# Перечень заданий к дифференцированному зачету по учебной практике УП.01 по ПМ.03 Ревьюирование программных продуктов (4 курс, 7 семестр, 2026-2027 уч. г.)

**Метод и форма контроля:** Практические работа в формате демонстрационного экзамена

**Вид контроля:** Практическая работа

## Перечень заданий:

## Задание 1 (Время выполнения: 60 минут)

### Оборудование

1. Персональный компьютер (208-II Лаборатория информационных ресурсов)

### Программное обеспечение

1. Microsoft Windows 10 (208-II Лаборатория информационных ресурсов)
2. Microsoft Visual Studio (208-II Лаборатория информационных ресурсов)

### Исходные файлы

- Microsoft Word_1715402403_1725791990.docx

### Проверяемые ПК

- ПК3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией

### Содержание задания:

1. Изучите техническую документацию проекта, в котором вы будете проводить ревью кода. Обратите внимание на следующие аспекты:

2. Ознакомьтесь с фрагментом кода для проведения ревью.

3. Проанализируйте код на предмет соответствия требованиям, изложенным в технической документации.

4. Подготовьте отчет о ревью кода, в котором отразите следующие пункты: - Список обнаруженных несоответствий между кодом и технической документацией; - Предложения по исправлению этих несоответствий (если применимо); - Комментарии и замечания по улучшению качества кода с точки зрения читаемости, производительности и расширяемости.

5. Представьте отчет о ревью кода.

### Критерии оценивания

| Критерий | Баллы | Включенность | Итого |
|---|---|---|---|
| | | | |

| | | | |
|---|---|---|---|
| Студент тщательно изучил техническую документацию и продемонстрировал глубокое понимание требований к функциональности, архитектуре, используемым технологиям, правилам безопасности и соглашениям по стилю кодирования. | 1 | 1 | 1 |
| В отчете о ревью кода выявлены все несоответствия между кодом и технической документацией, включая нарушения требований, архитектурных принципов, правил безопасности и соглашений по стилю кодирования. | 1 | 1 | 1 |
| Предложены обоснованные и эффективные способы исправления обнаруженных несоответствий. | 1 | 1 | 1 |
| Даны конструктивные комментарии и рекомендации по улучшению качества кода с точки зрения читаемости, производительности и расширяемости. | 1 | 1 | 1 |
| Студент четко и аргументированно представил отчет о ревью, ссылаясь на техническую документацию. | 1 | 1 | 1 |
| **Итого баллов** | | | **5** |

## Задание 2 (Время выполнения: 60 минут)

### Оборудование

1. Персональный компьютер (208-II Лаборатория информационных ресурсов)

### Программное обеспечение

1. Microsoft Windows 10 (208-II Лаборатория информационных ресурсов)
2. Microsoft Office Professional Plus 2019 (208-II Лаборатория информационных ресурсов)

### Проверяемые ПК

- ПК3.2 Выполнять процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям
- ПК3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием

### Содержание задания:

Необходимо провести сравнительный анализ трех популярных фреймворков для веб-разработки с целью выявления наилучшего решения для создания веб-приложения с определенными требованиями.

Требования к веб-приложению:

- Веб-приложение должно иметь современный и отзывчивый пользовательский интерфейс.
- Приложение должно обеспечивать безопасную авторизацию и аутентификацию пользователей.
- Должна быть возможность работы с базами данных для хранения и извлечения информации.
- Приложение должно хорошо масштабироваться и поддерживать большое количество одновременных пользователей.
- Необходимо наличие активного сообщества разработчиков и хорошей документации.

Фреймворки для сравнения: 1. React.js (JavaScript) 2. Angular (TypeScript) 3. Django (Python)

Задание:
1. Изучите документацию и ознакомьтесь с особенностями каждого фреймворка.
2. Проведите сравнительный анализ фреймворков по следующим критериям: - Производительность - Масштабируемость - Безопасность - Простота изучения и использования - Поддержка и документация - Сообщество разработчиков - Другие важные критерии (если применимо)
3. Для каждого критерия оцените и сравните фреймворки, обоснуйте свои оценки.
4. На основе проведенного анализа сделайте вывод о том, какой фреймворк лучше всего подходит для разработки веб-приложения с указанными требованиями. Обоснуйте свой выбор.
5. Подготовьте отчет, в котором изложите результаты сравнительного анализа и свои рекомендации.

**Критерии оценивания**

| **Критерий** | **Баллы** | **Включенность** | **Итого** |
|---|---|---|---|
| Проведен сравнительный анализ фреймворков по предложенным критериям. | 1 | 1 | 1 |
| Проведен сравнительный анализ фреймворков по иным критериям. | 1 | 1 | 1 |
| Каждый критерий оценен. Студент сравнил фреймворки между собой. Обоснованы оценки. | 2 | 1 | 2 |
| Сделан вывод о том, какой фреймворк лучше всего подходит для разработки веб-приложения с указанными требованиями. Выбор обоснован. | 1 | 1 | 1 |
| Сделан вывод о проделанной работе, в котором изложен результат сравнительного анализа и рекомендации. | 1 | 1 | 1 |
| **Итого баллов** | | | **6** |

## Задание 3 (Время выполнения: 60 минут)

**Оборудование**

1. Персональный компьютер (208-II Лаборатория информационных ресурсов)

**Программное обеспечение**

1. Microsoft Windows 10 (208-II Лаборатория информационных ресурсов)
2. Visual Studio Code (208-II Лаборатория информационных ресурсов)

**Исходные файлы**

- Microsoft Word (2)_1715995850_1725792327.docx

**Проверяемые ПК**

- ПК3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма

**Содержание задания:**

Необходимо провести исследование существующего программного кода (пример кода находится в исходных файлах) с использованием специализированных инструментов статического анализа кода. Цель задания - выявить потенциальные ошибки, уязвимости и отклонения от заданного алгоритма или стандартов кодирования.

Задача:
1. Изучите исходный код программы.

2. Изучите предоставленную документацию к программе.

3. Выберите подходящие инструменты статического анализа кода для выбранного языка программирования. Рекомендуется использовать не менее двух инструментов для более полного анализа. Примеры инструментов: - Для Python: PyLint, Pyright, Bandit, MyPy - Для Java: SonarQube, SpotBugs, PMD, CheckStyle

4. Настройте выбранные инструменты статического анализа кода в соответствии с требованиями и стандартами кодирования (если применимо).

5. Проведите анализ исходного кода программы с помощью выбранных инструментов.

6. Изучите результаты анализа, выявленные ошибки, предупреждения и потенциальные проблемы в коде.

7. Классифицируйте обнаруженные проблемы по категориям (ошибки, нарушения стандартов кодирования, потенциальные уязвимости и т.д.).

8. Для каждой обнаруженной проблемы оцените ее серьезность и предложите способы исправления или улучшения кода.

9. Подготовьте отчет, в котором изложите результаты анализа, классификацию проблем и рекомендации по улучшению кода.

**Критерии оценивания**

| **Критерий** | **Баллы** | **Включенность** | **Итого** |
|---|---|---|---|
| Выбран подходящий инструмент статического анализа кода. Верно настроены. | 2 | 1 | 2 |
| Проведен анализ исходного кода программы с помощью выбранных инструментов. | 1 | 1 | 1 |
| Изучены результаты анализа, выявленные ошибки, предупреждения и потенциальные проблемы в коде. | 1 | 1 | 1 |
| Классифицированны обнаруженные проблемы по категориям. | 1 | 1 | 1 |
| Каждая проблема оценена. Предложены способы по ее устранению. | 1 | 1 | 1 |
| Подготовлен отчет о результатах работы с рекомендациями по улучшению кода. | 1 | 1 | 1 |
| **Итого баллов** | | | **7** |

## Перевод баллов в оценку

| Критерий | Нижняя граница, в % | Баллы |
|---|---|---|
| Оценка «3» | 50 | 9 |
| Оценка «4» | 80 | 14.4 |
| Оценка «5» | 100 | 18 |