

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**РАБОЧАЯ ПРОГРАММА
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**
по профессиональному модулю
ПМ.02 Ревьюирование программных продуктов

программы подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование

Форма обучения: *очная*

Находка 2026

Рабочая программа производственной (по профилю специальности) практики по профессиональному модулю ПМ.02 *Ревьюирование программных продуктов* разработана в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности / профессии 09.02.07 *Информационные системы и программирование*, утвержденного приказом Минобрнауки России от «09» декабря 2016 г. № 1547, примерной образовательной программой.

Разработчик(и): *Лихацкий М.Б. преподаватель ОСПО филиала ФГБОУ ВО «ВВГУ» в г. Находке*

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № 10 от «29» июня 2026 г.

Председатель ЦМК _____ В.В. Плахотина

подпись

СОДЕРЖАНИЕ

1. Паспорт программы практики	5
2. Результаты практики	7
3. Структура и содержание практики.....	7
4. Условия реализации программы практики	11
5. Контроль и оценка результатов освоения практики	14

1. ПАСПОРТ ПРОГРАММЫ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ ПП.02.01 ПРОИЗВОДСТВЕННАЯ ПРАКТИКА

1.1. Область применения программы практики

Программа практики является составной частью Программы подготовки специалистов среднего звена, обеспечивающей реализацию ФГОС СПО.

Практика является частью учебного процесса и направлена на формирование у студентов практических профессиональных умений, приобретение первоначального практического опыта по основным видам профессиональной деятельности для последующего освоения ими профессиональных компетенций по избранной специальности:

ПК 3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией

ПК 3.2. Выполнять измерение характеристик компонент программного продукта для определения соответствия заданным критериям

ПК 3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма

ПК 3.4. Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.

1.2. Цели и задачи практики – требования к результатам освоения практики, формы отчетности

В ходе освоения программы практики студент должен:

Иметь практический опыт:

- в измерении характеристик программного проекта; использовании основных методологий процессов разработки программного обеспечения; оптимизации программного кода с использованием специализированных программных средств

Уметь:

- работать с проектной документацией, разработанной с использованием графических языков спецификаций; выполнять оптимизацию программного кода с использованием специализированных программных средств; использовать методы и технологии тестирования и ревьюирования кода и проектной документации; применять стандартные метрики по прогнозированию затрат, сроков и качества

Знать:

- задачи планирования и контроля развития проекта; принципы построения системы деятельности программного проекта; современные стандарты качества программного продукта и процессов его обеспечения

По окончании практики студент сдаёт отчет в соответствии с содержанием тематического плана практики и по установленной форме.

Итоговая аттестация проводится в форме - **дифференцированного зачёта.**

1.3. Количество часов на освоение программы практики

Рабочая программа практики рассчитана на прохождение студентами практики в объеме 51 час.

2. РЕЗУЛЬТАТЫ ПРАКТИКИ

Код и наименование результата обучения
ПК 3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией
ПК 3.2. Выполнять измерение характеристик компонент программного продукта для определения соответствия заданным критериям
ПК 3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма
ПК 3.4. Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием

3. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИКИ

3.1 Тематический план практики

Наименование профессионального модуля	Коды формируемых компетенций	Объем времени, отводимый на практику
1	2	3
ПМ.03 Ревьюирование программных продуктов	ПК 3.1	51 часов
	ПК 3.2	
	ПК 3.3	
	ПК 3.4	
Консультирование		4 часа

3.2. Содержание практики

Наименование разделов и тем	Содержание освоенной учебной информации, виды работ, выносимые на практику в соответствии с рабочими программам профессиональных модулей	Объем часов	Уровень освоения	Коды профессиональных компетенций
1	2	3	4	5
Раздел 1. Организационно-подготовительный этап прохождения практики на предприятии		2	4	
Тема 1.1. Инструктаж по прохождению производственной практики и правилам безопасности работы на предприятии.	<i>Содержание выполняемых работ</i> Знакомство с общими функциональными обязанностями, правилами техники безопасности на предприятии, на конкретном рабочем месте, при работе с электрическими приборами (устройствами)	2		
Раздел 2. Ознакомление со структурой и характером деятельности подразделения		4	4	
Тема 2.1. Ознакомление с организацией работы на предприятии или в структурном подразделении	<i>Содержание выполняемых работ</i> Знакомство с режимом работы, формой организации труда и правилами внутреннего распорядка, структурными подразделениями предприятия, штатным расписанием; с принципами управления, руководства и осуществления должностных обязанностей	2		ПК 3.1-3.4
Тема 2.2. Ознакомление с должностными и функциональными обязанностями	<i>Содержание выполняемых работ</i> Изучение прав и обязанностей сотрудника, должностной инструкции, регламентирующей его деятельность; знакомство с правами и обязанностями других сотрудников и руководителей; согласование с руководителем практики задание, постановку целей и задач практики	2		ПК 3.1-3.4
Раздел 3. Работа на рабочих местах или в подразделениях			4	

предприятия		45		
Тема 3.1. Ознакомление: с организацией информационного обеспечения подразделения; с процессом защиты на уровне проектирования и эксплуатации информационных средств; с методами планирования и проведения мероприятий по созданию (разработке) проекта (подсистемы) информационной среды предприятия для решения конкретной задачи.	<i>Содержание выполняемых работ</i> Ознакомление с производственными характеристиками и показателями деятельности предприятия. Изучение новых технологических средств в современных информационных системах, применяемых на предприятии. Изучение основных проектных решений по информационным системам на предприятии (в организации). Ознакомление с методологией проектирования, внедрения и эксплуатации актуальных информационных систем. Изучение технологии сбора, регистрации и обработки информации на данном предприятии. Проектирование подсистем защиты. Обеспечения защиты информации от несанкционированного доступа	2		ПК 3.1-3.4
Тема 3.2. Изучение структурных и функциональных схем предприятия, организации деятельности подразделения; порядка и методов ведения делопроизводства; требований к техническим, программным средствам, средствам защиты информации используемым на предприятии.	<i>Содержание выполняемых работ</i> Изучение основ финансов, организации денежного обращения и кредитования предприятия, приобретение навыков использования финансово-кредитного механизма с целью повышения эффективности работы предприятия и составления финансового плана. Изучение схем защиты денежных транзакций через сеть интернет Изучение организации расчета заработной платы на предприятии, приобретение навыков проектирования трудовых процессов с учетом комплекса технических, экономических, психофизиологических и социальных факторов, оценка затрат и результатов труда.	2		ПК 3.1-3.4

Тема 3.3. Выполнение производственных заданий	<i>Содержание выполняемых работ</i>			
	Методы организации работы в команде разработчиков. Системы контроля версий Цели, задачи, этапы и объекты ревьюирования. Планирование ревьюирования Цели, корректность и направления анализа программных продуктов. Выбор критериев сравнения. Представление результатов сравнения Примеры сравнительного анализа программных продуктов Цели, задачи и методы исследования программного кода Измерительные методы оценки программ: назначение, условия применения. Корректность программ. Эталоны и методы проверки корректности Метрики, направления применения метрик. Метрики сложности. Метрики стилистики Исследование программного кода на предмет ошибок и отклонения от алгоритма Программные измерительные мониторы Применение отладчиков и дизассемблера (например OllyDbg, WinDbg, IdaPro) Защита программ от исследования Исследование кода вредоносных программ	35		ПК 3.1-3.4
Тема 3.4. Сбор и анализ материалов для оформления отчета	<i>Содержание выполняемых работ</i>			
	Сбор материалов для отчета, подготовка отчетной документации по практике	4		ПК 3.1-3.4
Консультация	Консультирование студентов в процессе прохождения Практики	2		
Итоговая аттестация	Сдача отчета в соответствии с содержанием тематического плана практики и по установленной форме	2		
Экзамен по модулю «ПМ.02 Ревьюирование программных продуктов»	Сдача экзамена в соответствии с содержанием тематического плана и по установленной форме	4		
Всего		57		

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ПРАКТИКИ

4.1 Требования к документации, необходимой для проведения практики.

Для проведения практики в учебном заведении разработана следующая документация:

- рабочая программа практики;
- календарно-тематический план;
- распоряжение о направлении студентов на производственную практику и назначении руководителей практики.
- договоры с предприятиями по проведению практики;
- графики консультаций и контроля за выполнением студентами программы практики;
- график защиты отчетов по практике.

4.2 Требования к учебно-методическому обеспечению практики.

В целях реализации требований к учебно-методическому обеспечению практики разработаны и утверждены:

- Задания на практику;
- Методические рекомендации для студентов по выполнению видов работ на практике;
- Методические рекомендации по формированию отчетов по практике;
- Методические рекомендации по оформлению дневника по практике;
- Критерии оценки прохождения практики и защиты отчетов.

4.3. Требования к материально-техническому обеспечению практики

К рабочим местам выдвигаются следующие требования:

- оснащённость современными персональными электронно-вычислительными машинами (далее — ПЭВМ);
- подключение ПЭВМ к информационно-телекоммуникационной сети «Интернет»;
- оснащённость ПЭВМ программным обеспечением согласно таблице 5 (отдельные элементы могут быть заменены по инициативе практиканта).

Таблица 5 — Программное обеспечение производственной практики

1	2	3
№№	Тип	Наименование или характеристика
1	Webframework	Symfony/Silex
2	Браузер	Основанный на WebKit
3	Операционная система	Ubuntu или другая UNIXподобная
4	Интегрированная среда разработки	NetBeans версии 8 и выше
5	Интерпретатор языка	PHP версии 5.6 и выше
6	Сервер HTTP	The Apache HTTP server версии 2.4 и выше
7	Средство администрирования баз данных	phpMyAdmin версии 4.2 и выше или Adminer версии 4.1.0 и выше
8	Средство для создания	Совместимый с OpenDocument v1.0

	презентаций	(ГОСТР ИСО/МЭК 263002010)
9	Система управления базами данных	MySQL версии 5.6 и выше
10	Текстовый процессор	Совместимый с OpenDocument v1.0 (ГОСТР ИСО/МЭК 263002010)
11	Текстовый редактор	Совместимый с UTF8 без BOM

4.4. Информационное обеспечение обучения.

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

1. Рудаков А. Технология разработки программных продуктов: учебник. / Рудаков А. -Изд.Academia. Среднее профессиональное образование. 2025. 208 стр. <http://www.academia-moscow.ru/reader/?id=401005&demo=Y> ссылка на книгу 2024
2. Методы и средства инженерии программного обеспечения: Учебник. Автор/создатель Лавришева Е.М., Петрухин В.А. Единое окно доступа к образовательным ре-сурсам. <http://window.edu.ru/catalog/pdf2txt/699/41699/18857>

Дополнительные источники

1. Марков, А.С. Статический сигнатурный анализ безопасности программ [Текст]/ А.С.Марков, А.А. Фадин // Программная инженерия и информационная безопасность. – 2025. - № 1(1). С. 50-56.

Рекомендуемые информационные источники

1. www.znanium.com
2. www.biblioclub.ru
3. www.book.ru
4. <https://urait.ru/>

4.5 ТРЕБОВАНИЯ К РУКОВОДИТЕЛЮ ПРАКТИКИ

Руководителем практики от образовательного учреждения назначается педагогический работник, имеющий высшее образование, соответствующее профилю проводимой практики

4.5.1 Руководитель практики от образовательного учреждения:

1. Разрабатывает тематику заданий для студентов;
2. Проводит консультации со студентами перед направлением их на практику с разъяснением целей, задач и содержания практики;
3. Принимает участие в распределении студентов по рабочим местам или перемещении их по видам работ;
4. Осуществляет контроль правильного распределения студентов в период практики; формирует группы в случае применения групповых форм проведения практики;
5. Проводит индивидуальные и групповые консультации в ходе практики;
6. Оказывает методическую помощь студентам при выполнении ими заданий и сборе материалов к отчету по практике;
7. Контролирует выполнение требований охраны труда, безопасности жизнедеятельности и пожарной безопасности;

4.5.2 Руководитель практики от организации:

1. Согласовывает программу практики, планируемые результаты практики, задание на практику;
2. Участвует в организации и проведении дифференцированного зачета по практике, квалификационного экзамена и экзамена по профессиональному модулю;
3. Участвуют в организации и оценке результатов освоения общих и профессиональных компетенций, освоенных студентами в период прохождения практики;
4. Проводит инструктаж студентов по ознакомлению с требованиями охраны труда, безопасности жизнедеятельности и пожарной безопасности

4.6 Требования к соблюдению техники безопасности и пожарной безопасности

Регламентация требований по пожарной безопасности и техники безопасности осуществляется внутренними локальными актами образовательного учреждения и организации, куда был направлен обучающийся.

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРАКТИКИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

По результатам усвоения программы практики студенты представляют руководителю практики от образовательного учреждения: отчет, дневник и аттестационный лист на студента-практиканта от руководителя базы практики.

По окончании практики студент сдает дневник, отчет с дифференцированной оценкой и аттестационный лист руководителю производственной практики от образовательного учреждения. Защита дневников и отчетов проводится в организации или в образовательном учреждении (если группа размещена по разным объектам практики). На базах практики защита должна проводиться в последний день практики.

При оценке итогов работы студента на практике учитываются содержание и правильность оформления студентом дневника и отчета по практике, отзывы руководителей практики от организации, качество ответов на вопросы в ходе защиты отчета.

Зарегистрированные и защищенные дневники, отчеты хранятся в образовательном учреждении в течение трех лет в соответствии с номенклатурой дел.

Аттестация студента по итогам прохождения практики проводится только после сдачи документов по практике и фактической защиты отчета.

Защита отчета по практике, как правило, представляет собой краткий, 8-10-минутный доклад студента и его ответы на вопросы членов комиссии.

После защиты отчета руководитель практики от образовательного учреждения дает свое заключение о заполнении дневника, отчета, выполнении программы практики и ставит по итогам дифференцированную оценку по пятибалльной шкале (5 «отлично», 4 «хорошо», 3 «удовлетворительно», 2 «неудовлетворительно»). Оценка одновременно проставляется в ведомость, зачетную книжку студента, дневник студента по производственной практике и аттестационный лист.

<i>Результаты (освоенные профессиональные компетенции)</i>	<i>Основные показатели оценки результата</i>	<i>Формы и методы контроля и оценки</i>
ПК 3.1 Выполнять построение заданных моделей программного средства с помощью графического языка (обратное проектирование).	Оценка «отлично» - в системе контроля версий выбрана верная версия проекта, проанализированы архитектура и алгоритм проекта на соответствие спецификации, предложен альтернативный вариант решения поставленной задачи в виде описания и/или UML диаграмм; результаты ревью сохранены в системе контроля версий. Оценка «хорошо» - в системе контроля версий выбрана верная версия проекта, проанализированы архитектура или алгоритм проекта на соответствие спецификации, предложен альтернативный вариант решения поставленной задачи в виде описания или UML диаграмм; результаты ревью сохранены в системе контроля версий. Оценка «удовлетворительно» - в системе контроля версий выбрана верная версия проекта, проанализированы архитектура или алгоритм проекта на соответствие спецификации; результаты ревью в виде описания сохранены в системе контроля версий.	<i>Защита отчета Отзыв руководителя практики от организации</i>
ПК 3.2 Выполнять измерение характеристик компонент программного	Оценка «отлично» - определен полный набор качественных характеристик предложенного программного средства с помощью заданного набора метрик в том числе с использованием инструментальных средств; сделан вывод о	<i>Защита отчета Отзыв руководителя практики от организации</i>

продукта для определения соответствия заданным критериям.	соответствии заданным критериям; результаты сохранены в системе контроля версий. Оценка «хорошо» - определен набор качественных характеристик предложенного программного средства с помощью заданного набора метрик в том числе с использованием инструментальных средств; результаты сохранены в системе контроля версий. Оценка «удовлетворительно» - определены некоторые качественные характеристики предложенного программного средства из заданного набора метрик в том числе с использованием инструментальных средств; результаты сохранены в системе контроля версий.	
ПК 3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.	Оценка «отлично» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; программный код проанализирован на соответствие алгоритму; проведена оптимизация и подтверждено повышение качества программного кода; результаты сохранены в системе контроля версий. Оценка «хорошо» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; программный код проанализирован на соответствие алгоритму; проведена оптимизация и оценка качества программного кода. Оценка «удовлетворительно» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; программный код проанализирован на соответствие алгоритму; проведена оценка качества программного кода.	<i>Защита отчета Отзыв руководителя практики от организации</i>
ПК 3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.	Оценка «отлично» - указан набор возможных средств выполнения поставленной задачи, выполнен анализ достоинств и недостатков не менее, чем трех программных продуктов и средств разработки, обоснован выбор одного (возможно, двух и более) из них. Оценка «хорошо» - выполнен анализ достоинств и недостатков двух программных продуктов и средств разработки, обоснован выбор одного из них. Оценка «удовлетворительно» - выполнен анализ достоинств и недостатков программных продуктов и средств разработки, обоснован выбор одного (возможно, двух и более) из них.	<i>Защита отчета Отзыв руководителя практики от организации</i>

Формы и методы контроля и оценки результатов обучения должны позволять проверять у обучающихся не только сформированность профессиональных компетенций, но и развитие общих компетенций и обеспечивающих их умений.

Таблица 7 — Контроль и оценка результатов освоения общих компетенций

Результаты (освоенные общие компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	Демонстрация интереса к будущей профессии	Интерпретация результатов наблюдений за деятельностью, дифференцированный зачет
ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество	Выбор и применение методов и способов решения профессиональных задач в области технологии разработки, сопровождения и продвижения программного обеспечения; оценка эффективности и качества выполнения	Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики; степень самостоятельности при выполнении заданий, дифференцированный зачет
ОК 3. Принимать решения в стандартных и нестандартных ситуациях, нести за них ответственность	Решение стандартных и нестандартных профессиональных задач в области технологии разработки, сопровождения и продвижения программного обеспечения	Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики; степень самостоятельности при выполнении заданий, дифференцированный зачет
ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития	Эффективный поиск необходимой информации; использование различных источников, включая электронные	Анализ полноты, качества, достоверности, логичности изложения найденной информации; интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики, дифференцированный зачет
ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности	Использование информационно-коммуникационных технологий для сопровождения и продвижения программного обеспечения	
ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями	Взаимодействие с обучающимися, преподавателями и мастерами в ходе обучения	Анализ реакции на замечания и предложения разработчиков, руководителя практики; интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики, дифференцированный зачет

ОК 7. Брать на себя ответственность за работу членов команды (подчинённых), результат выполнения заданий	Самоанализ и коррекция результатов собственной работы	Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики, дифференцированный зачет
ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации	Организация самостоятельных занятий	Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики, дифференцированный зачет
ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности	Анализ инноваций в области технологии разработки, сопровождения и продвижения программного обеспечения	Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики, дифференцированный зачет

5.1 Экзаменационные билеты

- Вопрос:** Цели, корректность и направления анализа программных продуктов. Выбор критериев сравнения .

Задание: Составьте таблицу сравнения двух текстовых редакторов по критериям (функциональность, стоимость, удобство интерфейса), обоснуйте выбор наилучшего.
- Вопрос:** Понятие ревьюирования. Виды ревьюирования кода (формальное, неформальное, инспекция) .

Задание: Опишите роль ревьюера в коллективе разработчиков и правила совмещения ролей .
- Вопрос:** Методы организации работы в команде разработчиков. Системы контроля версий .

Задание: Опишите процесс создания и изучения возможностей репозитория проекта .
- Вопрос:** Задачи планирования и контроля развития проекта .

Задание: Составьте план-график ревьюирования программного модуля на 2 недели.
- Вопрос:** Жизненный цикл и организационная структура ИТ-проекта .

Задание: Определите место этапа ревьюирования в жизненном цикле разработки ПО (Waterfall/Agile).
- Вопрос:** Механизмы и контроль внесения изменений в код .

Задание: Опишите порядок работы с pull request в системе Git.
- Вопрос:** Современные стандарты качества программного продукта и процессов его обеспечения .

Задание: Перечислите критерии, которым должен соответствовать «чистый» код.
- Вопрос:** Понятие репозитория проекта. Структура и цели создания .

Задание: Создайте структуру репозитория для проекта «Интернет-магазин» и опишите назначение папок.
- Вопрос:** Методы ревьюирования кода с использованием специализированных программных средств (SonarQube, ESLint) .

Задание: Опишите процесс настройки автоматического анализатора кода для выявления уязвимостей.
- Вопрос:** Технологии решения задач планирования и контроля развития проекта .

Задание: Разработайте чек-лист для проведения ревьюирования программного модуля.

11. **Вопрос:** Метрики качества программного обеспечения (плотность дефектов, покрытие кода, сложность) .
Задание: Рассчитайте плотность дефектов для модуля, если в 5000 строк кода найдено 10 дефектов.
12. **Вопрос:** Стандартные метрики по прогнозированию затрат, сроков и качества .
Задание: Используя метод функциональных точек, оцените трудоемкость разработки модуля.
13. **Вопрос:** Понятие цикломатической сложности. Применение при ревьюировании.
Задание: Рассчитайте цикломатическую сложность для заданного графа программы и сделайте вывод о необходимости рефакторинга.
14. **Вопрос:** Измерение характеристик компонент программного продукта для соответствия заданным критериям .
Задание: Разработайте матрицу соответствия характеристик программного продукта требованиям технического задания.
15. **Вопрос:** Основные подходы к определению качества кода (Code Review, автоматический анализ).
Задание: Приведите примеры анти-паттернов программирования и укажите, как их выявить при ревью.
16. **Вопрос:** Методы исследования созданного программного кода с целью выявления ошибок и отклонения от алгоритма .
Задание: Опишите методику статического анализа кода для поиска неиспользуемых переменных.
17. **Вопрос:** Обратное проектирование. Анализ потоков данных и дизассемблирование .
Задание: Опишите процесс построения UML-диаграмм классов по готовому коду (обратное проектирование) .
18. **Вопрос:** Методы отладки программного обеспечения (пошаговое выполнение, контрольные точки).
Задание: Опишите порядок использования отладчика для поиска ошибки в заданном коде.
19. **Вопрос:** Инструментальные средства ревьюирования (IDE, статические анализаторы) .
Задание: Приведите примеры инструментов и опишите, какие типы ошибок они выявляют.
20. **Вопрос:** Типовые функциональные роли в коллективе разработчиков .
Задание: Распределите роли (разработчик, тестировщик, ревьюер) и опишите взаимодействие между ними в процессе ревью.
21. **Вопрос:** Этапы ревьюирования и процедура проведения инспекции программного кода .
Задание: Составьте регламент проведения формальной инспекции кода (вход, процесс, выход).
22. **Вопрос:** Методы и технологии тестирования и ревьюирования кода и проектной документации .
Задание: Определите различия между тестированием и ревьюированием кода, приведите примеры.
23. **Вопрос:** Принципы построения системы деятельности программного проекта .
Задание: Опишите, как ревьюирование интегрируется в процесс непрерывной интеграции (CI/CD).
24. **Вопрос:** Работа с проектной документацией, разработанной с использованием графических языков спецификаций .
Задание: Проанализируйте диаграмму вариантов использования на предмет полноты функциональных требований.
25. **Вопрос:** Управление рисками в процессе ревьюирования кода.
Задание: Перечислите основные риски при проведении ревьюирования и предложите меры по их минимизации.
26. **Вопрос:** Проведение сравнительного анализа программных продуктов и средств разработки .
Задание: Проведите сравнительный анализ сред разработки (Visual Studio, PyCharm,

- Eclipse) по критериям, определенным ТЗ.
27. **Вопрос:** Выявление наилучшего решения согласно критериям, определенным техническим заданием .
Задание: Выберите оптимальную технологию разработки (сравните 2-3 варианта) по заданным критериям.
28. **Вопрос:** Примеры сравнительного анализа программных продуктов .
Задание: Составьте отчет о сравнительном анализе фреймворков для web-разработки.
29. **Вопрос:** Обоснование выбора программного средства для ревьюирования.
Задание: Подберите инструмент для ревьюирования кода на конкретном языке программирования и обоснуйте выбор.
30. **Вопрос:** Итоговое обобщение: анализ и моделирование в контексте ревьюирования.
Задание: Выполните комплексное ревьюирование предоставленного фрагмента кода по чек-листу (стиль, логика, обработка ошибок).

5.2 Ключи к экзаменационным билетам

Билет 1

- **Теория:** Цели анализа — оценка эффективности, безопасности, сопровождаемости. Критерии: функциональность, производительность, стоимость владения, удобство интерфейса.
- **Практика (алгоритм):** Составить таблицу, где критериям присваиваются веса (например, цена 40%, удобство 30%, функциональность 30%), оценить продукты по 5-балльной шкале, умножить на веса и сложить. Выбрать наибольший результат.

Билет 2

- **Теория:** Ревьюирование — процесс проверки кода. Виды: неформальное (код-ревью коллегой), формальное (инспекция с совещанием), парное программирование.
- **Практика:** Ревьюер должен выполнять роль наставника. Запрещается критиковать личность, только код. Обязательно объяснять, *почему* нужно исправить ошибку.

Билет 3

- **Теория:** Методы: Scrum (спринты), Kanban (доски), XP (парное программирование). Git — распределенная система контроля версий.
- **Практика:** Создать репозиторий на GitHub (New Repository). Добавить README.md. Создать ветку feature/new-feature, сделать коммит, отправить в удаленный репозиторий (push), создать Pull Request.

Билет 4

- **Теория:** Планирование включает декомпозицию задач, оценку трудозатрат. Контроль — отслеживание отклонений.
- **Практика:** Составить простую диаграмму Ганта в Excel/MS Project. Дни 1-2: Анализ ТЗ; Дни 3-5: Написание кода; Дни 6-7: Ревью и исправление ошибок.

Билет 5

- **Теория:** Жизненный цикл: анализ требований -> проектирование -> разработка -> тестирование -> сопровождение. Структуры: функциональная, матричная, проектная.
- **Практика:** В Waterfall ревьюирование происходит после этапа кодирования перед тестированием. В Agile — постоянно в рамках каждого спринта (Definition of Done).

Билет 6

- **Теория:** Механизмы — отслеживание изменений, комментарии, проверка перед слиянием.
- **Практика:** Workflow Pull Request: 1) Создать ветку от main. 2) Внести изменения и закоммитить. 3) Отправить ветку в origin. 4) Открыть PR. 5) Назначить ревьюера. 6) Получить комментарии. 7) Исправить и запустить. 8) Смержить.

Билет 7

- **Теория:** Стандарты качества: ISO/IEC 25010. Чистый код: понятные имена переменных, короткие методы, отсутствие дублирования (DRY).
- **Практика:** Чек-лист "Чистый код": Приватные методы короткие; нет магических чисел; комментарии только там, где сложная логика; переменные названы осмысленно.

Билет 8

- **Теория:** Репозиторий — центральное хранилище кода. Структура повышает навигацию.
- **Практика:** Структура: /src (исходники), /docs (документация), /tests (тесты), /build (скомпилированные файлы, обычно в .gitignore), /configs.

Билет 9

- **Теория:** Методы: статический анализ (без запуска), линтеры, поиск уязвимостей.
- **Практика:** SonarQube: подключить проект, запустить анализ, посмотреть Quality Gate. ESLint: настроить конфиг .eslintrc, запустить команду `prx eslint` и исправить ошибки.

Билет 10

- **Теория:** Технологии — GitLab, GitHub, Jira (для задач), SonarQube.
- **Практика:** Чек-лист ревьюера: 1) Соответствие ТЗ. 2) Правильность именования. 3) Обработка ошибок (Try-Catch). 4) Логика. 5) Оценка производительности.

Билет 11

- **Теория:** Плотность дефектов = $(\text{Количество дефектов} / \text{Количество строк кода}) * 1000$.
- **Практика:** $10 \text{ дефектов} / 5000 \text{ строк} * 1000 = 2 \text{ дефекта на } 1000 \text{ строк (KLOC)}$. Код среднего качества.

Билет 12

- **Теория:** Функциональные точки (FP) оценивают сложность по внешним вводам/выводам, запросам, файлам, интерфейсам.
- **Практика (алгоритм):** 1) Посчитать количество элементов. 2) Присвоить веса (низкая/средняя/высокая сложность). 3) Перемножить и сложить. 4) Умножить на коэффициент производительности. 5) Умножить на часы на точку (обычно 10-15 часов).

Билет 13

- **Теория:** Цикломатическая сложность $(M) = E - N + 2P$ (где E - рёбра, N - вершины, P - компоненты связности). Показывает количество независимых путей.
- **Практика:** Если $M > 10$, метод слишком сложный, требует рефакторинга (разбить на несколько методов).

Билет 14

- **Теория:** Характеристики — функциональность, надежность, производительность, удобство.
- **Практика:** Построить матрицу трассировки (Traceability Matrix): Требование из ТЗ -> Раздел кода -> Тест-кейс. Проверить, что каждое требование покрыто кодом.

Билет 15

- **Теория:** Code Review (ручная проверка человеком), автоматический анализ (сканером).
- **Практика:** Анти-паттерны: "God Object" (объект, который делает всё), "Spaghetti Code" (запутанный код), "Copy-Paste" (дублирование). Ревьюер их выявляет вручную или с помощью SonarQube (Duplicate code detection).

Билет 16

- **Теория:** Статический анализ — исследование кода без его запуска.
- **Практика:** В IDE (Visual Studio/IntelliJ IDEA) найти серые переменные (неиспользуемые), удалить их, найти недостижимый код (после return).

Билет 17

- **Теория:** Обратное проектирование — восстановление модели по готовому коду. Дизассемблирование — перевод машинного кода в ассемблер.
- **Практика:** Использовать встроенные инструменты: Visual Studio -> Architecture -> Generate Code Map; или установить плагин PlantUML для генерации диаграмм классов.

Билет 18

- **Теория:** Отладка — поиск и устранение ошибок.
- **Практика:** Использовать точку останова (F9) на подозрительной строке, запустить (F5), пройти шаг за шагом (F10), посмотреть значения переменных в окне Locals, найти момент изменения.

Билет 19

- **Теория:** Инструментальные средства: IDE (Visual Studio, PyCharm), анализаторы (ReSharper, SonarQube, ESLint).
- **Практика:** ESLint выявляет неиспользуемые переменные, ошибки синтаксиса.

SonarQube выявляет уязвимости безопасности (SQL injections, XSS). IDE подсвечивает "Code Smells".

Билет 20

- **Теория:** Роли: Разработчик (пишет), Тестировщик (ищет баги), Ревьюер (проверяет качество, код-ревью).
- **Практика:** Разработчик отправляет PR. Ревьюер читает код, оставляет комментарии. Если замечания есть, разработчик исправляет, ревьюер ставит approve, мержит. Тестировщик потом проверяет результат в среде.

Билет 21

- **Теория:** Формальная инспекция (по Фагану): планирование, обзор, подготовка, совещание, доработка, контроль.
- **Практика:** Регламент: 1) Ревьюер получает код. 2) Студент сам находит 30% ошибок (анализ). 3) Совещание команды (30 мин). 4) Составление протокола. 5) Исправление (2 дня). 6) Повторная проверка.

Билет 22

- **Теория:** Тестирование проверяет *поведение* (работает или нет). Ревьюирование проверяет *качество кода* (как написано, соответствует ли стандартам).
- **Практика:** Пример: Кнопка открывает окно (Тест - Проверить, что окно открылось). Ревью - Проверить, не дублируется ли код открытия окон, не нарушает ли он принципы SOLID, нет ли уязвимостей.

Билет 23

- **Теория:** CI/CD — непрерывная интеграция и доставка.
- **Практика:** В GitLab CI настроить пайплайн: 1) Линтер (ESLint). 2) Юнит-тесты (Jest). 3) Сборка. 4) Автодеплой. Если ревью не пройдено, происходит блокировка (merge нельзя).

Билет 24

- **Теория:** Проектная документация с графическими языками (UML).
- **Практика:** Ревьюируем диаграмму Use Case: Все ли актеры указаны? Каждый ли сценарий имеет альтернативный поток? Не пропущены ли системные требования?

Билет 25

- **Теория:** Риски при ревью: субъективность, длительное время проверки, конфликты в команде.
- **Практика:** Минимизация: Использовать чек-листы, ограничить размер PR (не более 400 строк), обучать разработчиков и ревьюеров, использовать автоматические линтеры для рутины, оставлять ревьюеру только логику.

Билет 26

- **Теория:** Сравнительный анализ проводится по заданным критериям из ТЗ.
- **Практика:** Критерии: скорость разработки, стоимость лицензии, масштабируемость, наличие плагинов. Взвешенная оценка: Excel таблица, где каждой среде ставятся баллы. Побеждает та, что набрала больше всего баллов.

Билет 27

- **Теория:** Выявление наилучшего решения согласно критериям.
- **Практика:** Технология: React vs Vue. По критерию "Простота изучения" побеждает Vue. По критерию "Востребованность на рынке" — React. Выбор зависит от приоритета заказчика.

Билет 28

- **Теория:** Сравнение программных продуктов.
- **Практика:** Сравнение фреймворков Django vs Laravel: Язык (Python/PHP), безопасность (у Django выше из коробки), производительность (Laravel быстрее для малых проектов).

Билет 29

- **Теория:** Обоснование выбора ПО для ревью.
- **Практика:** Для языка Java выбираем SonarQube (интеграция с Maven, находит code smells). Для JavaScript выбираем ESLint (быстрый, кастомизируемый). Обоснование: инструмент должен поддерживать синтаксис конкретного языка и выявлять

конкретные виды ошибок (логические, безопасность, стиль).

Билет 30

- **Теория:** Комплексное ревьюирование.
- **Практика (итоговый алгоритм):** 1) Прочитать требования. 2) Проверить код по чек-листу. 3) Оценить по метрикам (сложность, дубли). 4) Запустить линтер. 5) Выставить вердикт (approved, needs work) и подробный отзыв с рекомендациями.