерфейс Eclipse
10	<p><i>Прочитайте текст и запишите развернутый обоснованный ответ:</i></p> <p>Для выявления какого класса уязвимостей особенно полезен статический анализ потоков данных (Data Flow Analysis)?</p>	Задание открытого типа с развернутым ответом	Для выявления инъекций (SQL, командной строки) и уязвимостей, связанных с некорректной обработкой пользовательского ввода, так как он отслеживает, как непроверенные данные проходят через программу.
<b>3 семестр</b>			
<b>ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.</b>			
11	<p><i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность цифр слева направо:</i></p> <p>Поставьте в правильном порядке этапы обратного проектирования простого алгоритма, начиная с готового исполняемого файла:</p> <p>1) Документирование/создание псевдокода.  2) Восстановление логики (анализ управляющих конструкций)  3) Анализ графа вызовов функций  4) Дизассемблирование</p>	Задание закрытого типа на установление последовательности	4132

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
12	<p><i>Прочитайте текст и установите соответствие:</i></p> <p>Установите соответствие между типом анализа и типичным инструментом:</p> <p>A) Valgrind, GDB  Б) SonarQube, Pylint  B) SourceMonitor, Understand</p> <p>1) Статический анализ кода  2) Динамический анализ (профилирование памяти)  3) Измерение метрик (сложность, зацепление)</p>	<p>Задание закрытого типа на установление соответствия</p>	<p>A - 2  Б - 1  B - 3</p>
13	<p><i>Прочитайте текст, выберите один правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа:</i></p> <p>Что такое «механизм pre-commit hook» в контексте контроля внесения изменений в код?</p> <p>1) Специальная ветка в репозитории, предназначенная для фиксации завершённых задач перед слиянием в main.  2) Автоматический скрипт, выполняемый на сервере непрерывной интеграции после отправки изменений в удалённый репозиторий.  3) Клиентский скрипт, запускаемый локально перед созданием коммита для проверки кода на соответствие стандартам, запуска тестов или форматирования.  4) Инструмент для визуального сравнения изменений в коде перед отправкой на ревью коллеге.</p>	<p>Задание комбинированного типа с выбором одного верного ответа из предложенных и обоснованием выбора</p>	<p>3</p> <p><b>Обоснование:</b>  Pre-commit hook — это клиентский (локальный) скрипт Git, который выполняется автоматически до фиксации коммита. Он позволяет разработчику на своей машине проверить код (линтер, тесты, форматирование) и предотвратить коммит, если проверка не пройдена.</p>

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
14	<p><i>Прочитайте текст и запишите ответ в виде термина:</i></p> <p>Какой технический критерий сравнения видео-плееров напрямую влияет на возможность воспроизведения конкретного файла?</p>	Задания открытого типа с кратким ответом	Поддержка видеокодеков и контейнеров (например, H.264, AV1, VP9 в контейнере MP4 или MKV).
15	<p><i>Прочитайте текст и запишите развернутый обоснованный ответ:</i></p> <p>В чём ключевое различие между валидацией кода на стороне разработчика (pre-commit) и на стороне сервера (pre-receive / CI)?</p>	Задание открытого типа с развернутым ответом	Pre-commit хуки работают локально и могут быть проигнорированы (git commit --no-verify), а серверные проверки являются принудительными и не позволяют запустить код, не прошедший требования (линтеры, тесты).

### 3 семестр

#### ПК 3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией

16	<p><i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность цифр слева направо:</i></p> <p>Расположите в правильной последовательности этапы внедрения нового механизма контроля изменений в код (например, обязательного ревью через Pull Request) в существующей команде:</p> <ol style="list-style-type: none"> <li>1) Автоматизация проверок (линтеры, тесты в CI)</li> <li>2) Формализация правил и создание чек-листов</li> <li>3) Пробный запуск на одном проекте и сбор обратной связи</li> <li>4) Выбор инструмента (GitHub, Gerrit, etc.)</li> <li>5) Обучение команды и документация процесса</li> <li>6) Полномасштабное внедрение и аудит</li> </ol>	Задание закрытого типа на установление последовательности	425316
----	---	---	--------

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
17	<p><i>Прочитайте текст и установите соответствие:</i></p> <p>Установите соответствие между этапом разработки и типичным инструментом автоматического ревью/проверки кода, который на этом этапе применяется:</p> <p>А) Написание кода (в IDE)  Б) Локальный коммит  В) Отправка изменений (Push)  Г) Создание Pull Request  Д) Слияние (Merge)</p> <p>1) Статический анализатор в CI/CD (например, SonarQube, Checkstyle).  2) Препроцессор/форматтер кода (например, Prettier, clang-format).  3) Git pre-commit хук (например, запуск линтера или unit-тестов).  4) Встроенный линтер IDE (подсветка ошибок, code completion).  5) Платформа для ревью (GitHub/GitLab PR, Gerrit) с интеграцией CI.</p>	Задание закрытого типа на установление соответствия	А-4 Б-3 В-5 Г-1

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
18	<p><i>Прочитайте текст, выберите все правильные варианты ответов и запишите аргументы, обосновывающие выбор ответа:</i></p> <p>Что такое «динамический анализ кода» в контексте методов исследования программного обеспечения?</p> <ol style="list-style-type: none"> <li>1) Метод исследования исходного кода без его выполнения, основанный на синтаксическом и семантическом анализе.</li> <li>2) Процесс автоматической проверки стиля кода и соответствия стандартам оформления.</li> <li>3) Метод анализа программы, выполняемый в процессе её работы с целью выявления ошибок времени выполнения, утечек памяти и узких мест производительности.</li> <li>4) Ручная проверка кода разработчиками для выявления логических ошибок и улучшения читаемости.</li> </ol>	<p>Задания комбинированного типа с выбором нескольких верных ответов из предложенных и обоснованием выбора</p>	<p>3</p> <p><b>Обоснование:</b> Динамический анализ — это исследование программы во время её выполнения. В отличие от статического анализа, он требует запуска кода и позволяет обнаруживать ошибки, проявляющиеся только в рантайме (деление на ноль, утечки памяти, гонки данных).</p>
19	<p><i>Прочитайте текст и запишите ответ в виде термина:</i></p> <p>Встроенная функция Visual Studio позволяет быстро оценить и улучшить «чистоту» кода проекта, показывая индикаторы и предлагая исправления</p>	<p>Задания открытого типа с кратким ответом</p>	<p>Анализатор кода (Code Analysis) и «Быстрые действия» (Quick Actions) с рефакторингами, доступные по лампочке или отвертке в редакторе.</p>
20	<p><i>Прочитайте текст и запишите развернутый обоснованный ответ:</i></p> <p>Почему при сравнении характеристик кода (метрик), полученных в разных средах</p>	<p>Задание открытого типа с развернутым ответом</p>	<p>Потому что разные инструменты могут использовать различные алгоритмы подсчета (например, что считается строкой кода — пустая? с комментарием?),</p>

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
	программирования (IDE) или разными инструментами, важно учитывать контекст и методологию измерения?		анализировать разные этапы сборки (исходный код, препроцессированный, байт-код) и иметь отличные наборы правил для расчета сложности. Сравнение «сырых» чисел без понимания методики может привести к некорректным выводам о качестве. Важно сравнивать метрики, полученные одним и тем же инструментом с одинаковыми настройками, или делать поправку на методологию.
<b>3 семестр</b>			
<b>ПК 3.2 Выполнять процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям.</b>			
21	<p><i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность цифр слева направо:</i></p> <p>Расположите в правильной последовательности этапы настройки доступа к Git-репозиторию на Linux-сервере для организации процесса ревьюирования с обязательной проверкой кода перед слиянием в основную ветку:</p> <p>1)Настройка pre-receive хука на сервере для проверки GPG-подписей коммитов и запуска CI-тестов</p> <p>2)Создание bare-репозитория на сервере и настройка прав доступа через SSH-ключи</p>	Задание закрытого типа на установление последовательности	423615

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
	<p>3) Определение ролевой модели (администратор, разработчик, рецензент) и создание соответствующих групп в Linux</p> <p>4) Установка Git и необходимых утилит (git-core, openssh-server) на сервер</p> <p>5) Настройка post-receive хука для отправки уведомлений рецензентам о новом Pull Request</p> <p>6) Конфигурация Git-клиентов разработчиков (user.email, GPG-ключи)</p>		
22	<p><i>Прочитайте текст и установите соответствие:</i></p> <p>Установите соответствие между командой и её действием:</p> <p>А) SSH-ключи  Б) Группы Linux  В) Pre-receive hook  Г) GPG-подпись  Д) Umask/ACL  Е) Post-receive hook</p> <p>1) Основа аутентификации в Git over SSH. Позволяет идентифицировать пользователя без пароля.  2) Традиционный способ разграничения прав в Linux: добавление пользователя в группу git-developers даёт право на запись.  3) Выполняется ДО обновления ссылок. Может</p>	Задание закрытого типа на установление соответствия	А-3 Б-4 В-1 Г-2 Д-5 Е-6

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
	<p>отклонить пуш, если не пройдены проверки.</p> <p>4) Подтверждает, что коммит создан именно этим разработчиком и не был изменён.</p> <p>5) Определяют права на новые файлы; ACL даёт более гибкие разрешения, чем классические chmod.</p> <p>6) Выполняется ПОСЛЕ обновления ссылок. Не влияет на принятие кода, но автоматизирует последующие действия.</p>		
23	<p><i>Прочитайте текст, выберите один правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа:</i></p> <p>Что является <b>конечной целью</b> обратного проектирования алгоритма из исполняемого файла?</p> <ol style="list-style-type: none"> <li>1) Получение точной копии исходного кода на языке высокого уровня (C++, Python и т.п.), идентичной оригинальному.</li> <li>2) Восстановление логики работы программы и представление её в форме, понятной человеку (псевдокод, блок-схема, структурированное описание).</li> <li>3) Дизассемблирование файла до полного листинга всех инструкций процессора.</li> <li>4) Извлечение всех строковых констант и ресурсов (изображения, иконки, звуки) из бинарного файла.</li> </ol>	<p>Задание комбинированного типа с выбором одного верного ответа из предложенных и обоснованием выбора</p>	<p>2</p> <p><b>Обоснование:</b> Обратное проектирование алгоритма не ставит целью восстановление <b>исходного кода</b> (это декомпиляция, и она часто невозможна или даёт трудночитаемый результат). Конечная цель — <b>понимание алгоритма</b>: как программа работает, какие решения принимает, какие данные обрабатывает и в какой последовательности.</p>
24	<p><i>Прочитайте текст и запишите ответ в виде термина:</i></p> <p>Механизм Git позволяющий убедиться, что коммит не был изменён после создания, и его автор</p>	<p>Задания открытого типа с кратким ответом</p>	<p>Цифровая подпись коммита с использованием GPG (GNU Privacy Guard). Подпись проверяет криптографическую хеш-сумму коммита и удостоверяет авторство</p>

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
25	<p><i>Прочитайте текст и запишите развернутый обоснованный ответ:</i></p> <p>Что измеряет цикломатическая сложность (McCabe) метода/функции и о чём предупреждает её высокое значение?</p>	Задание открытого типа с развернутым ответом	Измеряет количество линейно независимых путей в графе потока управления функции. Высокое значение (обычно >10) сигнализирует о чрезмерной сложности, что затрудняет тестирование, понимание и поддержку кода.
<b>3 семестр</b>			
<b>ПК 3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.</b>			
26	<p><i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность цифр слева направо:</i></p> <p>Расположите в правильной хронологической последовательности процесс прохождения изменения (фичи) от идеи до попадания в основную ветку при использовании модели <b>GitHub Flow</b>:</p> <ol style="list-style-type: none"> <li>1) Создание Pull Request и назначение рецензентов</li> <li>2) Создание feature-ветки от main</li> <li>3) Развёртывание (деплой) изменений в production</li> <li>4) Code review и утверждение изменений</li> </ol>	Задание закрытого типа на установление последовательности	2143

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
27	<p><i>Прочитайте текст и установите соответствие:</i></p> <p>Установите соответствие между ролью в команде разработки и её основной зоной ответственности в процессе управления изменениями и ревью.</p> <p>А) DevOps-инженер  Б) Tech Lead  В)Reviewer  Г) Contributor</p> <p>1) Проверка кода на соответствие архитектурным стандартам и бизнес-требованиям, утверждение критических изменений.  2) Обеспечение работоспособности CI/CD-пайплайнов, настройка хуков и автоматизации ревью.  3) Непосредственное написание кода, создание коммитов и Pull Request'ов.  4) Анализ кода коллег, выявление дефектов, проверка стиля и логики.</p>	<p>Задание закрытого типа на установление соответствия</p>	<p>А - 2  Б - 1  В - 4  Г - 3</p>
28	<p><i>Прочитайте текст, выберите все правильные варианты ответов и запишите аргументы, обосновывающие выбор ответа:</i></p> <p>Какие из перечисленных практик являются обязательными для успешного внедрения Trunk-Based Development (разработка в основном стволе)? Выберите все подходящие варианты.</p> <p>1) Feature flags (Feature toggles)  2) Долгоживущие ветки develop и release  3) Continuous Integration (CI)</p>	<p>Задания комбинированного типа с выбором нескольких верных ответов из предложенных и обоснованием выбора</p>	<p>1,3,4</p> <p>Trunk-Based Development требует минимального времени жизни веток, автоматизации и скрытия незавершённого кода. Обязательные элементы: feature-флаги (А), Continuous Integration (С), короткие ветки или прямой коммит в trunk (D). Вариант В (долгоживущие ветки) является антипаттерном для Trunk-Based и полностью противоречит его принципам.</p>

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
	4) Короткоживущие feature-ветки		
29	<i>Прочитайте текст и запишите ответ в виде термина:</i>  Утилита JDK позволяющая выполнять профилирование работающего Java-приложения, анализируя использование heap-памяти и CPU	Задания открытого типа с кратким ответом	Java VisualVM или более современный JDK Mission Control (JMC).
30	<i>Прочитайте текст и запишите развернутый обоснованный ответ:</i>  Назовите основное архитектурное отличие распределённых (например, Git, Mercurial) от централизованных (например, Subversion) систем контроля версий.	Задание открытого типа с развернутым ответом	В распределённых VCS каждый участник имеет на локальной машине полную копию репозитория со всей историей, в то время как в централизованных — только рабочую копию файлов, а вся история хранится на едином сервере.
<b>3 семестр</b>			
<b>ПК 3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.</b>			
31	<i>Прочитайте текст и установите последовательность. Запишите соответствующую последовательность цифр слева направо:</i>	Задание закрытого типа на установление последовательности	426153

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
	<p>Расположите в правильной хронологической последовательности этапы обратного проектирования исполняемого файла для восстановления алгоритма его работы:</p> <ol style="list-style-type: none"> <li>1) Анализ графа вызовов и идентификация ключевых функций</li> <li>2) Дизассемблирование — преобразование машинного кода в ассемблерный листинг</li> <li>3) Построение псевдокода или блок-схемы алгоритма</li> <li>4) Загрузка бинарного файла в дизассемблер (IDA Pro, Ghidra, radare2)</li> <li>5) Анализ потоков данных и выявление связей между функциями</li> <li>6) Идентификация строковых констант и импортируемых API-функций</li> </ol>		
32	<p><i>Прочитайте текст и установите соответствие:</i></p> <p>Установите соответствие между инструментом обратного проектирования и его ключевой особенностью или основным назначением:</p> <ol style="list-style-type: none"> <li>А) IDA Pro</li> <li>Б) Ghidra</li> <li>В) x64dbg</li> <li>Г) radare2</li> </ol> <p>1) Бесплатный дизассемблер с открытым исходным кодом и веб-интерфейсом, разработанный Агентством национальной безопасности (NSA) США.</p>	Задание закрытого типа на установление соответствия	<p>А - 2  Б - 1  В - 3  Г - 4</p>

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
	<p>2)Продвинутый статический и динамический анализ, мощный декомпилятор, де-факто стандарт в индустрии (платный).</p> <p>3)Отладчик для Windows, специализирующийся на анализе 64-разрядных приложений в реальном времени.</p> <p>4)Фреймворк для обратного проектирования с консольным интерфейсом, поддерживающий десятки архитектур и платформ.</p>		
33	<p><i>Прочитайте текст, выберите один правильный вариант ответа и запишите аргументы, обосновывающие выбор ответа:</i></p> <p>Что такое «анализ потоков данных» (Data Flow Analysis) в контексте обратного проектирования?</p> <ol style="list-style-type: none"> <li>1) Метод дизассемблирования, при котором каждая инструкция процессора преобразуется в последовательность байтов для дальнейшего анализа.</li> <li>2) Процесс отслеживания перемещения данных от точки их входа в программу (функции, сокет, файл) через преобразования к точке использования с целью выявления уязвимостей и понимания логики обработки.</li> <li>3) Техника визуализации графа вызовов функций, показывающая, какие функции вызывают другие функции в программе.</li> <li>4) Алгоритм сжатия дизассемблированного кода для уменьшения размера файла при передаче между реверс-инженерами.</li> </ol>	Задание комбинированного типа с выбором одного верного ответа из предложенных и обоснованием выбора	<p>2</p> <p><b>Обоснование:</b> Анализ потоков данных — это отслеживание жизненного цикла данных в программе. Ответ В точно описывает эту суть: от источника (вход) через преобразования к потребителю (выход/использование). Это ключевой метод для поиска уязвимостей типа инъекций и переполнения буфера.</p>
34	<i>Прочитайте текст и запишите ответ в виде термина:</i>	Задания открытого типа с кратким	Псевдокод

Номер задания	Формулировка задания	Тип задания	Ключ к оцениванию задания
	Конечный продукт обратного проектирования. Он читается человеком, содержит осмысленные имена переменных и комментарии.	ответом	
35	<p><i>Прочитайте текст и запишите развернутый обоснованный ответ:</i></p> <p>Объясните, что такое декомпиляция и её роль в системе.</p>	Задание открытого типа с развернутым ответом	Процесс преобразования машинного или промежуточного кода (байт-кода) обратно в код на языке высокого уровня, приближённый к исходному.