

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Баламирзоев Назим Лиманович  
Должность: Ректор  
Дата подписания: 04.06.2022 18:04  
Уникальный программный ключ:  
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926

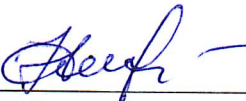
Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО «Дагестанский государственный технический университет»

## ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### ПМ. 02 Осуществление интеграции программных модулей

#### по ПП. 02.01 ПРОИЗВОДСТВЕННАЯ ПРАКТИКА (по профилю специальности)

специальность	09.02.07 Информационные системы и программирование
квалификация	администратор баз данных
уровень образования	СПО на базе основного общего образования

Разработчик  Мусаева У.А., к.т.н., доцент  
подпись

Фонд оценочных средств обсужден на заседании кафедры УиИвТСиВТ

« 1 » 11 2022 г., протокол № 3

Зав. кафедрой УиИвТСиВТ  Мусаева У.А., к.т.н., доцент  
подпись

г. Махачкала - 2022

## СОДЕРЖАНИЕ

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ .....	3
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ .....	3
3. ОЦЕНКА ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ .....	7
3.1. Контроль и оценка освоения производственной практики по темам .....	7
3.2. Перечень вопросов для текущего контроля знаний .....	8
4. ПЕРЕЧЕНЬ ЗАДАНИЙ ДЛЯ ОЦЕНКИ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ ...	9
5. КРИТЕРИИ ОЦЕНКИ .....	25

## 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств (ФОС) является неотъемлемой частью рабочей программы ПП 02.01 Производственная практика (по профилю специальности) и предназначен для контроля и оценки образовательных достижений, обучающихся (в т.ч. по самостоятельной работе студентов, далее – СРС), освоивших программу данного модуля.

Целью фонда оценочных средств является установление соответствия уровня подготовки обучающихся требованиям ФГОС СПО по специальности 09.02.07 Информационные системы и программирование.

Рабочей программой ПП 02.01 Производственная практика (по профилю специальности) предусмотрено формирование профессиональных компетенций:

- 1) ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент;
- 2) ПК 2.2 Выполнять интеграцию модулей в программное обеспечение;
- 3) ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств;
- 4) ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения;
- 5) ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

Формой аттестации по практике ПП 02.01 Производственная практика (по профилю специальности) является зачет с оценкой.

## 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по производственной практике осуществляется комплексная проверка следующих умений, знаний и практического опыта, а также динамика формирования профессиональных компетенций.

Таблица 1

Результаты обучения: знания, умения, практический опыт	Формируемые виды деятельности/ компетенции
<b>Знать:</b> 31 - модели процесса разработки программного обеспечения; 32 - основные принципы процесса разработки программного обеспечения; 33 - основные подходы к интегрированию программных модулей; 34 - виды и варианты интеграционных решений; 35 - современные технологии и инструменты интеграции; 36 - основные протоколы доступа к данным; 37 - методы и способы идентификации сбоев и ошибок при интеграции приложений; 38 - методы отладочных классов; 39 - стандарты качества программной документации; 310 - основы организации инспектирования и верификации;	Осуществление интеграции программных модулей/ ПК 2.1

Результаты обучения: знания, умения, практический опыт	Формируемые виды деятельности/ компетенции
311 - встроенные и основные специализированные инструменты анализа качества программных продуктов; 312 - графические средства проектирования архитектуры программных продуктов; 313 - методы организации работы в команде разработчиков.	
<b>Уметь:</b>	
У1 - анализировать проектную и техническую документацию; У2 - использовать специализированные графические средства построения и анализа архитектуры программных продуктов; У3 - организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов; У4 - определять источники и приемники данных; У5 - проводить сравнительный анализ; У6 - выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace); У7 - оценивать размер минимального набора тестов; У8 - разрабатывать тестовые пакеты и тестовые сценарии; У9 - выявлять ошибки в системных компонентах на основе спецификаций.	
<b>Иметь практический опыт в:</b>	
П1 - разрабатывать и оформлять требования к программным модулям по предложенной документации; П2 - разрабатывать тестовые наборы (пакеты) для программного модуля; П3 - разрабатывать тестовые сценарии программного средства; П4 - инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.	
<b>Знать:</b>	Осуществление интеграции программных модулей /ПК 2.2
31 - модели процесса разработки программного обеспечения; 32 - основные принципы процесса разработки программного обеспечения; 33 - основные подходы к интегрированию программных модулей; 34 - основы верификации программного обеспечения; 35 - современные технологии и инструменты интеграции; 36 - основные протоколы доступа к данным; 37 - методы и способы идентификации сбоев и ошибок при интеграции приложений; 38 - основные методы и виды тестирования программных продуктов; 39 - стандарты качества программной документации; 310 - основы организации инспектирования и верификации; 311 - приемы работы с инструментальными средствами	

Результаты обучения: знания, умения, практический опыт	Формируемые виды деятельности/ компетенции
<p>тестирования и отладки; 312 - методы организации работы в команде разработчиков.</p>	
<p><b>Уметь:</b></p>	
<p>У1 - использовать выбранную систему контроля версий; У2 - использовать методы для получения кода с заданной функциональностью и степенью качества; У3 - организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов; У4 - использовать различные транспортные протоколы и стандарты форматирования сообщений; У5 - выполнять тестирование интеграции; У6 - организовывать постобработку данных; У7 - создавать классы-исключения на основе базовых классов; У8 - выполнять ручное и автоматизированное тестирование программного модуля; У9 - выявлять ошибки в системных компонентах на основе спецификаций; У10 - использовать приемы работы в системах контроля версий.</p>	
<p><b>Иметь практический опыт в:</b></p>	
<p>П1 - интегрировать модули в программное обеспечение; П2 - отлаживать программные модули; П3 - инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>	<p>Осуществление интеграции программных модулей/ ПК 2.3</p>
<p><b>Знать:</b></p>	
<p>31 - модели процесса разработки программного обеспечения; 32 - основные принципы процесса разработки программного обеспечения; 33 - основные подходы к интегрированию программных модулей; 34 - основы верификации и аттестации программного обеспечения; 35 - методы и способы идентификации сбоев и ошибок при интеграции приложений; 36 - основные методы отладки; 37 - методы и схемы обработки исключительных ситуаций; 38 - приемы работы с инструментальными средствами тестирования и отладки; 39 - основы организации инспектирования и верификации; 310 - встроенные и основные специализированные инструменты анализа качества программных продуктов; 311 - методы организации работы в команде разработчиков.</p>	
<p><b>Уметь:</b></p>	

Результаты обучения: знания, умения, практический опыт	Формируемые виды деятельности/ компетенции
<p>У1 - использовать выбранную систему контроля версий;  У2 - использовать методы для получения кода с заданной функциональностью и степенью качества;  У3 - анализировать проектную и техническую документацию;  У4 - использовать инструментальные средства отладки программных продуктов;  У5 - определять источники и приемники данных;  У6 - выполнять тестирование интеграции;  У7 - организовывать постобработку данных;  У8 - использовать приемы работы в системах контроля версии;  У9 - выполнять отладку, используя методы и инструменты условной компиляции;  У10 - выявлять ошибки в системных компонентах на основе спецификаций.</p>	
<p><b>Иметь практический опыт в:</b></p>	
<p>П1 - отлаживать программные модули;  П2 - инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>	
<p><b>Знать:</b></p>	<p>Осуществление интеграции программных модулей/ПК 2.4</p>
<p>З1 - модели процесса разработки программного обеспечения;  З2 - основные принципы процесса разработки программного обеспечения;  З3 - основные подходы к интегрированию программных модулей;  З4 - основы верификации и аттестации программного обеспечения;  З5 - методы и способы идентификации сбоев и ошибок при интеграции приложений;  З6 - методы и схемы обработки исключительных ситуаций;  З7 - основные методы и виды тестирования программных продуктов;  З8 - приемы работы с инструментальными средствами тестирования и отладки;  З9 - стандарты качества программной документации;  З10 - основы организации инспектирования и верификации;  З11 - встроенные и основные специализированные инструменты анализа качества программных продуктов;  З12 - методы организации работы в команде разработчиков.</p>	
<p><b>Уметь:</b></p>	

Результаты обучения: знания, умения, практический опыт	Формируемые виды деятельности/ компетенции
<p>У1 - использовать выбранную систему контроля версий;  У2 - анализировать проектную и техническую документацию;  У3 - выполнять тестирование интеграции;  У4 - организовывать постобработку данных;  У5 - оценивать размер минимального набора тестов;  У6 - разрабатывать тестовые пакеты и тестовые сценарии;  У7 - выполнять ручное и автоматизированное тестирование программного модуля;  У8 - выявлять ошибки в системных компонентах на основе спецификаций.</p>	
<p><b>Иметь практический опыт в:</b></p>	
<p>П1 - разрабатывать тестовые наборы (пакеты) для программного модуля;  П2 - разрабатывать тестовые сценарии программного средства;  П3 - инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>	
<p><b>Знать:</b></p>	<p>Осуществление интеграции программных модулей/ПК 2.5</p>
<p>31 - модели процесса разработки программного обеспечения;  32 - основные принципы процесса разработки программного обеспечения;  33 - основные подходы к интегрированию программных модулей;  34 - основы верификации и аттестации программного обеспечения;  35 - стандарты качества программной документации;  36 - основы организации инспектирования и верификации;  37 - встроенные и основные специализированные инструменты анализа качества программных продуктов;  38 - методы организации работы в команде разработчиков.</p>	
<p><b>Уметь:</b></p>	
<p>У1 - использовать выбранную систему контроля версий;  У2 - использовать методы для получения кода с заданной функциональностью и степенью качества;  У3 - анализировать проектную и техническую документацию;  У4 - организовывать постобработку данных;  У5 - приемы работы в системах контроля версий;  У6 - выявлять ошибки в системных компонентах на основе спецификаций.</p>	
<p><b>Иметь практический опыт в:</b></p>	
<p>П1- инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>	

### 3. ОЦЕНКА ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

#### 3.1. Контроль и оценка освоения производственной практики по темам

Предметом оценки служат знания, умения и практический опыт, предусмотренные ФГОС СПО, направленные на формирование профессиональных компетенций.

Таблица 2

Элемент производственной практики	Формы и методы контроля			
	Текущий контроль		Промежуточная аттестация	
	Форма контроля	Проверяемые компетенции/знания/умения/ практический опыт	Форма контроля	Проверяемые компетенции/знания/умения/ практический опыт
Тема 1. Технический анализ.	Практическая работа	ПК 2.1, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, 312, 313, У1, У2, У3, У4, У5, У6, У7, У8, У9, П1, П2, П3, П4	Зачетная работа	ПК 2.1, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, 312, 313, У1, У2, У3, У4, У5, У6, У7, У8, У9, П1, П2, П3, П4
Тема 2. Проектирование.	Практическая работа	ПК 2.2, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, 312, У1, У2, У3, У4, У5, У6, У7, У8, У9, У10, П1, П2, П3		ПК 2.2, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, 312, У1, У2, У3, У4, У5, У6, У7, У8, У9, У10, П1, П2, П3
Тема 3. Программная реализация проекта	Практическая работа	ПК 2.2, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, 312, У1, У2, У3, У4, У5, У6, У7, У8, У9, У10, П1, П2, П3		ПК 2.2, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, 312, У1, У2, У3, У4, У5, У6, У7, У8, У9, У10, П1, П2, П3
Тема 4. Тестирование.	Практическая работа	ПК 2.3, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, У1, У2, У3, У4, У5, У6, У7, У8, У9, У10, П1, П2; ПК 2.4, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, 312, У1, У2, У3, У4, У5, У6, У7, У8, П1, П2, П3		ПК 2.3, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, У1, У2, У3, У4, У5, У6, У7, У8, У9, У10, П1, П2; ПК 2.4, 31, 32, 33, 34, 35, 36, 37, 38, 39, 310, 311, 312, У1, У2, У3, У4, У5, У6, У7, У8, П1, П2, П3
Тема 5. Документирование.	Практическая работа	ПК 2.5, 31, 32, 33, 34, 35, 36, 37, 38, У1, У2, У3, У4, У5, У6, П1		ПК 2.5, 31, 32, 33, 34, 35, 36, 37, 38, У1, У2, У3, У4, У5, У6, П1



### 3.2. Перечень вопросов для текущего контроля знаний

1. Основные понятия и определения.
2. Типы ПО.
3. Жизненный цикл и этапы разработки ПО.
4. Модели ЖЦ ПО.
5. Качество ПО.
6. Введение в анализ требований.
7. Описание требований заказчика.
8. Шаги разработки пользовательского интерфейса.
9. Методологии и инструментальные средства для требований заказчика.
10. Детальные требования и их типы.
11. Методы проектирования ПО.
12. Основы проектирования программных систем.
13. Модульное проектирование.
14. Структурное проектирование.
15. Основы ООП программных систем.
16. Архитектура ПО.
17. Диаграмма UML.
18. История развития технологии программирования и построения архитектуры ПО.
19. Оценка стоимости и причин ошибок в ПО.
20. Основные понятия отладки и тестирования ПО.
21. Тестирование интеграции ПО.
22. Тестированию по методу «белого ящика».
23. Методика тестирования программных систем.
24. Функциональное тестирование ПО.
25. Особенности процесса разработки ПО.
26. Управление приоритетами проектов.
27. Планирование проекта.
28. Меры, метрики и модели.
29. Управление персоналом проекта.
30. Реализация проекта.
31. Управление рисками проекта.
32. Документация, создаваемая и используемая в процессе разработки ПО.
33. Стандартизация программной документации.
34. Государственные стандарты РФ (ГОСТЫ Р).
35. Понятие ПО. Состав уровней ПО.
36. Понятия задачи и приложения.
37. Свойства сложных программ.
38. Процесс создания программ.
39. Свойства ПО.
40. Группы и направления разработки ПО.
41. Понятие ЯП. Классификация ЯП.
42. Краткая характеристика популярных ЯП: C++, Java, JavaScript, Python, 1С.
43. Классы инструментальных средств разработки ПО.
44. Инструментальная среда программирования.
45. Рабочее место компьютерной технологии.
46. Инструментальная система технологии программирования.
47. Интерфейс API.
48. Схема обращения с API.
49. Понятие системы программирования.

50. Компоненты системы программирования.
51. Функции системы программирования.
52. Виды инструментального программного обеспечения
53. Популярные системы программирования.
54. Установка PyCharm в Windows.
55. Настройка начальных параметров интерактивной среды PyCharm.
56. Создание проекта. Структура проекта.
57. Запись кода в PyCharm.
58. Загрузка кода в PyCharm.
59. Отладка в PyCharm.
60. Тестирование в PyCharm.
61. Редактирование существующего проекта в PyCharm.
62. Поиск и навигация в PyCharm.
63. Управление версиями в PyCharm.
64. Плагины и внешние инструменты в PyCharm.
65. Возможности профессиональной версии PyCharm.
66. Концепция системы «1С:Предприятие».
67. Встроенный язык платформы «1С:Предприятие».
68. Конфигурируемость.
69. Конфигурация и прикладное решение.
70. Режимы работы системы.
71. Интерфейс конфигуратора.
72. Дерево объектов конфигурации.
73. Объекты конфигурации.
74. Имя и синоним объекта конфигурации.
75. Окно редактирования объекта конфигурации и палитра свойств объекта конфигурации.
76. Типовые прикладные решения 1С.
77. Основная конфигурация и конфигурация базы данных.
78. Синтакс-помощник, контекстная подсказка и отладчик.
79. Понятие решения. Множество решений, оптимальное решение.
80. Показатель эффективности решения.
81. Математические модели, принципы их построения, виды моделей.
82. Задачи: классификация, методы решения, граничные условия.
83. Общий вид и основная задача линейного программирования.
84. Общий вид задач нелинейного программирования.
85. Постановка задачи динамического программирования.
86. Экономические задачи, решаемые с помощью методов динамического программирования.
87. Системы массового обслуживания: понятия, примеры, модели.
88. Основные понятия теории игр.

#### **4. ПЕРЕЧЕНЬ ЗАДАНИЙ ДЛЯ ОЦЕНКИ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ**

##### ***Формируемая компетенция ПК 2.1***

##### **Перечень заданий закрытого типа**

Задание № 1. Что такое жизненный цикл программного обеспечения?

- а) период технической поддержки, на протяжении которого разработчики сначала дорабатывают систему, а затем следят, чтобы все найденные уязвимости были исправлены;
- б) это непрерывный процесс, начинающийся с момента принятия решения и заканчивающийся в момент полного изъятия ее из эксплуатации;
- в) период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Задание №2. Какая модель является предметом формализации?

- а) структурно-функциональная;
- б) физическая;
- в) математическая;
- г) имитационная.

Задание №3. Определите, что такое математическая модель объекта?

- а) совокупность записанных на языке математики формул, отражающих те или иные свойства объекта-оригинала или его поведение;
- б) совокупность данных, содержащих информацию о количественных характеристиках объекта и его поведении в виде таблицы;
- в) созданная из какого-либо материала модель, точно отражающая внешние признаки объекта-оригинала.

Задание № 4. Выберите правильную формулировку понятия программного проекта:

- а) это точное описание порядка действий, которые должен выполнить исполнитель для решения задачи за конечное время;
- б) комплекс взаимосвязанных работ, выполняемых командой проекта с целью получения уникального программного продукта или услуги в течение заданного периода при установленном бюджете и потребляемых в ходе реализации проекта ресурсах в условиях повышенного риска, требующих специфического управления;
- в) это комплекс взаимосвязанных программ, предназначенный для поставки, передачи, продажи пользователю;
- г) это последовательность инструкций, предназначенная для исполнения вычислительной машиной.

Задание №5. Какой стандарт распространяется на системы для автоматизации различных видов деятельности (управление, проектирование, исследование и т. п.), включая их сочетания, и устанавливает состав, содержание, правила оформления документа “Техническое задание на создание (развитие или модернизацию) системы”?

- а) ГОСТ 34.602—89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;
- б) ГОСТ Р 51904-2002 — Программное обеспечение встроенных систем. Общие требования к разработке и документированию;
- в) ISO 12207 - Информационные технологии. Процессы жизненного цикла программного обеспечения;
- г) ISO 6592 – Обработка информации. Руководство по разработке документации для вычислительных систем.

Задание №6. Как называется проверка правильности программного продукта, т.е. подтверждение того, что он действительно удовлетворяет потребности и ожидания пользователей, заказчиков и других заинтересованных сторон?

- а) верификация;
- б) валидация;
- в) спецификация;
- г) компиляция.

Задание № 7. Сопоставьте схемы и характеристики моделей жизненного цикла ПО.

1.		а) Модель создания определенной версии продукта, уточнения требований проекта, определение его качества и планирования следующих работ;
2.		б) Модель перехода с одного этапа на следующий только после полного завершения работ на текущем этапе;
3.		в) Модель выполнения работ поэтапно с промежуточным контролем;

Задание № 8. Сопоставьте названия и краткое содержание работ этапов разработки ПО.

1. Анализ.	а) на этом этапе руководитель и команда проекта определяют масштаб и цели проекта;
2. Планирование.	б) на этом этапе команда проводит тщательный анализ проекта и требований;
3. Проектирование.	в) на этом этапе команда создает реальный продукт: код, дизайн и другие компоненты, необходимые для создания программного обеспечения;
4. Разработка.	г) на этом этапе команда создает чертежи программного обеспечения, включая макет и дизайн продукта, а также функциональность каждого компонента.

Задание №9. Установите правильную последовательность этапов создания документации на проект ПО:

- а) сбор и документирование архитектурной информации в виде группы представлений и специального блока с общей информацией для всех представлений;

- б) определение потребностей заинтересованных сторон;
- в) подготовка архитектурной документации в вид, пригодный для той или иной заинтересованной стороны;
- г) проверка того, что созданная документация удовлетворяет требованиям заинтересованных сторон.

Задание №10. Установите правильную последовательность этапов процесса разработки прототипа.

- а) доработка прототипа с учетом полученных замечаний и согласование с участниками проекта.
- б) проверка прототипа участниками проекта, выявление необходимых изменений и дополнений;
- в) разработка первого варианта прототипа;
- г) определение начальных и требований к продукту проекта;

#### **Перечень заданий открытого типа**

Задание №1. Какому типу тестирования принадлежат такие методы формирования тестовых наборов, как эквивалентное разбиение, анализ граничных значений, анализ причинно-следственных связей и предположение об ошибке?

Задание №2. Какой командой можно подключить к проекту программы на языке C# пространство имен, в котором находятся отладочные классы Debug и Trace?

Задание №3. В какой из двух конфигураций проекта на языке C# игнорируются методы отладочного класса Debug с атрибутом условной компиляции Conditional с константой DEBUG?

Задание №4. Определите оптимальную точку для решения задачи целочисленного программирования, представленной в виде следующей математической модели.

$$F = 5x_1 + 4x_2 + 4 \rightarrow \max,$$

$$\begin{cases} 2x_1 + 2x_2 \leq 5, \\ x_1 \geq 0, x_2 \geq 0, \\ x_1, x_2 - \text{целые числа,} \end{cases}$$

Задание №5. Что является областью допустимых решений задач целочисленного линейного программирования от двух переменных при решении геометрическим методом?

Задание №6. Какое тестирование называют тестированием по «маршрутам», так как в этом случае тестовые наборы формируют путем анализа маршрутов (последовательностей операторов программы, которые выполняются при конкретном варианте исходных данных), предусмотренных алгоритмом?

Задание № 7. Какой критерий формирования тестовых наборов подразумевает такой подбор тестов, чтобы каждый оператор программы выполнялся, по крайней мере, один раз?

Задание № 8. Как называется частичная, возможная или предварительная версия предлагаемого продукта проекта?

Задание № 9. Дополните предложение: «Главной задачей интеграционного тестирования программной системы является проверка разных модулей системы при их .... объединении».

Задание №10. Дополните предложение: «Объектно-ориентированный подход предполагает, что при разработке программы должны быть определены ...., используемых в программе, и построены их описания, затем созданы экземпляры необходимых объектов и определено взаимодействие между ними».

### **Формируемая компетенция ПК 2.2**

#### **Перечень заданий закрытого типа**

Задание №1. Как называется организационная структура программной системы, включающая модули, их внешние характеристики, а также отношения между модулями?

- а) архитектура;
- б) схема;
- в) иерархия;
- г) пакет.

Задание №2. Как называется согласно критерию универсальности ПО возможность использования ПО в различных областях применения?

- а) мобильность;
- б) модифицируемость;
- в) гибкость;
- г) совместимость.

Задание №3. Какой класс в Python является базовым для всех программных исключений, возбуждаемых вычислительными операциями?

- а) BaseException;
- б) ArithmeticError;
- в) EnvironmentError;
- г) LookupError.

Задание №4. Какой параметр интеграции основывается на том, что разработчики разных систем изначально приняли разные решения, предположения и допущения, которые не стыкуются между собой?

- а) технологическая разница;
- б) несовместимость лицензий;
- в) несовместимость стандартов;
- г) концептуальная разница.

Задание №5. К какому уровню относятся протоколы UDP (протокол пользовательских дейтаграмм) и TCP (протокол управления передачей) управления доставкой пакетов данных между устройствами в сети Интернет?

- а) транспортному;
- б) физическому;
- в) каналному;
- г) сетевому.

Задание №6. Выберите правильный вариант разбиения длинной строки на несколько строк согласно стандарту кодирования, PEP8 на Python:

а) long_string = "Очень длинная строка, которую "+ "надо разбить на несколько строк."
б) long_string = "Очень длинная строка, которую "/ "надо разбить на несколько строк."
в) long_string = "Очень длинная строка, которую \" "надо разбить на несколько строк."
г) long_string = "Очень длинная строка, которую "* "надо разбить на несколько строк."

Задание № 7. Сопоставьте характеристики и названия групп операций интеграции данных.

1. Трансформация данных	а) комбинация различных представлений одной и той же сущности реального мира в единое представление;
2. Разрешение сущностей	б) преобразование из исходной схемы (схемы коллекции - источника данных) в целевую (единую интегрированную схему);
3. Слияние сущностей	в) выделение и связывание информации об одной и той же сущности реального мира из разных коллекций данных.

Задание №8. Сопоставьте операторы обработки исключительных ситуаций в программном коде и названия языков программирования, к которым они относятся.

1. try: //операторы except: //операторы	а) C#;
2. try { //операторы } catch { //операторы }	б) 1C;
3. Попытка //операторы попытки Исключение //операторы исключения. КонецПопытки	в) Python.

Задание № 9. Установите правильную последовательность основных шагов процедуры регрессионного тестирования ПО.

- а) Шаг 3. Установите критерии входа и точку входа (первой команды программы);
- б) Шаг 1. Распознайте изменения исходного программного кода;
- в) Шаг 4. Выберите точку выхода (последней команды программы);
- г) Шаг 2. Установите приоритет этих изменений и требований к программному продукту;
- д) Шаг 5. Составьте план тестов.

Задание №10. Установите правильную последовательность выполнения блоков оператора исключения:

- а) если во время выполнения блока try в каком-то месте возникло исключение, оставшаяся часть блока try пропускается;
- б) если при выполнении блока try не возникло никаких исключений, блок except пропускается, и выполняется дальнейший код;
- в) если исключение, которое возникло, не указано в блоке except, выполнение программы прерывается и выдается ошибка;
- г) если в блоке except указано исключение, которое возникло, выполняется код в блоке except;
- д) выполняются выражения, которые записаны в блоке try;

### Перечень заданий открытого типа

Задание №1. Какие стандарты форматирования сообщений определяют, как данные структурируются и кодируются для передачи в сети Интернет?

Задание №2. Как называется способ обработки исключений, при котором осуществляется переход управления в специальный, заранее подготовленный, блок кода?

Задание № 3. Как называется метод верификации программного обеспечения, при котором программа представляется в виде формальной модели, которая затем проверяется на соответствие заданным требованиям и спецификациям?

Задание № 4. Какая качественная характеристика ПО определяется частотой отказов, вызванных наличием ошибок в программном обеспечении?

Задание № 5. Сколько групп базовых показателей используется для оценки качества программного средства?

Задание №6. В какой строке предложенного фрагмента программного кода на C# возникнет исключение типа FormatException?

```
try
{
    Console.WriteLine("Введите число: ");
    string str = Console.ReadLine();
    int number = Convert.ToInt32(str);
    number*=10;
}
catch (FormatException)
{
    Console.WriteLine("Ошибка! Вы ввели не число");
}
Console.WriteLine();
Console.WriteLine("Результат вычислений: "+ number);
Console.ReadKey(true);
```

Задание №7. Как называется описание способов взаимодействия одной компьютерной программы с другими?



Задание №8. Какую инструкцию должен использовать разработчик для создания исключения в программе на C#?

Задание № 9. Запишите название определения: «Для обработки таких ситуаций, как деление на ноль, невозможность считать данные из файла (устройства), отсутствие доступной памяти, доступ к закрытой области памяти и т.п. в языках программирования предусматривается специальный механизм, который называется обработка .... ».

Задание №10. Запишите название определения: «... - способность программного обеспечения быть перенесенным из одного окружения в другое».

### **Формируемая компетенция ПК 2.3**

#### **Перечень заданий закрытого типа**

Задание № 1. Какую из перечисленных возможностей предоставляет метод разработки тестов «белым ящиком»?

- а) выполнить полный охват кода;
- б) проверить работу на некорректных данных;
- в) тестировать с точки зрения пользователя;
- г) проверить работу сложных объектов;

Задание № 2. Что означает ключевое слово `static` в сигнатуре предложенного метода `test` ()?

```
public static void test ()  
{  
    Console.WriteLine("Вывод данных в консоль");  
}
```

- а) функция принадлежит конкретному объекту;
- б) функция принадлежит всему классу, а не конкретным объектам;
- в) вложенная функция;
- г) глобальная функция.

Задание № 3. Какую задачу позволяет решить способ разработки тестов «черным ящиком»?

- а) проверка потоков данных;
- б) отслеживание целостности данных;
- в) создание тестов параллельно с кодом;
- г) проверка внутренних граничных точек.

Задание № 4. Какой из модификаторов доступа методов наиболее ограничительный, допускающий доступ к методу только из содержащего его класса или структуры

- а) `public`;
- б) `private`;
- в) `protected`;
- г) `internal`.

Задание № 5. Выберите из предложенного списка признаки командных принципов работы на проекте ПО:

- а) несовместимость характеров членов команды.
- б) межличностное взаимодействие,
- в) несбалансированное распределение задач и ответственностей.

- г) недостаточная поддержка со стороны руководителя команды.
- д) гибкость и мобильность функций;

Задание №6. Какие существуют общие виды систем контроля версий;

- а) локальные и глобальные;
- б) внешние и внутренние;
- в) централизованные и распределенные;
- г) нет правильного ответа.

Задание №7. Сопоставьте названия и краткие характеристики методов отладки программного кода.

1. Метод индукции	вначале формируют множество причин, которые могли бы вызвать данное проявление ошибки, затем анализируя причины, исключают те, которые противоречат имеющимся данным. Если все причины исключены, то следует выполнить дополнительное тестирование исследуемого фрагмента;
2. Метод обратного прослеживания	основан на тщательном анализе симптомов ошибки, которые могут проявляться как неверные результаты вычислений или как сообщение об ошибке;
3. Метод дедукции	для точки вывода неправильного результата строится гипотеза о значениях основных переменных, которые могли бы привести к получению имеющегося результата. Далее, исходя из этой гипотезы, делают предложения о значениях переменных в предыдущей точке. Процесс продолжают, пока не обнаружат причину ошибки.

Задание № 8. Сопоставьте названия и формулировки типов программных ошибок.

1. defect	наблюдаемое нарушение требований, проявляющееся при каком-то реальном сценарии работы ПО;
2. failure	ошибка в коде программы, вызывающая нарушения требований при работе, то место, которое надо исправить;
3. fault	самое общее нарушение каких-либо требований или ожиданий, не обязательно проявляющееся вовне (например, нарушения стандартов кодирования, недостаточная гибкость системы и пр.);

Задание № 9. Установите правильную последовательность ключевых слов при объявлении

метода в проекте программы на C#?

- а) static указатель, что метод является статическим членом класса;
- б) идентификатор (имя метода) StartService;
- в) модификатор доступа public;
- г) тип возвращаемого значения bool;
- д) параметр (аргумент) метода string serviceName.

Задание №10. Установите правильную последовательность характеристик номера версии программного обеспечения:

- а) минорная версия (minor subversion, промежуточная версия) программного обеспечения;
- б) мажорная версия (major version) программного обеспечения;
- в) сборка (build) программного обеспечения;
- г) релиз (release) программного обеспечения.

### Перечень заданий открытого типа

Задание № 1. Как называются специальные логические операторы с коротким программным кодом, которые применяются для приема входных данных нижними модулями от модулей верхнего уровня при интеграционном тестировании?

Задание № 2. Как называется метод, нацеленный на выявление нарушений формализованных правил построения исходного кода ПО и заключающийся в целенаправленном поиске определенных дефектов и нарушений требований в коде на основе набора шаблонов?

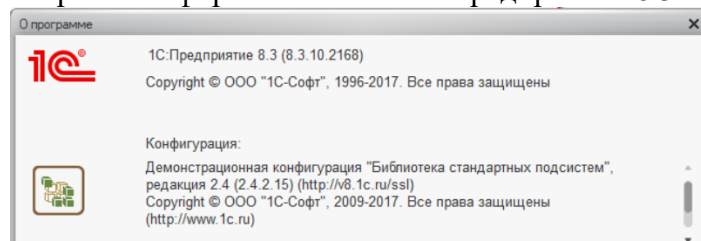
Задание № 3. Как называется программа, предназначенная для поиска ошибок в других программах и позволяет выполнять трассировку, отслеживать, устанавливать или изменять значения переменных в процессе выполнения кода, устанавливать и удалять контрольные точки или условия остановки и т. д.?

Задание № 4. Как называется фиксированное состояние реализации программного продукта на конкретную дату, выполненной в соответствии с техническим заданием заказчика, которому присваивается символическое обозначение в виде номера?

Задание № 5. При каком изменении функциональности программного продукта происходит изменение номера его мажорной версии?

Задание №6. Как называется версия программного продукта с любыми изменениями в функционале (удаление устаревших функций, расширение семантики компонентов и т. д.), которые позволяют упростить реализацию какого-либо программного решения?

Задание № 7. Определите по предложенному рисунку с информацией о программе номер сборки платформы системы 1С:Предприятие 8.3?



Задание №8. Как называется структурированный набор требований/запросов (функциональность, производительность, конструктивные ограничения и атрибуты) к программному обеспечению и его внешним интерфейсам?

Задание №9. Допишите предложение: «Номерами релизов обозначаются ... исправлений ошибок».

Задание №10. Допишите предложение: «Если в этом блоке оператора try не возникло исключений, то после его выполнения начинает выполняться блок ....».

**Формируемая компетенция: ПК 2.4**

**Перечень заданий закрытого типа**

Задание №1. Как называется численное значение критерия качества, определяющее степень, в которой программе присуще определенное критерием свойство?

- а) метрика;
- б) оценочный элемент;
- в) показатель качества;
- г) фактор качества.

Задание №2. Как называется процесс анализа или эксплуатации программного обеспечения с целью выявления дефектов?

- а) тестирование;
- б) испытание;
- в) исследование;
- г) валидация.

Задание №3. Для какого процесса проверки и анализа качества программного средства создается представительная комиссия из экспертов, представителей заказчика и представителей разработчика?

- а) верификации;
- б) аттестации;
- в) валидации;
- г) эксплуатации.

Задание №4. Как называется событие при выполнении программы, которое приводит к ее ненормальному или неправильному поведению?

- а) исключение;
- б) дефект;
- в) семантическая ошибка;
- г) логическая ошибка.

Задание №5. К какому виду исключений относится попытка деления на 0?

- а) программным;
- б) аппаратным;
- в) эксплуатационным;
- г) алгоритмическим.

Задание №6. С какого ключевого слова начинается оператор обработки исключительной ситуации в языке программирования 1С?

- а) Пока;
- б) эксерт;
- в) Попытка;
- г) try;

Задание № 7. Сопоставьте названия и назначение блоков оператора обработки исключений в Python.

Название блока	Назначение
1. try	а) запуск кода всегда

2. except	б) запуск кода, если не было исключений
3. else	в) запуск кода
4. finally	г) запуск кода, если возникло исключение

Задание № 8. Сопоставьте названия исключений и фрагменты программного кода с исключениями.

Название исключения	Программный код с исключением
1. SyntaxError	а) <pre>for i in range(10):     print('Привет Мир!')</pre>
2. TypeError	б) <pre>a = 2 b = 0 c = a/b</pre>
3. IndentationError	в) <pre>a = 2 b = 3 c = a b</pre>
4. ZeroDivisionError	г) <pre>a = 2 b = 'PythonRu' c = a + b</pre>

Задание №9. Установите правильную последовательность этапов процесса компиляции программы.

- а) оптимизация;
- б) лексический анализ;
- в) генерация кода;
- г) семантический анализ;
- д) синтаксический анализ;

Задание №10. Установите правильную последовательность процесса разработки тестового сценария.

- а) Определение цели;
- б) Описание ожидаемых результатов;
- в) Документирование;
- г) Описание шагов;
- д) Понимание требований.

### Перечень заданий открытого типа

Задание № 1. Какая отладочная функция Python проверяет условие на истинность, и если условие истинно, то код работает дальше, а если ложно, то возбуждает исключение типа AssertionError?

Задание №2. К какой группе исключений в Python относятся исключения ZeroDivisionError (Ошибка деления на ноль) FloatingPointError (Ошибка плавающей точки) и OverflowError (Ошибка переполнения)?

Задание № 3. В какой блок нужно заключить программный код на языке Python, если он может привести к исключению при выполнении?

Задание №4. Какое сообщение увидит пользователь при выполнении данного фрагмента программы?

Попытка

Попытка

a = 1/0;

Исключение

Сообщить("Конкретная ошибка");

КонецПопытки;

Исключение

Сообщить("Общая ошибка");

КонецПопытки;

Задание №5. Какой текст ошибки платформа 1С выдаст, если в имени переменной допущена опечатка?

Задание №6. Какой параметр сложности интеграции определяется как несовместимость форматов обмена данными, протоколов взаимодействия и интерфейсов программ?

Задание №7. При какой технике тестирования у тестировщика есть доступ к разрабатываемому коду?

Задание №8. Какой вид тестирования следует применить в первую очередь после выхода новой версии программы?

Задание №9. Допишите название определения: «... - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения».

Задание №10. Допишите название определения: «... – это автономный элемент программного обеспечения, предназначенный для многократного использования, который может распространяться для использования в других программах в виде скомпилированного кода».

**Формируемая компетенция: ПК 2.5**

### **Перечень заданий закрытого типа**

Задание №1. Как называется характеристика программного обеспечения, демонстрирующая то, что продукт удовлетворяет установленным требованиям или превышает их?

а) полнота;

б) качество;

в) стандарт;

г) алгоритмическая сложность.

Задание №2. Как называется единый стандарт кодирования, который применим к объектно-ориентированному языку программирования Python?

- а) PEP 8;
- б) UTF-8;
- в) Latin-1;
- г) ISO-8.

Задание №3 На каком этапе компиляции программы выполняется удаление излишних конструкций и упрощение кода с сохранением его смысла.

- а) лексического анализа;
- б) оптимизации;
- в) генерации кода;
- г) семантического анализа.

Задание №4. Какое качество задается требованиями заказчика в спецификациях и отражается в характеристиках конечного программного продукта?

- а) внутреннее;
- б) при использовании в процессе эксплуатации;
- в) внешнее;
- г) эксплуатационное.

Задание № 5. Как называется инструмент, который используется для отслеживания, внесения и управления изменениями в программном коде?

- а) схема контроля версий;
- б) схема компоновки данных;
- в) система компоновки данных;
- г) система контроля версий.

Задание №6. Как называется официально выпущенная версия программного продукта, предназначенная для распространения и использования?

- а) серия;
- б) релиз;
- в) реализация;
- г) коробочное решение.

Задание № 7. Сопоставьте описание и роли участников процесса инспектирования программного обеспечения.

1. Автор;	находит ошибки, упущения и противоречия в программах и документах;
2. Инспектор;	излагает код или документ на собрании инспекционной группы;
3. Рецензент;	отвечает за создание программы или документа, несет ответственность за исправление дефектов, обнаруженных в процессе инспектирования;
4. Секретарь;	управляет и организует процесс инспектирования, докладывает о результатах инспектирования руководству компании;
5. Председатель или координатор.	записывает результаты собрания инспекционной группы;

Задание № 8. Сопоставьте класс программных ошибок и вопросы, помогающие их выявить.

Класс ошибок	Вопросы, помогающие выявлять ошибки
1. Ошибки данных;	Выполняются ли условия для каждого условного оператора? Все ли циклы завершаются? Правильно ли в составных операторах расставлены скобки?
2. Ошибки управления;	Все ли переменные в программе инициализированы до начала использования их значений? Все ли константы именованы? Возможно ли переполнение буфера?
3. Ошибки интерфейса;	Используются ли в программе входные переменные? Все ли выходным переменным перед выводом присваиваются значения? Могут ли какие-нибудь входные данные привести к нарушению системных данных?
4. Ошибки ввода-вывода.	Все ли вызовы процедур и функций содержат правильное количество параметров? Согласованы ли типы формальных и фактических параметров? В правильном ли порядке расположены параметры?

Задание №9. Установите правильную последовательность этапов процесса проектирования ПО.

- а) обобщенная спецификация;
- б) проектирование интерфейсов;
- в) архитектурное проектирование;
- г) проектирование алгоритмов;
- д) проектирование структур данных;
- е) компонентное проектирование.

Задание №10. Установите правильную последовательность уровней групповой работы над разработкой ПО.

- а) бизнес-подразделение;
- б) команда;
- в) разработчик;
- г) компания;
- д) проект.

### Перечень заданий открытого типа

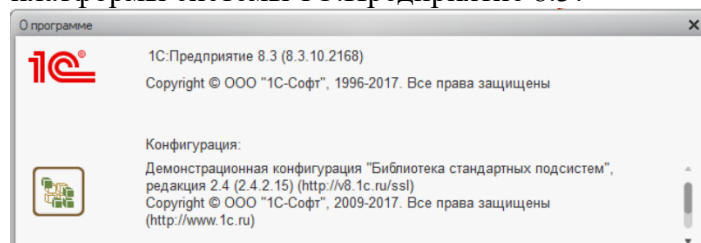
Задание № 1. Как называется точное и, по возможности, полное описание назначения программы, её функций, требований на входные данные, её результатов, особенностей функционирования и нефункциональных свойств, таких как: надёжность, отказоустойчивость, требований к ресурсам, и описание состава, структуры программы и её интерфейсов?

Задание №2. Какое требование описано в PEP 8 по оформлению Python-кода, с помощью которого задают структуру условий, циклов, функций?



Задание № 3. Как называют существенно изменённую программу или пакет программ, в который добавлены принципиально новые функции, используется иная организация программы, данных или применяются новые способы взаимодействия пользователя с программой.

Задание № 4. Определите по предложенному рисунку с информацией о программе релиз платформы системы 1С:Предприятие 8.3?



Задание №5. На каком этапе процесса проектирования ПО определяются и документируются подсистемы программы и взаимосвязи между ними?

Задание №6. К каким автоматизированным средствам разработки программного обеспечения относятся редакторы проектов, словари данных, компиляторы, отладчики, средства построения систем и т.п.?

Задание № 7. Как называется группа специалистов, объединенных общей целью создания и совершенствования программных продуктов;

Задание № 8. Какие специалисты в команде разработчиков делятся на front-end и back-end разработчиков?

Задание № 9. Допишите название определения: « .... программы - это превращение её исходного кода, написанного на каком-нибудь компилируемом языке программирования, который понятен программисту, в бинарный код (последовательность нулей и единиц), который понятен центральному процессору компьютера».

Задание №10. Допишите определение: «...интеграция заключается в том, что код разрабатывается и тестируется малыми компонентами, которые затем постепенно собираются в единое целое».

## 5. КРИТЕРИИ ОЦЕНКИ

В ФГБОУ ВО «ДГТУ» для обучающихся по образовательным программам среднего профессионального образования применяется пятибалльная шкала знаний, умений, практического опыта.

Таблица 3

<b>Шкалы оценивания</b>		<b>Критерии оценивания</b>
<b>пятибалльная</b>	<b>зачет</b>	
«Отлично» - 5 баллов	зачет	<p>Показывает высокий уровень сформированности компетенций, т.е.:</p> <ul style="list-style-type: none"> <li>– демонстрирует глубокое и прочное освоение материала;</li> <li>– исчерпывающе, четко, последовательно, грамотно и логически стройно излагает теоретический материал;</li> <li>– правильно формирует определения;</li> <li>– демонстрирует умения самостоятельной работы с нормативно-правовой литературой;</li> <li>– умеет делать выводы по излагаемому материалу.</li> </ul>
«Хорошо» - 4 балла	зачет	<p>Показывает достаточный уровень сформированности компетенций, т.е.:</p> <ul style="list-style-type: none"> <li>– демонстрирует достаточно полное знание материала, основных теоретических положений;</li> <li>– достаточно последовательно, грамотно логически стройно излагает материал;</li> <li>– демонстрирует умения ориентироваться в нормативно-правовой литературе;</li> <li>– умеет делать достаточно обоснованные выводы по излагаемому материалу.</li> </ul>
«Удовлетворительно» - 3 балла	Зачтено	<p>Показывает пороговый уровень сформированности компетенций, т.е.:</p> <ul style="list-style-type: none"> <li>– демонстрирует общее знание изучаемого материала;</li> <li>– испытывает затруднения при ответах на дополнительные вопросы;</li> <li>– знает основную рекомендуемую литературу;</li> <li>– умеет строить ответ в соответствии со структурой излагаемого материала.</li> </ul>
«Неудовлетворительно» - 2 балла	Не зачтено	<p>Ставится в случае:</p> <ul style="list-style-type: none"> <li>– незнания значительной части программного материала;</li> <li>– не владения понятийным аппаратом дисциплины;</li> <li>– допущения существенных ошибок при изложении учебного материала;</li> <li>– неумения строить ответ в соответствии со структурой излагаемого вопроса;</li> <li>– неумения делать выводы по излагаемому материалу.</li> </ul>

## Критерии оценки тестовых заданий

Таблица 4

<b>Процент выполненных тестовых заданий</b>	<b>Оценка</b>
до 50%	неудовлетворительно
50-69%	удовлетворительно
70-84%	хорошо
85-100%	отлично

## КЛЮЧИ К ЗАДАНИЯМ

Таблица 5

Формируемые компетенции	№ задания	Ответ
ПК 2.1	Задания закрытого типа	
	№ 1	в
	№ 2	в
	№ 3	а
	№ 4	б
	№ 5	а
	№ 6	б
	№ 7	1-б, 2-в, 3-а
	№ 8	1-б,2-а, 3-г,4-в
	№ 9	багв
	№ 10	гвба
	Задания открытого типа	
	№ 1	функциональному
	№ 2	using Diagnostics
	№ 3	Release-конфигурации
	№ 4	(2,0)
	№ 5	многоугольник, с целочисленными координатами вершин.
	№ 6	структурное
	№ 7	покрытия операторов
	№ 8	прототип
№ 9	системном	
№ 10	классы объектов	
ПК 2.2	Задания закрытого типа	
	№ 1	а
	№ 2	в
	№ 3	б
	№ 4	г
	№ 5	а
	№ 6	в
	№ 7	1-б, 2-в, 3-а
	№ 8	1-в,2-а,3-б
	№ 9	бгавд
	№ 10	дбагв
	Задания открытого типа	
	№ 1	HTTP и SMTP
	№ 2	обработка без возврата
	№ 3	модельная проверка
	№ 4	стабильность
	№ 5	б
	№ 6	int number = Convert.ToInt32(str)
	№ 7	программный интерфейс
	№ 8	throw
№ 9	исключений	

	№ 10	мобильность
ПК 2.3	Задания закрытого типа	
	№ 1	а
	№ 2	б
	№ 3	в
	№ 4	б
	№ 5	б,д
	№ 6	в
	№ 7	1-б, 2-в,3-а
	№ 8	1-в, 2-а, 3-б
	№ 9	вагбд
	№ 10	багв
	Задания открытого типа	
	№ 1	заглушки
	№ 2	инспектирование кода
	№ 3	отладчик
	№ 4	версия
	№ 5	глобальном
	№ 6	минорная
	№ 7	2168
	№ 8	спецификация
№ 9	выходы	
№ 10	finally	
ПК 2.4	Задания закрытого типа	
	№ 1	в
	№ 2	а
	№ 3	б
	№ 4	а
	№ 5	б
	№ 6	в
	№ 7	1-в, 2-г, 3-б, 4-а
	№ 8	1-в, 2-г, 3-а, 4-б
	№ 9	бдгав
	№ 10	дагбв
	Задания открытого типа	
	№ 1	Assert
	№ 2	ArithmeticError (арифметические ошибки)
	№ 3	try
	№ 4	Конкретная ошибка
	№ 5	Переменная не определена
	№ 6	технологическая разница
	№ 7	метод белого ящика.
	№ 8	дымовое тестирование
№ 9	тест-план	
№ 10	программный компонент	
ПК 2.5	Задания закрытого типа	
	№ 1	б
	№ 2	а
	№ 3	б

	№ 4	в
	№ 5	г
	№ 6	б
	№ 7	1-в, 2-а, 3-б, 4-д, 5-г
	№ 8	1-б, 2-а, 3-г, 4-в
	№ 9	вабедг
	№ 10	вбдаг
	Задания открытого типа	
	№ 1	спецификация программы
	№ 2	отступы
	№ 3	версия
	№ 4	10
	№ 5	архитектурное проектирование
	№ 6	CASE
	№ 7	команда разработки по
	№ 8	программисты
	№ 9	сборка
	№ 10	пошаговая

**Критерии оценки тестовых заданий, заданий на дополнение, с развернутым ответом и на установление правильной последовательности**

Верный ответ - 2 балла.

Неверный ответ или его отсутствие - 0 баллов.

**Критерии оценки заданий на сопоставление**

Верный ответ - 2 балла

1 ошибка - 1 балл

более 1-й ошибки или ответ отсутствует - 0 баллов.