

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Первякова Светлана Валерьевна
Должность: Директор колледжа СКФУ в г. Ставрополе
Дата подписания: 18.06.2026 12:25:50
Уникальный программный ключ:
63a18c373da05c089b08fcc89c468af301900f17

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»
Колледж СКФУ в г. Ставрополе

УТВЕРЖДАЮ

и.о. директора института
перспективной инженерии, канд. тех. наук,
доцент Порохня А.А.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПМ.02 РАЗРАБОТКА И ИНТЕГРАЦИЯ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Специальность 09.02.11 Разработка и управление программным обеспечением

Форма обучения очная

Фонд оценочных средств по профессиональному модулю ПМ.02 Разработка и интеграция модулей программного обеспечения разработан на основании федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением, утвержденного приказом Министерства образования и науки Российской Федерации от 24 февраля 2025 года № 138.

Разработчики: Безпалько Е.Л.-А., Шелегеда Д.С., преподаватели колледжа СКФУ в г. Ставрополе

СОГЛАСОВАНО

Директор
ООО «МИКРОКАПСУЛА»



Д.А. Салманова

1. Паспорт фонда оценочных средств

1.1. Область применения

Фонд оценочных средств (далее - ФОС) предназначен для проверки результатов освоения вида деятельности (ВД) Осуществление интеграции программных модулей и составляющих его профессиональных и общих компетенций, образовательной программы СПО по специальности 09.02.07 Информационные системы и программирование.

ФОС разработан на основании ФГОС, образовательной программы СПО и рабочей программы профессионального модуля (далее - ПМ).

1.2. Планируемые результаты освоения профессионального модуля

Результатом освоения ПМ является готовность обучающегося к выполнению вида профессиональной деятельности «Осуществление интеграции программных модулей» и сформированность профессиональных и общих компетенций.

Формой аттестации по ПМ - экзамен, итогом которого является решение: «вид профессиональной деятельности освоен с оценкой «освоен /не освоен».

Форма проведения экзамена: выполнение заданий, которые проверяют сформированность общих и профессиональных компетенций.

2. Формы контроля и оценивания элементов профессионального модуля

Таблица 1

Элемент профессионального модуля	Форма контроля и оценивания	
	Промежуточная аттестация	Текущий контроль
МДК.02.01 Разработка, поддержка и тестирование программных модулей	Экзамен	Устный опрос (собеседование)
МДК.02.02 Осуществление интеграции программных модулей	Зачет, Экзамен	Устный опрос (собеседование)
МДК.02.03 Математическое моделирование	Экзамен	Устный опрос (собеседование)
МДК.02.04 Безопасность программного обеспечения	Экзамен	Устный опрос (собеседование)
УП 02.01 Учебная практика «Разработка и интеграция модулей программного обеспечения»	Дифференцированный зачет	Защита отчета по учебной практике
ПП 02.01 Производственная практика «Разработка и интеграция модулей программного обеспечения»	Дифференцированный зачет	Защита отчета по производственной практике
ПМ.02.01(К) Экзамен по модулю «Разработка и интеграция модулей программного обеспечения»	Экзамен по модулю	

3. Результаты освоения профессионального модуля

3.1. Оценка профессиональных и общих компетенций

В результате контроля и оценки по ПМ осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Таблица 2

Профессиональные	Показатели оценки результата
------------------	------------------------------

компетенции	
ПК 2.1.	Проектировать модули программного обеспечения.
ПК 2.2.	Разрабатывать модули программного обеспечения.
ПК 2.3.	Выполнять интеграцию модулей и компонентов программного обеспечения.
ПК 2.4.	Выполнять тестирование и отладку программного обеспечения.
ПК 2.5.	Осуществлять документирование программных модулей программного обеспечения.
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.
ОК 03.	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях.
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде.

3.2. Общие и (или) профессиональные компетенции, проверяемые дополнительно:
(не предусмотрено)

3.3. Требования к портфолио: (не предусмотрено)

3.4. Требования к курсовой работе (проекту): представлены в методических указаниях по выполнению курсовой работы (проекта).

4. Оценка освоения теоретического курса профессионального модуля

4.1 Оценочные средства текущего контроля успеваемости и критерии оценки

4.1.1 Вопросы к устному (собеседованию)

МДК. 02.01 Разработка, поддержка и тестирование программных модулей

Раздел 1. Разработка, поддержка и тестирование программных модулей

1. Перечислите основные этапы (фазы) жизненного цикла программного обеспечения. Кратко охарактеризуйте каждую.
2. Какие модели жизненного цикла ПО вы знаете? В чем преимущества и недостатки каскадной модели?
3. Что такое структурное программирование? Назовите его основные принципы.
4. Дайте определение понятиям «алгоритм» и «структура данных». Приведите примеры структур данных.
5. Что такое модульность в программировании? Каковы критерии выделения модулей?
6. Перечислите основные принципы объектно-ориентированного программирования. Раскройте суть каждого.
7. Объясните разницу между понятиями «класс» и «объект».
8. Что такое инкапсуляция? Как она реализуется в языках программирования? Зачем нужны модификаторы доступа?
9. В чем заключается полиморфизм? Приведите пример перегрузки методов или переопределения методов.

10. Что такое наследование? Что понимается под базовым и производным классом?
11. Объясните понятия «абстрактный класс» и «интерфейс». В чем их сходство и различие?
12. Какие инструменты используются для коллективной разработки ПО?
13. Что такое репозиторий? Опишите базовый цикл работы с Git.
14. Для чего нужны комментарии в коде? Какие существуют стандарты оформления комментариев?
15. Что такое «качество кода»? Какие метрики качества кода вы знаете?
16. Перечислите основные типы ошибок, возникающих в процессе разработки ПО.
17. Дайте определение понятию «тестирование». Какова главная цель тестирования?
18. Чем отличается верификация от валидации?
19. Опишите уровни тестирования: модульное, интеграционное, системное, приемочное.
20. В чем разница между тестированием методов «черного ящика» и «белого ящика»?
21. Перечислите основные виды функционального тестирования.
22. Что такое модульное (unit) тестирование? Какие фреймворки для этого используются?
23. Что такое регрессионное тестирование и когда оно применяется?
24. Что такое отладка? Какие методы и инструменты отладки вы знаете?
25. В чем заключается разница между отладкой и тестированием?
26. Что такое оптимизация кода? Каких видов она бывает?
27. Назовите основные «запахи кода», которые ухудшают качество программного модуля.
28. Что входит в понятие «сопровождение программного обеспечения»? Какие бывают виды сопровождения?
29. Какие виды документации на ПО вы знаете?
30. Опишите процесс исправления ошибки в уже работающем программном продукте.

МДК. 02.02 Осуществление интеграции программных модулей

Раздел 2 Осуществление интеграции программных модулей

1. Дайте определение понятию «интеграция программных модулей». Опишите основные цели и виды интеграции.
2. Сформулируйте определение репозитория проекта. Опишите классы уровней репозитория, которые вы знаете.
3. Что такое система управления версиями? Опишите основные принципы организации работы команды в системе контроля версий.
4. Опишите процесс выбора и настройки работы системы контроля версий (типы импортируемых файлов, фильтры, параметры импорта в репозиторий).
5. Какие современные технологии и инструменты интеграции программных модулей вы знаете? Дайте их краткую характеристику.
6. Перечислите и охарактеризуйте основные подходы к интегрированию программных модулей.
7. Что такое структура проекта? Назовите основные задачи структуризации и охарактеризуйте виды проектов.
8. Опишите процесс разработки модульной структуры проекта (например, с помощью диаграмм модулей).
9. Дайте определение понятия «качество программного обеспечения». Перечислите известные вам критерии оценки качества ПО.

10. Дайте определение свойств качественного программного обеспечения: надежность, эффективность, мобильность, модифицируемость. Поясните их назначение.
11. Что такое стандарты кодирования? Для чего они необходимы при интеграции и разработке ПО? .
12. Сформулируйте определение понятия «тестирование» применительно к интеграции модулей. Опишите методы и средства организации тестирования.
13. Перечислите и охарактеризуйте функциональные виды тестирования программного обеспечения.
14. Перечислите и охарактеризуйте виды тестирования производительности.
15. Что такое интеграционное тестирование? Чем оно отличается от модульного и системного тестирования? .
16. Опишите процесс разработки тестовых модулей и тестовых сценариев для проверки взаимодействия отдельных модулей.
17. Что такое регрессионное тестирование и какова его роль в процессе интеграции программных модулей? .
18. Дайте определение понятий «ручное» и «автоматизированное» тестирование. Расскажите об их преимуществах и недостатках.
19. Дайте определение понятий «отладка» и «локализация ошибки». Какие виды ошибок существуют? Дайте им характеристику.
20. Перечислите и охарактеризуйте основные методы и принципы отладки программного обеспечения.
21. В чем заключается разница между процессами тестирования и отладки программных модулей?
22. Что такое обработка исключительных ситуаций? Опишите инструменты среды разработки, которые для этого применяются.
23. Опишите инструментальные средства поддержки процесса документирования при разработке и интеграции ПО.
24. Какие виды документации на программное обеспечение вы знаете? Какие стандарты оформления документации применяются?
25. Расскажите про методы организации коллективной разработки ПО и работы в команде разработчиков.
26. Что такое API (программный интерфейс приложения)? Какую роль он играет в интеграции модулей?
27. Опишите принцип работы протоколов транспортного уровня при взаимодействии программных компонентов.
28. Что такое графический интерфейс пользователя? В чем заключаются особенности его тестирования?
29. Расскажите об инструментарии анализа качества программных продуктов в среде разработки.
30. Какие методы и способы идентификации сбоев и ошибок в программных компонентах вы знаете?

МДК 02.03 Математическое моделирование

Раздел 3 Математическое моделирование

1. Понятие модели. Свойства моделей. Моделирование как метод познания.
2. Охарактеризуйте основные стадии моделирования.
3. Приведите и охарактеризуйте основные этапы построения модели.
4. Дайте определение линейного программирования.
5. Приведите математическую модель линейного программирования.
6. Приведите примеры экономических задач, приводящих к задачам линейного

программирования.

7. Сформулируйте постановку транспортных задач.
8. Представьте математическую модель транспортной задачи.
9. Перечислите и дайте характеристику этапов моделирования в линейном программировании.
10. Представьте математическую модель задачи нелинейного программирования
11. Опишите метод множителей Лагранжа.
12. Дайте понятие временных рядов.
13. Приведите виды зависимостей в трендовых моделях.
14. Опишите методику использования трендовых моделей.
15. Охарактеризуйте функциональные и стохастические связи.
16. Опишите методику использования регрессионных моделей.
17. Приведите виды зависимостей в регрессионных моделях.
18. Дайте понятие мультиколлинеарности.
19. Приведите показатели, которые используются при оценке статистической значимости трендовых и регрессионных моделей
20. Классификация математических моделей по основанию в зависимости от: сложности объекта моделирования. Примеры сложных и простых объектов моделирования.
21. Оператор модели. Классификация математических моделей по основанию в зависимости от оператора модели. Примеры.
22. Параметры модели. Классификация математических моделей по основанию в зависимости от параметров модели. Примеры.
23. Цели моделирования. Классификация математических моделей по основанию в зависимости от целей моделирования. Примеры.
24. Классификация математических моделей по основанию в зависимости от методов реализации модели. Примеры.
25. Исследование объекта моделирования. Описательная (концептуальная) постановка задачи моделирования.
26. Математическое формулирование задачи моделирования. Примеры. Задача Коши
27. Вариационные принципы построения моделей.
28. Моделирование процессов теплопроводности.
29. Общая характеристика пакета Maple.
30. Дайте определение имитационного моделирования.
31. Охарактеризуйте статистическое моделирование (метод Монте-Карло).
32. Приведите классификацию имитационных моделей.
33. Перечислите технологические этапы имитационного моделирования и дайте их краткую характеристику.
34. Опишите процессы планирования и проведения вычислительного эксперимента.
35. Укажите назначение и составляющие систем моделирования.
36. Укажите назначение языков моделирования.
37. Приведите классификацию языков моделирования.
38. Охарактеризуйте язык моделирования GPSS.
39. Укажите особенности моделирования систем массового обслуживания.
40. Охарактеризуйте возможности использования MATLAB/Simulink для имитационного моделирования.

МДК 02.04 Безопасность программного обеспечения

Раздел 4 Безопасность программного обеспечения

1. Дайте определение понятию «безопасность программного обеспечения». Почему она является критически важной на всех этапах жизненного цикла разработки ПО?
2. Перечислите и кратко охарактеризуйте основные свойства (триада CIA)

информационной безопасности, которые должно обеспечивать программное обеспечение.

3. Что такое модель угроз? Для чего она необходима при проектировании безопасного приложения?

4. Опишите методологию моделирования угроз STRIDE. Какие типы угроз она помогает выявить? Приведите примеры .

5. В чем разница между понятиями «уязвимость», «угроза» и «риск» в контексте безопасности ПО?

6. Что такое статический анализ безопасности приложений (SAST)? На каком этапе разработки он применяется и какие проблемы позволяет обнаружить?

7. Что такое динамический анализ безопасности приложений (DAST)? Чем он отличается от SAST и для каких целей используется?

8. Объясните назначение анализа состава программного обеспечения (SCA). Почему важно отслеживать уязвимости в open-source компонентах?

9. Что такое тестирование на проникновение (пентест)? Какие существуют виды пентеста (внешний, внутренний, black box, white box)?

10. Чем отличается пентест от анализа защищенности и от red team-мероприятия?

11. Что такое SQL-инъекция? Опишите механизм этой атаки и основные способы защиты от нее (параметризованные запросы, ORM) .

12. Опишите уязвимость межсайтового скриптинга (XSS). Какие существуют типы XSS и к каким последствиям может привести их эксплуатация?

13. Что такое уязвимость межсайтовой подделки запроса (CSRF)? Как работает эта атака и какие методы защиты применяются?

14. Объясните понятие «небезопасная прямая ссылка на объект» (IDOR). Приведите пример такой уязвимости и способы ее предотвращения .

15. Какие существуют методы безопасного хранения паролей пользователей? Почему нельзя хранить пароли в открытом виде или с простым хешированием (MD5, SHA)?

16. Что такое криптография и для каких целей она применяется в разработке ПО? Объясните разницу между симметричным и асимметричным шифрованием .

17. Что такое управление доступом? Опишите модели дискреционного (DAC), мандатного (MAC) и ролевого (RBAC) управления доступом.

18. Какие принципы безопасного проектирования ПО вы знаете? (например, минимизация привилегий, defense in depth, fail secure)

19. Объясните принцип «безопасный отказ» (fail secure / fail safe). Приведите пример его реализации .

20. Что такое проверка входных данных (input validation)? Почему она необходима и какие данные следует проверять?

21. Какие инструменты и практики обеспечения безопасности применяются в CI/CD пайплайнах (DevSecOps)?

22. Что такое «безопасность цепочки поставок ПО» (software supply chain security)? Какие риски с этим связаны?

23. Опишите процесс реагирования на инциденты информационной безопасности. Назовите его основные этапы .

24. Что такое управление уязвимостями (vulnerability management)? Опишите жизненный цикл управления уязвимостями.

25. Для чего проводится security code review (анализ кода на безопасность)? Что ищут в коде при таком анализе?

26. Назовите и кратко охарактеризуйте известные стандарты и методологии в области безопасности ПО (например, OWASP Top 10, BSIMM, SAMM) .

27. Что такое «принцип наименьших привилегий» (principle of least privilege)? Как он применяется при проектировании архитектуры и написании кода?

28. Какие существуют методы обеспечения безопасности при передаче данных по сети? Что такое TLS и для чего он используется?

29. Расскажите об особенностях обеспечения безопасности мобильных приложений. Чем они отличаются от безопасности веб-приложений?

30. Какие существуют законодательные и регуляторные требования к безопасности ПО в Российской Федерации (например, 152-ФЗ о персональных данных, требования ФСТЭК)?

Критерии оценки:

Оценка «отлично»: студент владеет знаниями предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивал при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное: устанавливать причинно-следственные связи. Четко формулирует ответы.

Оценка «хорошо»: студент владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах). Самостоятельно и отчасти при наводящих вопросах дает полноценные ответы, не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах.

Оценка «удовлетворительно»: студент владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками. В процессе ответов допускаются ошибки по существу вопросов.

Оценка «неудовлетворительно»: студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал, отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.

4.2. Оценочные средства для промежуточной аттестации и критерии оценки

4.2.1 Комплект тестовых заданий для проведения промежуточной аттестации и критерии оценки

4.2.1.1. Комплект тестовых заданий для проведения экзамена по МДК. 02.01 Разработка, поддержка и тестирование программных модулей

1. Какое из перечисленных ниже программных средств чаще всего используется для отладки в языке программирования C++?
а) Eclipse
б) Visual Studio Debugger
в) PyCharm
г) IntelliJ IDEA
Ответ: б) Visual Studio Debugger
2. Какой инструмент предназначен для трассировки выполнения кода и анализа значений переменных в процессе отладки?
а) Profiler
б) Breakpoint Manager
в) Debugger
г) Code Analyzer
Ответ: в) Debugger
3. Каким образом устанавливаются точки останова (breakpoints) в коде при использовании средства отладки?
а) Вручную добавляются командой в коде.
б) Автоматически устанавливаются на строки с ошибками.
в) Создаются автоматически при запуске отладки.
г) Вызываются из контекстного меню в процессе выполнения.

Ответ: а) Вручную добавляются командой в коде.

4. Какой инструмент используется для анализа утечек памяти в программном модуле в процессе отладки?
- а) Memory Profiler
 - б) Code Inspector
 - в) Leak Detector
 - г) Resource Monitor

Ответ: а) Memory Profiler

5. Какие из перечисленных инструментов могут быть использованы для отладки в среде разработки на языке Java?
- а) GDB
 - б) Xcode Debugger
 - в) Eclipse Debugger
 - г) Visual Studio Code

Ответ: в) Eclipse Debugger

6. Какой вид информации обычно выводится в окне "Watch" в процессе отладки?
- а) Значения переменных и выражений, выбранных для отслеживания.
 - б) Список доступных функций отладчика.
 - в) Журнал событий отладки.
 - г) Список ошибок и предупреждений.

Ответ: а) Значения переменных и выражений, выбранных для отслеживания.

7. Каким образом можно оценить производительность программного модуля в процессе отладки?
- а) Анализом логов.
 - б) Использованием Memory Profiler.
 - в) Запуском профилировщика кода.
 - г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

8. Какой инструмент обычно используется для анализа исходного кода и выявления потенциальных проблем во время отладки?
- а) Code Inspector
 - б) Profiler
 - в) Breakpoint Manager
 - г) Debugger

Ответ: а) Code Inspector

9. Какой тип инструмента может использоваться для отладки в асинхронных приложениях, таких как веб-серверы?
- а) Concurrency Profiler
 - б) Code Analyzer
 - в) Debugger
 - г) Network Monitor

Ответ: а) Concurrency Profiler

10. Каким образом можно управлять выводом отладочных сообщений в коде, чтобы они не засоряли логи в продакшене?
- а) Использовать более мощный отладочный уровень.
 - б) Внедрять условные конструкции в коде для временного отключения логирования.
 - в) Заменять отладочные сообщения на пустые строки перед публикацией кода.
 - г) Все вышеперечисленное.

Ответ: б) Внедрять условные конструкции в коде для временного отключения логирования.

11. Какие инструменты могут использоваться для отладки в многозадачных сценариях, таких как приложения с использованием многопоточности или параллелизма?

- а) Thread Profiler
- б) Breakpoint Manager
- в) Memory Analyzer
- г) Code Inspector

Ответ: а) Thread Profiler

12. Какие из перечисленных инструментов могут быть применены для отладки в среде разработки на языке Python?

- а) Visual Studio Code
- б) Xcode Debugger
- в) NetBeans Debugger
- г) IntelliJ IDEA

Ответ: а) Visual Studio Code

13. Какой инструмент часто используется для анализа времени выполнения кода и выявления узких мест в процессе отладки?

- а) Profiler
- б) Breakpoint Manager
- в) Code Inspector
- г) Debugger

Ответ: а) Profiler

14. Каким образом можно определить, что проблема в программном модуле связана с внешними зависимостями, такими как библиотеки или API?

- а) Анализ логов отладки.
- б) Профилирование кода.
- в) Проверка целостности файлов библиотек.
- г) Все вышеперечисленное.

Ответ: а) Анализ логов отладки.

15. Какой вид информации обычно предоставляется инструментами анализа памяти в процессе отладки?

- а) Использование CPU.
- б) Список загруженных библиотек.
- в) Выделение и освобождение памяти.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

16. Каким образом устанавливаются условные точки останова (conditional breakpoints) в коде при использовании средства отладки?

- а) При помощи дополнительных плагинов.
- б) Путем задания условий, при которых точка останова срабатывает.
- в) Автоматически при каждом запуске отладки.
- г) Через меню "Options" в отладчике.

Ответ: б) Путем задания условий, при которых точка останова срабатывает.

17. Какие инструменты могут быть использованы для анализа сетевого взаимодействия в процессе отладки веб-приложений?

- а) Network Profiler
- б) Memory Analyzer
- в) Breakpoint Manager
- г) Code Inspector

Ответ: а) Network Profiler

18. Каким образом управляется версионность исходного кода в процессе отладки?

- а) С помощью специализированных программ-версионирования.
- б) Вручную, через создание резервных копий перед отладкой.
- в) С использованием системы контроля версий.
- г) Все вышеперечисленное.

Ответ: в) С использованием системы контроля версий.

19. Каким образом отладка в продакшене может быть организована без прямого доступа к исходному коду?

- а) Использование отладочных логов и дампов состояния.
- б) Установка точек останова в байт-коде.
- в) Использование аппаратных отладочных средств.
- г) Все вышеперечисленное.

Ответ: а) Использование отладочных логов и дампов состояния.

20. Какие методы используются для воспроизведения проблемы в процессе отладки, особенно если она происходит не всегда или в определенных условиях?

- а) Запуск отладки на идентичной конфигурации.
- б) Использование реплицируемых сценариев тестирования.
- в) Создание автоматизированных средств воспроизведения.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

21. Какой инструмент может помочь в отладке кода, работающего в условиях ограниченного доступа к ресурсам, например, на встроенных системах?

- а) Code Inspector
- б) Memory Profiler
- в) Emulator
- г) Network Analyzer

Ответ: в) Emulator

22. Каким образом можно обеспечить безопасность исходного кода в процессе отладки, чтобы избежать случайного раскрытия конфиденциальной информации или утечек?

- а) Отключение логирования при отладке.
- б) Использование анонимизации данных в логах.
- в) Маскирование конфиденциальных данных в выводе.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

23. Каким образом отладка параллельного или распределенного кода может отличаться от отладки однопоточных приложений?

- а) Использование инструментов анализа параллельного выполнения.
- б) Работа с потоками и процессами.
- в) Мониторинг взаимодействия между компонентами.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

24. Как вы определяете точки входа для начала отладки программного модуля, особенно если проблема проявляется только в определенных сценариях использования?

- а) Запуск отладки сразу после запуска приложения.
- б) Использование данных от пользователей и логов ошибок.
- в) Создание реплицируемых сценариев тестирования.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

25. Каким образом вы оцениваете эффективность отладки и какие критерии вы используете для определения успешности процесса?

- а) Количество исправленных багов.
- б) Время нахождения и устранения ошибок.
- в) Обратная связь от тестировщиков и пользователей.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

26. Каким образом вы поддерживаете чистоту и структурированность кода в процессе отладки, чтобы избежать создания "костылей" или временных исправлений?

- а) Регулярное использование отладочных логов.
- б) Создание временных веток в системе управления версиями.
- в) Систематическое сохранение рабочих копий кода перед отладкой.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

27. Каким образом вы проверяете эффективность ваших изменений в программном модуле после отладки?

- а) Анализ отчетов от системы управления версиями.
- б) Создание регрессионных тестов.
- в) Мониторинг изменений производительности.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

28. Как вы справляетесь с ситуациями, когда проблема в программном модуле проявляется только в продуктивной среде, и отладка в локальной среде не выявляет проблемы?

- а) Использование инструментов мониторинга продакшен среды.
- б) Создание дампов состояния системы в продакшене.
- в) Тестирование с различными конфигурациями в продакшене.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

29. Каким образом вы управляете временными изменениями в коде, которые могут быть необходимы в процессе отладки, но не предназначены для окончательного включения в кодовую базу?

- а) Использование флагов функциональности.
- б) Документирование временных изменений для последующего удаления.
- в) Создание отдельных веток в системе управления версиями.
- г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

30. Каким образом вы определяете, что проблема в программном модуле связана с асинхронным или параллельным выполнением кода?

- а) Использование инструментов для анализа выполнения кода в многозадачных сценариях.
- б) Трассировка выполнения потоков.
- в) Анализ логов асинхронных операций.
- г) Все вышеперечисленное.

31. Что включает в себя процесс разработки требований к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент?

- а) Изучение проектной и технической документации.
- б) Выделение ключевых компонентов системы.
- в) Определение функциональности и взаимодействия компонент.
- г) Все вышеперечисленное.

32. Какие ключевые характеристики компонентов программной системы учитываются при разработке требований к их взаимодействию?

- а) Интерфейсы.
- б) Протоколы обмена данными.
- в) Частота взаимодействия.
- г) Все вышеперечисленное.

33. Как обеспечивается согласованность требований к взаимодействию компонент в различных частях документации?

- а) Следование стандартам документации.
- б) Использование единого языка и терминологии.
- в) Регулярные обзоры документации с командой разработки.

- г) Все вышеперечисленное.
34. Каким образом учитываются возможные изменения в требованиях к взаимодействию компонент в процессе разработки?
- а) Предусмотрение гибкости в требованиях.
 - б) Регулярное обновление требований на основе изменений в документации.
 - в) Взаимодействие с командой разработки для корректировки требований.
 - г) Все вышеперечисленное.
35. Как решаются конфликты в требованиях к взаимодействию компонент, возникающие между членами команды разработки?
- а) Открытый диалог.
 - б) Обсуждение противоречий.
 - в) Поиск компромиссов.
 - г) Все вышеперечисленное.
36. Как оценивается готовность и выполнимость требований к взаимодействию компонент в рамках текущего проекта?
- а) Анализ технической реализуемости.
 - б) Консультации с техническими специалистами.
 - в) Учет ресурсов и ограничений проекта.
 - г) Все вышеперечисленное.
37. Как внедряются требования к взаимодействию компонент в рабочий процесс разработки программного продукта?
- а) Постепенно, обеспечивая понимание всей командой.
 - б) Интеграция требований в рабочие процессы.
 - в) Контроль соответствия решений утвержденным требованиям.
 - г) Все вышеперечисленное.
38. Как проверяется соответствие реализации программных модулей требованиям к их взаимодействию?
- а) Тестирование, включая модульное и системное.
 - б) Использование инструментов статического анализа кода.
 - в) Внутренние проверки.
 - г) Все вышеперечисленное.
39. Как реагируете на изменения в проектной и технической документации, которые могут повлиять на требования к взаимодействию компонент?
- а) Оперативный анализ изменений.
 - б) Коррекция требований при необходимости.
 - в) Поддержание соответствия требований новым условиям.
 - г) Все вышеперечисленное.
40. Какая роль вашего анализа проектной и технической документации сыграла в успешном выявлении требований к взаимодействию компонент в проекте?
41. Что включает в себя процесс разработки требований к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент?
- а) Изучение проектной и технической документации.
 - б) Выделение ключевых компонентов системы.
 - в) Определение функциональности и взаимодействия компонент.
 - г) Все вышеперечисленное.
42. Какие ключевые характеристики компонентов программной системы учитываются при разработке требований к их взаимодействию?
- а) Интерфейсы.
 - б) Протоколы обмена данными.
 - в) Частота взаимодействия.
 - г) Все вышеперечисленное.
43. Как обеспечивается согласованность требований к взаимодействию компонент в

- различных частях документации?
- Следование стандартам документации.
 - Использование единого языка и терминологии.
 - Регулярные обзоры документации с командой разработки.
 - Все вышеперечисленное.
44. Каким образом учитываются возможные изменения в требованиях к взаимодействию компонент в процессе разработки?
- Предусмотрение гибкости в требованиях.
 - Регулярное обновление требований на основе изменений в документации.
 - Взаимодействие с командой разработки для корректировки требований.
 - Все вышеперечисленное.
45. Как решаются конфликты в требованиях к взаимодействию компонент, возникающие между членами команды разработки?
- Открытый диалог.
 - Обсуждение противоречий.
 - Поиск компромиссов.
 - Все вышеперечисленное.
46. Как оценивается готовность и выполнимость требований к взаимодействию компонент в рамках текущего проекта?
- Анализ технической реализуемости.
 - Консультации с техническими специалистами.
 - Учет ресурсов и ограничений проекта.
 - Все вышеперечисленное.
47. Как внедряются требования к взаимодействию компонент в рабочий процесс разработки программного продукта?
- Постепенно, обеспечивая понимание всей командой.
 - Интеграция требований в рабочие процессы.
 - Контроль соответствия решений утвержденным требованиям.
 - Все вышеперечисленное.
48. Как проверяется соответствие реализации программных модулей требованиям к их взаимодействию?
- Тестирование, включая модульное и системное.
 - Использование инструментов статического анализа кода.
 - Внутренние проверки.
 - Все вышеперечисленное.
49. Как реагируете на изменения в проектной и технической документации, которые могут повлиять на требования к взаимодействию компонент?
- Оперативный анализ изменений.
 - Коррекция требований при необходимости.
 - Поддержание соответствия требований новым условиям.
 - Все вышеперечисленное.
50. Какая роль вашего анализа проектной и технической документации сыграла в успешном выявлении требований к взаимодействию компонент в проекте?
- Выделение критических точек взаимодействия.
 - Эффективная интеграция компонентов.
 - Успешное завершение проекта.
 - Все вышеперечисленное.

Ключ к тесту

1	2	3	4	5	6	7	8	9	10
б)	в)	а)	а)	в)	а)	г)	а)	а)	б)

11	12	13	14	15	16	17	18	19	20
а)	а)	а)	а)	г)	б)	а)	в)	а)	г)
21	22	23	24	25	26	27	28	29	30
в)	г)	г)	г)	г)	г)	г)	г)	г)	г)
31	32	33	34	35	36	37	38	39	40
г)	г)	г)	г)	г)	г)	г)	г)	г)	г)
41	42	43	44	45	46	47	48	49	50
г)	г)	г)	г)	г)	г)	г)	г)	г)	г)

4.2.1.2. Комплект тестовых заданий для проведения экзамена МДК.02.02 Осуществление интеграции программных модулей

- Какой тип тестирования применим при создании тестовых сценариев для проверки функциональности платежной системы в интернет-магазине?
а) Тестирование производительности
б) Тестирование безопасности
в) Функциональное тестирование
г) Интеграционное тестирование
Ответ: в) Функциональное тестирование
- Что из перечисленного является основным шагом при создании тестового сценария для пользовательского интерфейса веб-приложения?
а) Написание кода для автоматизации теста
б) Определение шагов взаимодействия пользователя с интерфейсом
в) Проверка безопасности приложения
г) Создание тестовых данных
Ответ: б) Определение шагов взаимодействия пользователя с интерфейсом
- Какой из нижеперечисленных видов тестирования направлен на проверку работы приложения под различными нагрузками и условиями?
а) Модульное тестирование
б) Функциональное тестирование
в) Нагрузочное тестирование
г) Тестирование на соответствие требованиям
Ответ: в) Нагрузочное тестирование
- Что означает термин "регрессионное тестирование"?
а) Проверка, что новые изменения не вызывают отклонений от требований
б) Проверка безопасности системы
в) Тестирование производительности
г) Проверка функциональности отдельных модулей
Ответ: а) Проверка, что новые изменения не вызывают отклонений от требований
- Какой метод тестирования предназначен для проверки сценариев взаимодействия между различными компонентами программного продукта?
а) Тестирование производительности
б) Интеграционное тестирование
в) Функциональное тестирование
г) Модульное тестирование

Ответ: б) Интеграционное тестирование

6. Какова цель тестирования на отказ?

- а) Проверка, как система ведет себя при работе с различными объемами данных
- б) Проверка способности системы восстанавливаться после сбоев
- в) Тестирование безопасности
- г) Проверка соответствия системы требованиям

Ответ: б) Проверка способности системы восстанавливаться после сбоев

7. Какое тестирование направлено на определение эффективности работы программного продукта при различных нагрузках и условиях использования?

- а) Тестирование производительности
- б) Функциональное тестирование
- в) Тестирование совместимости
- г) Тестирование безопасности

Ответ: а) Тестирование производительности

8. Что включает в себя тестирование совместимости?

- а) Проверку на соответствие требованиям заказчика
- б) Тестирование работы программы на разных платформах и устройствах
- в) Проверку отказоустойчивости системы
- г) Тестирование производительности

Ответ: б) Тестирование работы программы на разных платформах и устройствах

9. Какой вид тестирования проверяет соответствие программы установленным требованиям и ожиданиям пользователей?

- а) Тестирование производительности
- б) Тестирование на соответствие требованиям
- в) Тестирование безопасности
- г) Интеграционное тестирование

Ответ: б) Тестирование на соответствие требованиям

10. Что представляет собой тестирование безопасности программного продукта?

- а) Проверка, как быстро работает система при высокой нагрузке
- б) Анализ уязвимостей и защищенности системы от внешних атак
- в) Тестирование взаимодействия между различными компонентами системы
- г) Проверка соответствия системы функциональным требованиям

Ответ: б) Анализ уязвимостей и защищенности системы от внешних атак

11. Какой метод тестирования направлен на проверку правильности работы отдельных модулей программного продукта?

- а) Интеграционное тестирование
- б) Модульное тестирование
- в) Тестирование производительности
- г) Тестирование безопасности

Ответ: б) Модульное тестирование

12. Какие данные необходимы для успешного проведения тестирования производительности?

- а) Тестовые сценарии и наборы данных
- б) Спецификации безопасности
- в) Инструкции пользователя
- г) Информация о конфигурации системы и ожидаемых нагрузках

Ответ: г) Информация о конфигурации системы и ожидаемых нагрузках

13. Какой вид тестирования оценивает, насколько хорошо программное обеспечение может быть адаптировано для использования на разных устройствах и в различных условиях?

- а) Тестирование производительности
- б) Тестирование на соответствие требованиям
- в) Тестирование совместимости

г) Тестирование на отказ

Ответ: в) Тестирование совместимости

14. Что включает в себя тестирование на отказ?

а) Проверку на соответствие требованиям заказчика

б) Тестирование производительности

в) Тестирование функциональности приложения при сбоях

г) Анализ уязвимостей системы

Ответ: в) Тестирование функциональности приложения при сбоях

15. Какой метод тестирования направлен на проверку функциональности и взаимодействия между компонентами системы в целом?

а) Модульное тестирование

б) Интеграционное тестирование

в) Тестирование на отказ

г) Тестирование производительности

Ответ: б) Интеграционное тестирование

16. Какова цель тестирования производительности?

а) Проверка, что система восстанавливается после сбоев

б) Оценка эффективности работы программного продукта под различными нагрузками

в) Тестирование на отказ

г) Проверка соответствия системы требованиям

Ответ: б) Оценка эффективности работы программного продукта под различными нагрузками

17. Что включает в себя тестирование на соответствие требованиям?

а) Проверка отказоустойчивости системы

б) Анализ уязвимостей и защищенности системы

в) Тестирование работы на разных платформах и устройствах

г) Проверка, что система соответствует установленным требованиям

Ответ: г) Проверка, что система соответствует установленным требованиям

18. Какой вид тестирования предназначен для выявления уязвимостей и защиты программного продукта от внешних атак?

а) Тестирование производительности

б) Тестирование на отказ

в) Тестирование безопасности

г) Тестирование совместимости

Ответ: в) Тестирование безопасности

19. Что включает в себя тестирование функциональности?

а) Проверку способности системы восстанавливаться после сбоев

б) Тестирование производительности приложения

в) Проверку соответствия системы установленным требованиям

г) Анализ уязвимостей и защищенности системы

Ответ: в) Проверку соответствия системы установленным требованиям

20. Какой вид тестирования оценивает работоспособность системы при высокой нагрузке и стремится выявить пределы ее возможностей?

а) Тестирование на отказ

б) Тестирование производительности

в) Тестирование функциональности

г) Тестирование на соответствие требованиям

Ответ: б) Тестирование производительности

21. Что означает термин "регрессионное тестирование"?

а) Проверка, как система ведет себя при работе с различными объемами данных.

б) Проверка безопасности системы.

в) Тестирование производительности.

г) Проверка, что новые изменения не вызывают отклонений от требований.

Ответ: г) Проверка, что новые изменения не вызывают отклонений от требований.

22. Какой вид тестирования оценивает, насколько хорошо программное обеспечение может быть адаптировано для использования на разных устройствах и в различных условиях?

а) Тестирование производительности.

б) Тестирование на соответствие требованиям.

в) Тестирование совместимости.

г) Тестирование на отказ.

Ответ: в) Тестирование совместимости.

23. Каким образом проводится тестирование на отказ?

а) Запуск тестов в условиях высокой нагрузки.

б) Проверка работы системы при сбоях или отказах.

в) Оценка производительности приложения.

г) Тестирование работы на различных платформах.

Ответ: б) Проверка работы системы при сбоях или отказах.

24. Какие данные необходимы для успешного проведения тестирования производительности?

а) Тестовые сценарии и наборы данных.

б) Спецификации безопасности.

в) Инструкции пользователя.

г) Информация о конфигурации системы и ожидаемых нагрузках.

Ответ: г) Информация о конфигурации системы и ожидаемых нагрузках.

25. Каким образом оценивается эффективность тестирования? Какие метрики могут использоваться для измерения качества тестирования?

а) Количество написанных тестов.

б) Процент выполненных тестов без ошибок.

в) Время, затраченное на написание тестовых сценариев.

г) Все вышеперечисленное.

Ответ: б) Процент выполненных тестов без ошибок.

26. Каким образом вы оцениваете риски и приоритеты при разработке тестовых сценариев? Какие критерии влияют на решение о приоритете того или иного тестового сценария?

а) Оценка риска включает в себя анализ важности функциональности, частоты использования, критичности для бизнеса и предыдущих проблем.

б) Критерии влияния на приоритет включают в себя критические для бизнеса сценарии, сценарии высокой сложности и сценарии, связанные с изменениями или добавлениями в функциональности.

в) Оба варианта.

Ответ: в) Оба варианта.

27. Каким образом вы обеспечиваете поддерживаемость ваших тестовых сценариев в долгосрочной перспективе? Как реагируете на изменения в требованиях или структуре программного продукта?

а) Писание модульных, читаемых и документированных тестов.

б) Регулярные ревью и обновление тестовых сценариев после изменений в продукте.

в) Использование фреймворков для автоматизации тестирования.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

28. Как вы оцениваете потребности в тестировании безопасности? Какие виды тестирования безопасности включаете в ваш процесс тестирования?

а) Оценка включает в себя анализ уязвимостей, оценку влияния уязвимостей на бизнес, соблюдение законодательных требований.

б) Включение методов тестирования безопасности, таких как сканирование уязвимостей, тестирование на отказ в обслуживании (DoS), и аудит безопасности кода.

в) Оба варианта.

Ответ: в) Оба варианта.

29. Каким образом вы взаимодействуете с разработчиками в процессе тестирования? Как обеспечиваете эффективное взаимодействие между QA и разработкой?

а) Регулярные совещания и обсуждения тестовых сценариев.

б) Ревью кода и тестов между QA и разработчиками.

в) Участие QA в планировании и ретроспективе разработочных итераций.

г) Использование общих инструментов для отслеживания дефектов и задач.

Ответ: г) Использование общих инструментов для отслеживания дефектов и задач.

30. Что включает в себя первоначальный этап интеграции модулей в программное обеспечение?

а) Анализ требований.

б) Разработка модульных тестов.

в) Изучение интерфейсов модулей.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

31. Какой метод используется для тестирования взаимодействия модулей в условиях контролируемой среды?

а) Модульное тестирование.

б) Интеграционное тестирование.

в) Системное тестирование.

г) Приемочное тестирование.

Ответ: б) Интеграционное тестирование.

32. Что представляет собой "stub" в контексте интеграции модулей?

а) Инструмент для автоматизации тестирования.

б) Заглушка, заменяющая вызовы других модулей.

в) Спецификация интерфейсов модулей.

г) Инструмент анализа кода.

Ответ: б) Заглушка, заменяющая вызовы других модулей.

33. Какой вид интеграции предполагает последовательное добавление и тестирование модулей до полной интеграции?

а) Вертикальная интеграция.

б) Горизонтальная интеграция.

в) Интеграция по этапам.

г) Интеграция "Big Bang".

Ответ: в) Интеграция по этапам.

34. Что означает термин "smoke testing" в контексте интеграции модулей?

а) Тестирование на дымовые испытания.

б) Быстрое поверхностное тестирование для выявления критических ошибок.

в) Тестирование дымовым образом.

г) Тестирование интеграции с соседними системами.

Ответ: б) Быстрое поверхностное тестирование для выявления критических ошибок.

35. Какова цель проведения "regression testing" в процессе интеграции модулей?

а) Проверка работы интегрированных модулей.

б) Выявление новых ошибок после интеграции.

в) Проверка обратной совместимости с предыдущими версиями.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

36. Какие виды зависимостей между модулями являются наиболее проблематичными при интеграции?

а) Сильные зависимости.

б) Слабые зависимости.

в) Циклические зависимости.

г) Однократные зависимости.

Ответ: в) Циклические зависимости.

37. Каким образом используются технологии "continuous integration" (непрерывная интеграция) при интеграции модулей?

а) Интеграция модулей происходит только в конце разработки.

б) Автоматизированные средства интеграции и тестирования применяются на каждом этапе разработки.

в) Интеграция модулей происходит вручную перед релизом.

г) Непрерывная интеграция не применяется в процессе разработки.

Ответ: б) Автоматизированные средства интеграции и тестирования применяются на каждом этапе разработки.

38. Что означает термин "integration build" в контексте интеграции модулей?

а) Сборка, используемая только для интеграции модулей.

б) Сборка, созданная после завершения интеграции модулей.

в) Сборка, включающая изменения отдельных разработчиков для проверки совместимости.

г) Все вышеперечисленное.

Ответ: в) Сборка, включающая изменения отдельных разработчиков для проверки совместимости.

39. Каким образом обеспечивается отслеживание версий исходного кода при интеграции модулей в системе управления версиями?

а) Использование только одной ветки разработки.

б) Ручное отслеживание изменений.

в) Использование различных веток для различных модулей и регулярное слияние.

г) Все вышеперечисленное.

Ответ: в) Использование различных веток для различных модулей и регулярное слияние.

40. Какие основные этапы включаются в процесс "user acceptance testing" (приемочное тестирование) при интеграции модулей?

а) Подготовка данных для тестирования.

б) Проведение тестовых сценариев, основанных на требованиях пользователей.

в) Анализ результатов тестирования.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

41. Что представляет собой "deployment" в контексте интеграции модулей в программное обеспечение?

а) Автоматическое тестирование.

б) Разработка новых функциональностей.

в) Процесс развертывания интегрированных модулей в рабочей среде.

г) Проверка совместимости с браузерами.

Ответ: в) Процесс развертывания интегрированных модулей в рабочей среде.

42. Что означает термин "roll-back" в контексте интеграции модулей?

а) Процесс отмены изменений и возврата к предыдущей версии.

б) Создание резервной копии данных перед интеграцией.

в) Использование нескольких версий одновременно.

г) Расширение функционала интегрированных модулей.

Ответ: а) Процесс отмены изменений и возврата к предыдущей версии.

43. Какие меры безопасности предпринимаются при интеграции модулей, работающих с конфиденциальными данными?

а) Шифрование данных.

б) Контроль доступа.

в) Аудит событий.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

44. Какой метод используется для автоматизации процесса интеграции и тестирования модулей при каждом изменении в коде?

а) Continuous Integration (непрерывная интеграция).

б) Big Bang Integration (интеграция "Big Bang").

в) Incremental Integration (постепенная интеграция).

г) Parallel Integration (параллельная интеграция).

Ответ: а) Continuous Integration (непрерывная интеграция).

45. Какова роль тестирования совместимости при интеграции модулей?

а) Проверка работоспособности модулей на различных платформах и браузерах.

б) Тестирование совместимости с другими программными продуктами.

в) Обеспечение корректной работы модулей в различных средах.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

46. Каким образом обеспечивается отслеживание изменений в коде модулей и их влияния на другие части системы при интеграции?

а) Ручное сравнение кода.

б) Использование системы управления версиями.

в) Автоматизированные средства отслеживания зависимостей.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

47. Что включает в себя процесс "dependency injection" при интеграции модулей?

а) Внедрение зависимостей между модулями.

б) Автоматическое разрешение зависимостей.

в) Передача внешних зависимостей в модуль.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

48. Каким образом обеспечивается обратная совместимость при интеграции новых модулей с существующей системой?

а) Регулярное обновление старых модулей.

б) Создание дополнительных интерфейсов для поддержки новых функциональностей.

в) Поддержка существующих API.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

49. Какие виды тестов эффективны для выявления скрытых проблем при интеграции модулей?

а) Юнит-тесты.

б) Функциональные тесты.

в) Стресс-тесты.

г) Тесты на случайные входные данные (fuzz testing).

Ответ: г) Тесты на случайные входные данные (fuzz testing).

50. Какова роль "mock objects" в тестировании интеграции модулей?

а) Заглушки, заменяющие реальные объекты в тестах для изоляции модулей.

б) Модули, предназначенные для подделки результатов тестов.

в) Инструменты для многократного использования кода в разных модулях.

г) Все вышеперечисленное.

Ответ: а) Заглушки, заменяющие реальные объекты в тестах для изоляции модулей.

51. Какие меры безопасности рекомендуется принимать при интеграции модулей для предотвращения утечек конфиденциальных данных?

а) Шифрование данных в памяти.

б) Ограничение доступа к конфиденциальной информации.

в) Аудит операций с конфиденциальными данными.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

52. Каким образом "sandboxing" (изоляция) используется при интеграции модулей для обеспечения безопасности?

а) Создание отдельной среды для тестирования модулей.

б) Ограничение доступа к ресурсам и данным.

в) Изоляция модулей друг от друга в отдельных контейнерах.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

53. Каким образом происходит разрешение конфликтов при интеграции модулей с изменениями в общей кодовой базе?

а) Автоматическое разрешение при помощи инструментов слияния кода.

б) Ручное разрешение конфликтов командой разработчиков.

в) Принятие изменений от того разработчика, кто первым внес изменения.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

54. Какие метрики можно использовать для оценки эффективности интеграции модулей в проекте?

а) Время, затраченное на интеграцию.

б) Количество выявленных дефектов после интеграции.

в) Процент покрытия тестами интегрированных модулей.

г) Все вышеперечисленное.

Ответ: г) Все вышеперечисленное.

55. Каким образом применяются "feature toggles" (переключатели функциональности) при интеграции новых модулей?

а) Автоматическое включение новой функциональности после интеграции.

б) Временное скрывание новой функциональности для тестирования в рабочей среде.

в) Ручное включение новой функциональности пользователем.

г) Все вышеперечисленное.

Ответ: б) Временное скрывание новой функциональности для тестирования в рабочей среде.

Ключ к тесту:

1	2	3	4	5	6	7	8	9	10	11
в)	б)	в)	а)	б)	б)	а)	б)	б)	б)	б)
12	13	14	15	16	17	18	19	20	21	22
г)	в)	в)	б)	б)	г)	в)	в)	б)	г)	в)
23	24	25	26	27	28	29	30	31	32	33
б)	г)	б)	а)	г)	в)	г)	г)	б)	б)	в)
34	35	36	37	38	39	40	41	42	43	44
б)	г)	в)	б)	в)	в)	г)	в)	а)	г)	а)
45	46	47	48	49	50	51	52	53	54	55

г)	г)	г)	г)	г)	а)	г)	г)	г)	г)	б)
----	----	----	----	----	----	----	----	----	----	----

**4.2.1.1 Комплект тестовых заданий для проведения экзамена по
МДК. 02.01 Разработка, поддержка и тестирование программных модулей**

Номер задания	Содержание вопроса	Правильный ответ	Компетенция
1.	Что понимается под термином «тестирование программного обеспечения»? А) Процесс написания кода программы Б) Процесс проверки соответствия программы требованиям путем наблюдения за ее работой в искусственно созданных ситуациях В) Процесс установки программы на компьютер пользователя Г) Процесс составления технической документации	Б	ПК 2.1
2.	Как называется метод тестирования, при котором тестировщик имеет доступ к исходному коду программы и строит тесты, анализируя логику его работы? А) Метод «черного ящика» Б) Метод «серого ящика» В) Метод «белого ящика» (стеклянного ящика) Г) Метод «дымового тестирования»	В	ПК 2.2
3.	На каком уровне тестирования проверяется взаимодействие между несколькими модулями или компонентами приложения? А) Модульное тестирование Б) Интеграционное тестирование В) Системное тестирование Г) Приемочное тестирование	Б	ПК 2.4
4.	Что такое «баг» (bug) или дефект в терминологии тестирования? А) Синтаксическая ошибка в коде, найденная компилятором Б) Любое несоответствие фактического результата работы программы ожидаемому результату (согласно требованиям или здравому смыслу) В) Зависание компьютера при работе программы Г) Отсутствие комментариев в коде	Б	ПК 2.3
5.	Какой вид тестирования проверяет, что новое изменение в коде (например, исправление ошибки) не нарушило уже существующую, ранее работавшую функциональность? А) Дымовое тестирование Б) Нагрузочное тестирование В) Регрессионное тестирование Г) Модульное тестирование	В	ПК 2.1
6.	Что такое «тест-кейс» (test case)? А) Программа для автоматического поиска ошибок	Б	ПК 2.2

	Б) Набор входных данных, условий выполнения и ожидаемых результатов, разработанный для проверки определенного свойства программы В) Список найденных ошибок Г) Этап жизненного цикла разработки ПО		
7.	Для чего используются утверждения (assertions) в модульном тестировании (например, в JUnit или xUnit)? А) Для остановки выполнения теста Б) Для сравнения фактического результата работы метода с ожидаемым значением В) Для генерации тестовых данных Г) Для логирования действий программы	Б	ПК 2.4
8.	Какая характеристика качества ПО означает способность продукта безотказно выполнять заданные функции при заданных условиях в течение заданного периода времени? А) Функциональность Б) Практичность В) Надежность Г) Мобильность	В	ПК 2.3
9.	Что проверяется при «дымовом тестировании» (smoke testing)? А) Работа программы на предельных нагрузках Б) Самая важная, ключевая функциональность, чтобы убедиться, что приложение в принципе запускается и не «дымится» В) Корректность работы отдельного модуля Г) Удобство пользовательского интерфейса	Б	ПК 2.1
10.	Что такое «эмулятор» или «заглушка» (mock-объект) при тестировании? А) Реальный объект базы данных Б) Объект, имитирующий поведение реального компонента для изоляции тестируемого модуля от внешних зависимостей В) Пользователь, тестирующий программу Г) Документ, описывающий архитектуру ПО	Б	ПК 2.2
11.	В чем основное отличие верификации от валидации? А) Верификация отвечает на вопрос «Правильно ли мы сделали продукт?» (соответствие спецификации), а валидация — «Правильный ли продукт мы сделали?» (удовлетворение потребностей пользователя) Б) Это термины-синонимы В) Верификация проводится только заказчиком, а валидация — только разработчиком Г) Верификация — это ручное тестирование, а валидация — автоматизированное	А	ПК 2.5
12.	Что должен содержать правильно составленный баг-репорт (отчет об ошибке)? А) Только название ошибки	Б	ПК 2.3

	Б) Заголовок (Что? Где? Когда?), описание шагов воспроизведения, фактический и ожидаемый результат, окружение В) Имя программиста, написавшего код Г) Исходный код программы		
13.	Какой инструмент используется для автоматизации тестирования веб-интерфейсов (UI) через браузер? А) JUnit / xUnit Б) Selenium WebDriver В) Postman Г) Git	Б	ПК 2.1
14.	Что такое «покрытие кода» (code coverage)? А) Количество строк кода в программе Б) Метрика, показывающая, какой процент исходного кода программы выполняется при запуске тестов В) Процесс написания кода Г) Документирование кода комментариями	Б	ПК 2.5
15.	Какова цель нагрузочного тестирования (performance testing)? А) Найти как можно больше ошибок в логике программы Б) Определить, насколько быстро и стабильно работает система под определенной (ожидаемой) нагрузкой В) Проверить удобство интерфейса для пользователя Г) Проверить корректность установки программы	Б	ПК 2.4

4.2.1.2. Комплект тестовых заданий для проведения экзамена по МДК. 02.02 Осуществление интеграции программных модулей

Номер задания	Содержание вопроса	Правильный ответ	Компетенция
1.	Что является основной целью интеграции программных модулей? А) Написание кода для каждого модуля по отдельности Б) Объединение автономных модулей в единую работающую систему и обеспечение корректности их взаимодействия В) Создание резервных копий программных модулей Г) Документирование кода каждого модуля	Б	ПК 2.1
2.	Как называется подход к интеграции, при котором все модули собираются и тестируются вместе только после завершения разработки всех компонентов? А) Нисходящая интеграция (Top-down) Б) Восходящая интеграция (Bottom-up) В) Монолитная интеграция (Big-Bang) Г) Непрерывная интеграция (Continuous	В	ПК 2.2

	Integration)		
3.	При использовании какой стратегии интеграции для имитации еще не разработанных или отсутствующих вызываемых модулей используются специальные заменяющие компоненты? А) Только при восходящей интеграции Б) Только при нисходящей интеграции В) При нисходящей интеграции используются заглушки (stubs), при восходящей — драйверы (drivers) Г) Ни при одной из стратегий заглушки не используются	В	ПК 2.4
4.	Что проверяет интеграционное тестирование? А) Корректность работы отдельно взятого метода или функции Б) Соответствие программы требованиям заказчика В) Корректность взаимодействия между несколькими модулями или компонентами системы и их интерфейсов Г) Производительность системы под критической нагрузкой	В	ПК 2.3
5.	Какой инструмент чаще всего используется для тестирования взаимодействия программных модулей через REST API? А) Selenium Б) Postman В) Git Г) Microsoft Word	Б	ПК 2.1
6.	Что такое «заглушка» (mock-объект) в контексте интеграции и тестирования? А) Реальный компонент базы данных Б) Объект, имитирующий поведение реального компонента для изоляции тестируемого модуля и проверки взаимодействия В) Документ, описывающий архитектуру системы Г) Ошибка в программном коде	Б	ПК 2.2
7.	Для чего используется система контроля версий (например, Git) в процессе интеграции модулей? А) Для автоматического тестирования кода Б) Для управления изменениями в коде, синхронизации работы команды и слияния (merge) разработанных модулей в общую кодовую базу В) Для компиляции программных модулей Г) Для развертывания приложения на сервере	Б	ПК 2.4
8.	Какой процесс подразумевает частое (несколько раз в день) слияние кода всех разработчиков в основную ветку с автоматическим запуском сборки и тестов?	Б	ПК 2.3

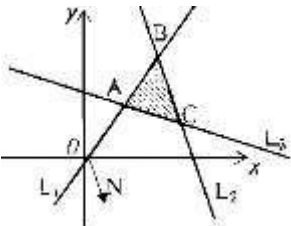
	А) Монолитная разработка Б) Непрерывная интеграция (Continuous Integration / CI) В) Регрессионное тестирование Г) Код-ревью		
9.	Что такое «конфликт слияния» (merge conflict) в системе контроля версий? А) Ситуация, когда два разработчика исправили одну и ту же строку кода по-разному, и система не может автоматически объединить изменения Б) Ошибка компиляции программы В) Спор между разработчиками о структуре проекта Г) Отказ в доступе к репозиторию	А	ПК 2.1
10.	Какой вид тестирования интеграции проверяет, что после добавления нового модуля или изменения существующего, ранее работавшая функциональность не нарушилась? А) Дымовое тестирование Б) Нагрузочное тестирование В) Регрессионное тестирование Г) Модульное тестирование	В	ПК 2.2
11.	Какая проблема наиболее характерна для этапа интеграции модулей и редко проявляется при их изолированном тестировании? А) Синтаксическая ошибка в коде Б) Ошибки в алгоритме внутри модуля В) Несоответствие форматов данных, передаваемых между модулями Г) Неправильное наименование переменных	В	ПК 2.5
12.	Что такое API (Application Programming Interface) в контексте интеграции? А) Готовая программа для тестирования Б) Набор правил и протоколов, с помощью которого один программный модуль может взаимодействовать с другим В) Тип базы данных Г) Язык программирования	Б	ПК 2.3
13.	Какова цель использования драйвера (driver) при восходящей интеграции (Bottom-up)? А) Заменить еще не готовый низкоуровневый модуль Б) Имитировать работу вышестоящего модуля, чтобы вызывать и тестировать нижестоящие В) Увеличить скорость выполнения программы Г) Создать графический интерфейс пользователя	Б	ПК 2.1
14.	Что должна содержать спецификация интерфейса программного модуля для успешной интеграции? А) Только название модуля Б) Исходный код модуля	В	ПК 2.5

	В) Описание передаваемых параметров, их типов, формата данных и возможных исключительных ситуаций Г) График работы разработчика		
15.	Что такое «оркестрация сервисов» (service orchestration) в сложных интеграционных проектах? А) Способ написания музыки для приложения Б) Централизованный подход к управлению взаимодействием множества микросервисов, где координатор (оркестратор) управляет последовательностью вызовов В) Остановка всех сервисов для обслуживания Г) Ручное тестирование каждого сервиса	Б	ПК 2.4

**4.2.1.3. Комплект тестовых заданий для проведения экзамена по
МДК. 02.03 Математическое моделирование**

Номер задания	Содержание вопроса	Правильный ответ	Компетенция
1.	Моделирование — это процесс замены реального ... моделью, отражающей его существенные признаки с точки зрения достижения конкретной цели (вставьте пропущенные слова)	объекта (процесса, явления)	ПК 2.1.
2.	Объект, который используется в качестве «заместителя», представителя другого объекта с определенной целью, называется ...	моделью	ПК 2.1.
3.	Совокупность записанных на языке математики формул, отражающих те или иные свойства объекта-оригинала или его поведение называется ...	математическая модель объекта	ПК 2.1.
4.	Способ представления описываемых объектов и процессов в виде изображений это ... информационной моделью (вставьте пропущенное слово)	графическая	ПК 2.1.
5.	При разделении математических моделей по принципам их построения получают виды (выберите не менее двух правильных ответов): а) аналитические	А,Г	ПК 2.1.
	б) детерминированные в) стохастические г) имитационные		

6.	По виду входной информации математические модели можно разделить на группы (выберите не менее двух правильных ответов): а) статические б) дискретные в) непрерывные г) динамические	Б,В	ПК 2.1.
7.	Примерами образных информационных моделей являются (выберите не менее двух правильных ответов): а) рисунок б) фотография в) словесное описание г) формула	А,Б	ПК 2.1.
8.	Математические модели описывают основные свойства объекта, процесса или системы, его параметры, внутренние и внешние связи посредством...	логико-математических конструкций	ПК 2.1.
9.	В задачах проектирования или исследования поведения реальных объектов, процессов или систем чаще всего используется тип математических моделей...	ДНА	ПК 2.1.
10.	Модели, в которых предполагается отсутствие всяких случайных воздействий и их элементы (элементы модели) достаточно точно установлены, называются...	детерминированные	ПК 2.1.
11.	Установите последовательность решения оптимизационных задач с использованием MathCad: 1. выбор метода оптимизации 2. под Given вводятся ограничения, условие неотрицательности 3. пишется функция Given. 4. вводятся значения переменных для начала счета 5. вставляется функция maximize (minimize) 6. записывается целевая функция	643251	ПК 2.4.
12.	Установите соответствие между типами задач математического программирования и их характерному признаку:	1Б,2А,3В	ПК 2.4.
		Признак	Вид задач

	<table border="1"> <tr> <td>1</td><td>По характеру взаимосвязи между переменными</td><td>а</td><td>Непрерывные и дискретные</td></tr> <tr> <td>2</td><td>По характеру изменения переменных</td><td>б</td><td>Линейные и нелинейные</td></tr> <tr> <td>3</td><td>По учету фактора времени</td><td>в</td><td>Статистические и динамические</td></tr> </table>	1	По характеру взаимосвязи между переменными	а	Непрерывные и дискретные	2	По характеру изменения переменных	б	Линейные и нелинейные	3	По учету фактора времени	в	Статистические и динамические		
1	По характеру взаимосвязи между переменными	а	Непрерывные и дискретные												
2	По характеру изменения переменных	б	Линейные и нелинейные												
3	По учету фактора времени	в	Статистические и динамические												
13.	Направление математики, изучающее методы решения экстремальных задач с линейной структурой это ...	линейное программирование	ПК 2.4.												
14.	Максимум функции Z достигается в точке: 	С	ПК 2.4.												
15.	Установите правильную последовательность основных этапов операционного исследования: 1. постановка задачи; 2. наблюдение явления и сбор исходных данных; 3. расчет модели; 4. построение математической модели 5. тестирование модели и анализ выходных данных; 6. применение результатов исследования.	214356	ПК 2.4.												
16.	Какие методы применяют для решения задач линейного программирования: а) графический б) линейный в) симплексный г) метод потенциалов	А,В,Г	ПК 2.4.												
17.	В какой координатной четверти расположено множество решений системы ограничений задач линейного программирования (ответ напишите словом)	первой	ПК 2.4.												
18.	Если задача линейного программирования имеет хотябы один оптимальный план, она называется ...	разрешимой	ПК 2.4.												

19.	Графический метод решения задач линейного программирования основан на ... интерпретации задачи линейного программирования (вставьте пропущенное слово)	геометрической	ПК 2.4.
20.	Системы ограничений задач линейного программирования, состоящая из неравенств, геометрически определяет...	полуплоскость	ПК 2.5.
21.	Областью допустимых решений системы неравенств может быть (выберите не менее двух правильных ответов): а) выпуклый прямоугольник; б) выпуклая многоугольная неограниченная область; в) выпуклый многоугольник или выпуклая многоугольная неограниченная область; г) только выпуклый прямоугольник; д) пустая область; е) луч; ж) отрезок; з) единственная точка.	А,Б,Д,Е,Ж,З	ПК 2.5.
22.	Для нахождения оптимального плана в решении транспортной задачи используется метод ...	потенциалов	ПК 2.5.
23.	Методами определения начального плана транспортной задачи являются (выберите не менее двух правильных ответов): а) симплекс-метод; б) метод «северо-западного угла»; в) метод минимального элемента; г) метод Крамера.	Б,В	ПК 2.5.
24.	Метод множителей Лагранжа, графический метод, градиентные методы относятся к методам решения задач...	нелинейного программирования	ПК 2.5.
25.	Методом множителей Лагранжа определить условный экстремум функции $Z=1-x^2-y^2$ при условии $x+y=1$ (ответ напишите в виде числа)	1/2	ПК 2.5.
26.	Численные методы интегрирования являются...	приближенными	ПК 2.5.

27.	Дифференциальное уравнение, в котором функция y зависит от нескольких аргументов, называется дифференциальное уравнение в ...	частных производных	ПК 2.5.
28.	Для применения метода простых итераций необходима...	сходимость итерационного процесса	ПК 2.5.
29.	Системы, в которых входные переменные являются функциями от времени или каких-либо других параметров называют...	Динамическими системами	ПК 2.5.
30.	Какой принцип лежит в основе решения всех задач динамического программирования, который гласит: <i>«Каково бы ни было состояние системы S перед очередным шагом, надо выбирать управление на этом шаге так, чтобы выигрыш на данном шаге плюс оптимальный выигрыш на всех последующих шагах был максимальным»</i>	оптимальности	ПК 2.5.

4.2.1.4 Комплект тестовых заданий для проведения экзамена МДК.02.04 Безопасность программного обеспечения

Номер задания	Содержание вопроса	Правильный ответ	Компетенция
16.	Что из перечисленного является основными свойствами информации (триада CIA)? А) Конфиденциальность, целостность, доступность Б) Секретность, полнота, актуальность В) Достоверность, конфиденциальность, целостность Г) Аутентичность, доступность, конфиденциальность	А	ПК 2.5
17.	Какой метод тестирования безопасности выполняется на работающем приложении без доступа к исходному коду? А) SAST Б) DAST В) SCA Г) Code review	Б	ПК 2.5
18.	Какая методология моделирования угроз использует категории Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege? А) PASTA Б) STRIDE В) LINDDUN	Б	ПК 2.5

	Г) OCTAVE		
19.	Что такое SQL-инъекция? А) Внедрение вредоносного JavaScript-кода в веб-страницу Б) Внедрение произвольного SQL-кода в запрос к базе данных В) Межсайтовая подделка запроса Г) Перехват сессии пользователя	Б	ПК 2.5
20.	Какой метод является наиболее эффективным для защиты от SQL-инъекций? А) Экранирование спецсимволов Б) Использование параметризованных запросов (prepared statements) В) Фильтрация входных данных с помощью регулярных выражений Г) Ограничение длины ввода	Б	ПК 2.5
21.	Уязвимость межсайтового скриптинга (XSS) позволяет атакующему: А) Выполнить произвольный код на сервере Б) Выполнить произвольный JavaScript-код в браузере жертвы В) Получить доступ к файловой системе сервера Г) Изменить конфигурацию сети	Б	ПК 2.5
22.	Атака CSRF (межсайтовая подделка запроса) заключается в: А) Выполнении действий от имени аутентифицированного пользователя без его ведома Б) Внедрении SQL-кода В) Перехвате cookie Г) Подмене DNS	А	ПК 2.5
23.	Что такое IDOR (Insecure Direct Object References)? А) Небезопасная прямая ссылка на объект Б) Отсутствие проверки прав доступа при обращении к объектам по идентификатору В) Оба варианта верны Г) Ни один из вариантов не верен	В	ПК 2.5
24.	Какое свойство криптографии означает, что отправитель не может отказаться от факта отправки сообщения? А) Конфиденциальность Б) Целостность В) Аутентификация Г) Неотказуемость	Г	ПК 2.5
25.	Какой алгоритм относится к симметричному шифрованию? А) RSA Б) AES В) DSA Г) ECDSA	Б	ПК 2.5

26.	Для чего используется «соль» (salt) при хешировании паролей? А) Для увеличения длины хеша Б) Для защиты от радужных таблиц и одинаковых хешей для одинаковых паролей В) Для обратимого шифрования пароля Г) Для ускорения хеширования	Б	ПК 2.5
27.	Какой стандарт описывает топ-10 наиболее критичных рисков безопасности веб-приложений? А) ISO 27001 Б) OWASP Top 10 В) PCI DSS Г) ГОСТ Р 56545	Б	ПК 2.5
28.	Что такое принцип наименьших привилегий (least privilege)? А) Все пользователи имеют равные права Б) Каждый субъект должен иметь минимально необходимые права для выполнения своих функций В) Права предоставляются только администраторам Г) Права выдаются на ограниченное время	Б	ПК 2.5
29.	Какой тип анализа безопасности приложений выполняется на исходном коде без его выполнения? А) SAST Б) DAST В) IAST Г) RASP	А	ПК 2.5
30.	Что такое пентест (тестирование на проникновение)? А) Автоматическое сканирование уязвимостей Б) Легализованная имитация атак злоумышленника для оценки защищенности В) Анализ конфигурации сети Г) Установка обновлений безопасности	Б	ПК 2.5

Критерии оценивания:

Оценка «отлично» выставляется студенту, если он правильно ответил на 90-100% от общего числа вопросов тестовых заданий.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 70-89% от общего числа вопросов тестовых заданий.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 50-69% от общего числа вопросов тестовых заданий.

Оценка «неудовлетворительно» выставляется студенту, если он правильно ответил менее чем на 50% от общего числа вопросов тестовых заданий.

4.2.2. Комплект заданий для проведения дифференцированного зачета по учебной и производственной практикам

Дифференцированный зачет по учебной и производственным практикам выставляется на основании выполнения видов работ, предусмотренных программой практик.

Учебная практика

Виды работ	Проверяемые результаты (ПК)
Вводный инструктаж по технике безопасности	ПК 2.1.
Знакомство с предметной области разработки программного обеспечения.	ПК 2.2.
Изучение требования к программному обеспечению.	ПК 2.3.
Анализ функциональных требований.	ПК 2.4.
Построение функциональных диаграмм.	ПК 2.5.
Объектно-ориентированный анализ требований к программному обеспечению.	
Участие в проектировании интерфейса пользователя.	
Участие в разработке кода программного средства.	
Изучение программной документации.	
Участие в разработке и проведении тестов.	
Защита отчета по учебной практике.	

Производственная практика

Виды работ	Проверяемые результаты (ПК)
Вводный инструктаж по технике безопасности.	ПК 2.1.
Изучение предметной области разработки программного обеспечения.	ПК 2.2.
Формирование требования к программному обеспечению.	ПК 2.3.
Анализ функциональных и нефункциональных требований.	ПК 2.4.
Объектно-ориентированный анализ требований к программному обеспечению.	ПК 2.5.
Проектирование интерфейса пользователя.	
Разработка кода программного средства.	
Анализ предметной области индивидуального задания.	
Обследование объекта автоматизации системы.	
Сбор данных для	
Формирование требований пользователя к ИС.	
Определение программных средств разрабатываемой информационной системы.	
Осуществление выбора модели построения информационной модели.	
Защита отчета по производственной практике	

5. Фонд оценочных средств для экзамена по модулю

1. Паспорт

ФОС предназначен для контроля и оценки результатов освоения профессионального модуля ПМ.02 Осуществление интеграции программных модулей по специальности 09.02.07 Информационные системы и программирование.

Профессиональные компетенции	Показатели оценки результата
ПК 2.1.	Проектировать модули программного обеспечения.
ПК 2.2.	Разрабатывать модули программного обеспечения.

ПК 2.3.	Выполнять интеграцию модулей и компонентов программного обеспечения.
ПК 2.4.	Выполнять тестирование и отладку программного обеспечения.
ПК 2.5.	Осуществлять документирование программных модулей программного обеспечения.

2. Задание для экзаменуемого

Инструкция

1. Внимательно прочитайте условия задачи и заданий.
2. Последовательно выполняйте задания, строго следуя их хронологии, указанной в билете.
3. При решении задачи, содержание которой основано на реальных (модельных) условиях профессиональной деятельности, соблюдайте принятую методику решения, пошагово демонстрируя ход ее выполнения, аргументируя полученный результат.
4. При выполнении заданий, основанных на реальных (модельных) условиях профессиональной деятельности, обоснуйте свой ответ.

Задания для проведения экзамена по ПМ. 02 Разработка и интеграция модулей программного обеспечения

Вариант №1

1. Модульное программирование
2. Практическое задание. Найти фамилии работников данного предприятия, чья зарплата за месяц ниже средней по этому предприятию, а также распечатать список проработавших здесь более 10 лет с указанием фамилий, зарплаты, стажа работы и должности.

Вариант №2

1. Структурное проектирование
2. Практическое задание. Результаты переписи населения хранятся в файле. Напечатать фамилии, имена, и определить общее число жителей, родившихся после 1990 года. Составить тестовые таблицы для проверки работы программы.

Вариант №3

1. Модели разработки ПО
2. Спроектировать пользовательский интерфейс, определить цели и исходных требований к программе, провести анализ пользователей и создание сценариев поведения пользователей.
 - Определить предметную область и сферу применения программного продукта. •
 Определить целевую аудиторию.
 - Построить описательную модель пользователя (профиль). Выделить группы пользователей.
 - Сформировать множество сценариев поведения пользователей на основании составленной модели.
 - Выделить функциональные блоки приложения и схему навигации между ними (структуру диалога).

Вариант №4

1. Объектно-ориентированное программирование и его принципы
2. Практическое задание. Дан файл Товар, содержащий сведения об экспортируемых товарах с указанием наименований этих товаров, стран импортеров и объемов поставляемых партий в штуках. Составить список стран, в которые экспортируется заданный товар, и общий объем его экспорта. Составить тестовые таблицы для проверки работы программы.

Вариант №5

1. Возможности отладчика
2. Практическое задание. В файле хранятся сведения о студентах: фамилия, имя, отчество, рост, масса. Вычислить средний рост студентов, а также рост самого высокого и самого низкого студента. Составить тестовые таблицы для проверки работы программы.

Вариант №6

1. Определение тестирования. Особенности тестирования «белого» и «черного» ящика.
2. Практическое задание. Вычислить средний балл успеваемости студентов группы, если известны оценки каждого из них по математике, русскому языку и физике. Определить список студентов, имеющих средний балл выше среднего балла в группе. Составить тестовые таблицы для проверки работы программы.

Вариант №7

1. Структурное тестирование. Способ тестирования базового пути.
2. Практическое задание. Определить подходящий возраст для вступления в брак, используя следующее соображение: возраст девушки должен быть равен половине возраста мужчины плюс 7, а возраст мужчины должен определяться соответственно как удвоенный возраст девушки минус 14. Составить тестовые таблицы для проверки работы программы.

Вариант №8

1. Цикломатическая сложность программы.
2. Разработать программный код для задач: – Конвертер валют. Пользователь вводит сумму, выбирает две валюты и получает эквивалент во второй валюте. – Средняя температура. Пользователь вводит данные о температуре за некоторый период времени. Программа находит среднее значение. – Пользователь указывает цвет и радиус круга. Программа показывает прямоугольник, в котором круг данного размера и цвета движется горизонтально, меняя направление при касании границы.

Вариант № 9

1. Способы тестирования условий. Тестирование ветвей и операторов отношений.
2. Практическое задание. Используя методы защитного программирования разработать программу формирования следующего файла при условии, что номер группы составляет длину от 3 до 5 символов, а оценки находятся в интервале от 2 до 5.

группа	Фамилия студента	Оценка по 1 предмету	Оценка по 2 предмету	Оценка по 3 предмету	Оценка по 4 предмету

Вариант №10

1. Тестирование циклов
2. Практическое задание. Даны три произвольных числа. Определить, можно ли построить треугольник с такими длинами сторон. Составить тестовые таблицы для проверки работы программы.

Вариант №11

1. Тестирование циклов.
2. Практическое задание. Дана последовательность действительных чисел $a_1 \dots a_n$. Поменять местами наибольшее и наименьшее числа. Составить тестовые таблицы для проверки работы программы.

Вариант №12

1. Методика тестирования программных систем.
2. Практическое задание. По результатам сессии составляется сводная ведомость успеваемости студентов по группе. Группа студентов состоит из 15 человек, каждый студент имеет 5 оценок. Необходимо выбрать данные о студентах, имеющих пятерки по всем предметам. Составить тестовые таблицы для проверки работы программы.

Вариант №13

1. Тестирование элементов.
2. Практическое задание. По результатам сессии составляется сводная ведомость успеваемости студентов по группе. Группа студентов состоит из 15 человек, каждый студент имеет 5 оценок. Вычислить средний балл по группе. Составить тестовые таблицы для проверки работы программы.

Вариант №14

1. Системное тестирование: тестирование восстановления, тестирование безопасности.
2. Практическое задание. Составить тестовые таблицы для проверки работы программы, вычисляющей факториал числа, вводимого с клавиатуры.

Вариант №15

1. Классификация методов проектирования программных продуктов
2. Практическое задание. Составить тестовые таблицы для проверки работы программы, находящей максимальный элемент в матрице $A(3 \times 4)$.

Вариант №16

1. Спиральная модель жизненного цикла разработки программного продукта.
2. Практическое задание. Дано целое N , за которым следует N вещественных чисел. Определить, сколько среди них отрицательных. Составить тестовые таблицы для проверки работы программы.

Вариант №17

1. Модульное программирование. Зависимость критериев качества (функциональность, надежность, легкость применения, эффективность, сопровождаемость, изучаемость, модифицируемость) от примитивов качества программных средств.
2. С помощью конструктора графического интерфейса пользователя IDE NetBeans создать графический интерфейс пользователя для приложения с именем MyApplication, по плану: – создать GUI Container (контейнер GUI), добавить компоненты, выронить 23 их положение, изменить размер, осуществить привязку компонентов, продемонстрировать управление поведением при автоматическом изменении размера и редактировании свойства компонентов.

Вариант № 18

1. Примитивы качества программных средств.
2. Практическое задание. Дана последовательность действительных чисел $a_1, \dots, a_n$. Заменить все ее члены, которые больше заданного числа Z , этим числом. Определить количество выполненных замен. Составить тестовые таблицы для проверки работы программы.

Вариант № 19

1. Организационные процессы жизненного цикла программного продукта: процесс управления, процесс создания инфраструктуры, процесс усовершенствования, процесс обучения,
2. Практическое задание. Дана матрица $A(3 \times 4)$ целых чисел. Вычислить сумму и произведение отрицательных чисел. Составить тестовые таблицы для проверки работы программы.

Вариант № 20

1. Тестирование интеграций. Восходящее тестирование интеграций.

2. Практическое задание. Для проверки данных от клиентов разработайте приложение, которое позволит провести валидацию на корректность данных. Результат проверки необходимо фиксировать в документе ТестКейс.docx. Проверить два любых критерия.

Вариант №21

1. Стандартная техника отладки. Обработчики исключений. Способы тестирования условий. Тестирование ветвей и операторов отношений.
2. Практическое задание. Дан массив действительных чисел с размерностью N. Определить, сколько в нем отрицательных, положительных и нулевых элементов. Составить тестовые таблицы для проверки работы программы.

3. Пакет экзаменатора

3.1. Условия выполнения заданий

1. Ознакомьтесь с заданиями для экзаменующихся.
2. Количество вариантов заданий для экзаменующихся: 21
3. Время на выполнение комплексного задания отводится максимальное время: 60 мин.
4. Экзамен проводится одновременно для всей учебной группы.
5. Каждый экзаменующийся выполняет выбранный им вариант.
6. Оборудование: персональный компьютер с необходимым программным обеспечением.

Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет» колледж СКФУ в г. Ставрополе Утвержден предметно-цикловой комиссией протокол №__ от «__» _____ 20__ г. ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № <table border="1" style="width: 100%;"><tr><td style="width: 40%;">Код и наименование специальности</td><td style="width: 60%;">09.02.11</td></tr><tr><td>Учебная дисциплина (МДК, ПМ)</td><td>Разработка и управление программным обеспечением</td></tr><tr><td></td><td>ПМ.02 Разработка и интеграция модулей программного обеспечения</td></tr></table>		Код и наименование специальности	09.02.11	Учебная дисциплина (МДК, ПМ)	Разработка и управление программным обеспечением		ПМ.02 Разработка и интеграция модулей программного обеспечения
Код и наименование специальности	09.02.11						
Учебная дисциплина (МДК, ПМ)	Разработка и управление программным обеспечением						
	ПМ.02 Разработка и интеграция модулей программного обеспечения						
Теоретическая часть: Примитивы качества программных средств. Практическая часть: Дана последовательность действительных чисел $a_1 \dots a_n$. Заменить все ее члены, которые больше заданного числа Z, этим числом. Определить количество выполненных замен. Составить тестовые таблицы для проверки работы программы. Преподаватель_____							

Оценочные средства для проверки уровня сформированности компетенций

Номер задания	Содержание вопроса	Правильный ответ	Компетенция
1.	На каком этапе жизненного цикла разработки ПО создаётся архитектура приложения и определяются основные модули? А) Этап кодирования Б) Этап проектирования В) Этап тестирования Г) Этап эксплуатации	Б	ПК 2.1
2.	Какой метод тестирования предполагает, что тестировщик не знает внутреннего устройства модуля и проверяет его только по внешнему поведению согласно спецификации? А) Метод «белого ящика» Б) Метод «чёрного ящика» В) Метод «серого ящика» Г) Регрессионное тестирование	Б	ПК 2.2
3.	Для чего в процессе интеграции модулей используются «заглушки» (stubs)? А) Для замены ещё не разработанных или отсутствующих вызываемых модулей при нисходящей интеграции Б) Для увеличения производительности системы В) Для автоматической генерации документации Г) Для защиты от несанкционированного доступа	А	ПК 2.4
4.	Какой инструмент чаще всего используется для автоматизации модульного тестирования в Java-проектах? А) Postman Б) Selenium В) JUnit Г) Git	В	ПК 2.3
5.	Что такое «регрессионное тестирование»? А) Тестирование только новой функциональности Б) Повторное тестирование уже проверенной функциональности после внесения изменений для обнаружения ошибок в ранее работавшем коде В) Тестирование производительности под высокой нагрузкой Г) Тестирование удобства пользовательского интерфейса	Б	ПК 2.1
6.	Какую команду Git необходимо выполнить, чтобы загрузить изменения из удалённого репозитория и сразу объединить их с локальной веткой? А) git commit Б) git push	В	ПК 2.2

	В) git pull Г) git branch		
7.	Что из перечисленного является примером инструмента для тестирования REST API? А) Visual Studio Б) Postman В) GitKraken Г) SonarQube	Б	ПК 2.4
8.	Какая стратегия интеграции предполагает объединение всех модулей одновременно только после завершения их разработки («всё и сразу»)? А) Нисходящая (top-down) Б) Восходящая (bottom-up) В) Монолитная (big-bang) Г) Непрерывная (continuous)	В	ПК 2.3
9.	Что такое «покрытие кода» (code coverage) в тестировании? А) Процент времени, затраченного на тестирование Б) Метрика, показывающая, какая доля исходного кода программы выполняется при прогоне тестов В) Количество тестировщиков в проекте Г) Число найденных ошибок на тысячу строк кода	Б	ПК 2.1
10.	Для чего используется механизм обработки исключений (try-catch) в программах? А) Для оптимизации скорости выполнения кода Б) Для корректной реакции на ошибки времени выполнения и предотвращения аварийного завершения программы В) Для документирования кода Г) Для управления версиями	Б	ПК 2.2
11.	Какой вид тестирования проверяет корректность взаимодействия между несколькими модулями системы? А) Модульное тестирование Б) Интеграционное тестирование В) Дымовое тестирование Г) Приёмочное тестирование	Б	ПК 2.5
12.	Что такое «непрерывная интеграция» (Continuous Integration, CI)? А) Процесс постоянного написания кода без перерыва Б) Практика частого слияния кода всех разработчиков в основную ветку с автоматической сборкой и запуском тестов В) Интеграция модулей только в конце проекта Г) Использование одного языка программирования для всех модулей	Б	ПК 2.3

13.	Что должно быть указано в качественном отчёте об ошибке (баг-репорте)? А) Только название модуля, в котором найдена ошибка Б) Заголовок, шаги воспроизведения, фактический и ожидаемый результат, среда (окружение) В) Предполагаемый разработчик, допустивший ошибку Г) Исходный код всего приложения	Б	ПК 2.1
14.	Какая характеристика программного обеспечения проверяется при нагрузочном тестировании? А) Удобство интерфейса Б) Производительность и стабильность работы под ожидаемой нагрузкой В) Соответствие кода стандартам оформления Г) Полнота пользовательской документации	Б	ПК 2.5
15.	Как разрешается «конфликт слияния» (merge conflict) в Git? А) Git автоматически выбирает один из вариантов Б) Разработчик вручную редактирует файл, выбирая нужные изменения и убирая маркеры конфликта, после чего выполняет коммит В) Удаляется последний коммит Г) Создаётся новая ветка	Б	ПК 2.4

3.2. Критерии оценивания:

Оценки **"отлично"** заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой. Как правило, оценка "отлично" выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.

Оценки **"хорошо"** заслуживает студент, обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе. Как правило, оценка "хорошо" выставляется студентам, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.

Оценки **"удовлетворительно"** заслуживает студент, обнаруживший знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справляющийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой. Как правило, оценка "удовлетворительно" выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

Оценка **"неудовлетворительно"** выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в

выполнении предусмотренных программой заданий. Как правило, оценка "неудовлетворительно" ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании учебы без дополнительных занятий по соответствующей дисциплине.