

**Государственное образовательное учреждение
"Приднестровский государственный университет им. Т.Г. Шевченко"**

Инженерно-технический институт

**Кафедра программного обеспечения вычислительной техники
и автоматизированных систем**

УТВЕРЖДАЮ

Заведующий кафедрой ПОВТ и АС

 С.Г. Федорченко

«30» августа 2021 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

**по дисциплине
ПРОМЫШЛЕННОЕ ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Направление подготовки
2.09.04.04 Программная инженерия

Профиль подготовки
Разработка программно-информационных систем

Квалификация (степень)
выпускника:

магистр

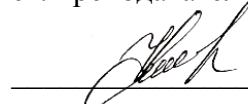
Форма обучения:

очная, заочная

Год набора:

2020 г.

Разработал:
ст. преподаватель кафедры ПОВТ и АС,

 /О.С. Белоконов/

«30» августа 2021 г.

Тирасполь, 2021

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Промышленное тестирование программного обеспечения» у обучающегося должны быть сформированы следующие компетенции:

Категория (группа) компетенций	Код и наименование	Код и наименование индикатора достижения универсальной компетенции
<i>Универсальные компетенции и индикаторы их достижения</i>		
Самоорганизация и саморазвитие (в том числе здоровьесбережение)	УК-6. Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	ИД-1 _{УК-6} Знать методики самооценки, самоконтроля и саморазвития с использованием подходов здоровьесбережения ИД-2 _{УК-6} Уметь решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности; применять методики самооценки и самоконтроля; применять методики, позволяющие улучшить и сохранить здоровье в процессе жизнедеятельности ИД-3 _{УК-6} Владеть технологиями и навыками управления своей познавательной деятельностью и ее совершенствования на основе самооценки, самоконтроля и принципов самообразования в течение всей жизни, в том числе с использованием здоровьесберегающих подходов и методик
<i>Профессиональные компетенции и индикаторы их достижения</i>		
Использование и разработка методов формализации и системный анализ, моделирование прикладных и информационных процессов и управление аналитическими алгоритмизации информационных процессов; анализ и обобщение результатов научно-исследовательской работы с использованием современных достижений науки и техники; исследование перспективных направлений прикладной информатики; анализ и развитие методов управления	ПК-6. Понимание существующие подходы к верификации моделей программного обеспечения.	ИД-1 _{ПК-6} Знает методы верификации моделей программного обеспечения
		ИД-2 _{ПК-6} Умеет использовать методы верификации моделей программного обеспечения

информационными ресурсами; работами в области создания информационных систем; исследование и разработка эффективных методов создания и управления информационными системами в прикладных областях; управление сервисами и информационными ресурсами в информационных системах		
Использование и разработка методов формализации и системный анализ, моделирование прикладных и информационных процессов; анализ и обобщение результатов научно-исследовательской работы с использованием современных достижений науки и техники; исследование перспективных направлений ПО; анализ и развитие методов управления информационными ресурсами; работами в области создания информационных систем	ПК-11. Владение навыками организации промышленного тестирования создаваемого программного обеспечения	ИД-1 _{ПК-11} Знает методы организации промышленного тестирования создаваемого программного обеспечения
		ИД-2 _{ПК-11} Умеет использовать методы организации промышленного тестирования создаваемого программного обеспечения

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины их название	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
РУБЕЖНЫЙ КОНТРОЛЬ	Раздел 1 Раздел 2	УК-6, ПК-6, ПК-11	Презентация №1 Презентация №2 Кейс-задача №1
РУБЕЖНАЯ АТТЕСТАЦИЯ	Раздел 2 Раздел 3		Тест №1 Кейс-задача №2 Реферат №1 Итоговый тест
Промежуточная аттестация		Код контролируемой компетенции (или ее части)	Наименование оценочного средства
№1		УК-6, ПК-6, ПК-11	Экзамен

3. Показатели и критерии оценивания компетенции по этапам формирования, описание шкал оценивания

Этапы оценивания компетенции	Показатели достижения заданного уровня освоения компетенции	Критерии оценивания результатов обучения			
		2	3	4	5
Первый этап	ИД-1 _{УК-6} Знать методики самооценки, самоконтроля и саморазвития с использованием подходов здоровьесбережения	Не знает	Знает основные понятия, но не знает особенности методик самооценки, самоконтроля и саморазвития	Знает основные понятия и основы, но не может применять знания в полной мере в реальных ситуациях	Знает основные понятия. Умеет применять методики самооценки, самоконтроля и саморазвития с использованием подходов здоровьесбережения
Второй этап	ИД-2 _{УК-6} Уметь решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности; применять методики самооценки и самоконтроля; применять	Не умеет	Правильно определяет задачи собственного личностного и профессионального развития, но не умеет определять и реализовывать приоритеты совершенствования собственной деятельности; применять методики самооценки и самоконтроля	Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности; применять методики самооценки и самоконтроля, но не умеет	Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности; применять методики самооценки и самоконтроля; применять методики,

Этапы оценивания компетенции	Показатели достижения заданного уровня освоения компетенции	Критерии оценивания результатов обучения			
		2	3	4	5
	методики, позволяющие улучшить и сохранить здоровье в процессе жизнедеятельности			выбирать оптимальные решения	позволяющие улучшить и сохранить здоровье в процессе жизнедеятельност и
Третий этап	ИД-З _{УК-6} Владеть технологиями и навыками управления своей познавательной деятельностью и ее совершенствования на основе самооценки, самоконтроля и принципов самообразования в течение всей жизни, в том числе с использованием здоровьесберегающ их подходов и методик	Не владеет	Владеет технологиями и навыками управления своей познавательной деятельностью, но не владеет технологиями совершенствова ния.	Владеет технологиями и навыками управления своей познавательной деятельностью и ее совершенствов ания на основе самооценки, самоконтроля и принципов самообразован ия	Владеет технологиями и навыками управления своей познавательной деятельностью и ее совершенствован ия на основе самооценки, самоконтроля и принципов самообразования в течение всей жизни, в том числе с использованием здоровьесберегаю щих подходов и методик
Первый этап	ИД-1 _{ПК-6} Знает методы верификации моделей программного обеспечения	Не знает	Знает основные понятия, но не знает способы использования в профессиональн ой деятельности	Знает методы верификации моделей ПО, но не может применять знания в полной мере в профессиональ ной деятельности	Знает методы верификации моделей ПО и может использовать в профессионально й деятельности
Второй этап	ИД-2 _{ПК-6} Умеет использовать методы верификации моделей программного обеспечения	Не умеет	Правильно определяет модель ПО, но не умеет применять метод ы верификации ПО	Умеет использовать методы верификации моделей ПО, но не в полной мере	Умеет правильно определить модель ПО и использовать необходимые методы верификации ПО
Первый этап	ИД-1 _{ПК-11} Знает методы организации промышленного тестирования создаваемого программного обеспечения;	Не знает	Знает методы организации промышленного тестирования создаваемого ПО, но не знает способы применения этих	Знает методы организации промышленног о тестирования создаваемого ПО, но делает ошибки, не влияющие на	Знает методы организации промышленного тестирования создаваемого ПО. Умеет адекватно применять эти методы

Этапы оценивания компетенции	Показатели достижения заданного уровня освоения компетенции	Критерии оценивания результатов обучения			
		2	3	4	5
			методов	результаты	
Второй этап	ИД-2 _{ПК-11} Умеет использовать методы организации промышленного тестирования создаваемого программного обеспечения	Не умеет	Правильно определяет методы организации промышленного тестирования создаваемого программного обеспечения, но не умеет применять эти методы для решения профессиональн ых задач	Умеет использовать ме тоды организации промышленног о тестирования создаваемого программного обеспечения, но не умеет обрабатывать результаты	Умеет правильно определять использовать методы организации промышленного тестирования создаваемого программного обеспечения для решения профессиональн ых задач

4. Шкала оценивания

Согласно Положению «О порядке организации аттестации в ИТИ ПГУ им. Т.Г. Шевченко, итоговая оценка представляет собой сумму баллов, полученных студентом по итогу освоения дисциплины (модуля):

Оценка в традиционной шкале	Оценка в 100-балльной шкале	Буквенные эквиваленты оценок в шкале 3Е (% успешно аттестованных)
5 (отлично)	88–100	А (отлично) – 88-100 баллов
4 (хорошо)	70–87	В (очень хорошо) – 80-87 баллов
		С (хорошо) – 70-79 баллов
3 (удовлетворительно)	50–69	Д (удовлетворительно) – 60-69 баллов
		Е (посредственно) – 50-59 баллов
2 (неудовлетворительно)	0–49	Фх – неудовлетворительно, с возможной пересдачей – 21-49 баллов
		Ф – неудовлетворительно, с повторным изучением дисциплины – 0-20 баллов

Расшифровка уровня знаний, соответствующего полученным баллам, дается в таблице, указанной ниже

А	“Отлично” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.
В	“Очень хорошо” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному.
С	“Хорошо” - теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения

	ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.
D	“Удовлетворительно” - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.
E	“Посредственно” - теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному.
FX	“Условно неудовлетворительно” - теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий.
F	“Безусловно неудовлетворительно” - теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий.

5. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций при изучении учебной дисциплины в процессе освоения образовательной программы

5.1 Примерный перечень тематик презентации №1

1. Жизненный цикл продукта.
2. Методологии разработки программного обеспечения.
3. Жизненный цикл тестирования.
4. Классификация тестирования.
5. Модели программных систем.
6. Требования. Анализ требований. Тестирование документации. Виды и направления тестирования.
7. Отчеты о дефектах. Жизненный цикл “бага”. Багтрекинг-системы.

5.2. Примерный перечень тематик презентации №2

1. Тестовая документация. Чек-листы. Тест-кейсы.
2. Программное обеспечение для управления тест-кейсами.
3. Техники тест-дизайна. Планирование. Отчетность.
4. Метод определения точек тестирования, основанный на анализе цикломатической сложности Мак-Кейба.
5. Архитектура клиент-сервер. Тестирование web.
6. Тестирование UI и верстки. Программное обеспечение, применяемое при тестировании UI.
7. Интеграционное тестирование. Тестирование API.
8. Сравнительный анализ инструментов тестирования ПО (генераторы тестов, схемы выполнения тестов, оценка тестов, управление тестами).
9. Программное обеспечение, применяемое при тестировании API.
10. Тестирование мобильных приложений.
11. Программное обеспечение, применяемое при тестировании мобильных приложений.

5.3 Типовой вариант Кейс-задача №1. Тема: Контроль и обеспечение качества. Анализ требований

Цель кейс-задачи разработка набора модульных тестов, покрывающих функциональность, разработанного ПО.

Типовой *вариант* задания

1. Автоматическая генерация тестов.
2. Наполнение тестов содержимым.
3. Запуск тестов и просмотр результатов.
4. Изменение конфигурации работы тестов.

Ход работы

1. Автоматическая генерация тестов. Для ускорения разработки тестов команды могут воспользоваться возможностью VisualStudio по автоматической генерации тестов.

После создания тестового проекта, будет предложен выбор из тех типов и методов, для тестирования которых необходимо создать заглушки.

2. Наполнение тестов содержимым. После генерации тестового покрытия команда получить набор скелетов тестов для всех методов, которые были выбраны для тестирования. На следующем шаге команда должна заполнить эти тесты необходимым содержимым, используя функциональность по валидации (Assert), предоставляемую тестовой платформой.

3. Запуск тестов. Команды должны добиться того, чтобы все разработанные тесты проходили успешно.

4. Изменение конфигурации тестов. Для того чтобы проанализировать качество разработанных тестов, команда должна вычислить тестовое покрытие.

5. Отчет о проделанной работе.

5.4 Типовой вариант теста №1.

В. Что такое динамическое тестирование?

О. Это тестирование за счет выполнения кода или программы с различными входными значениями и подтверждением результатов.

В. Что такое GUI-тестирование (GUI Testing)?

О. Тестирование GUI (графического интерфейса пользователя): интерфейс программного обеспечения проверяется на предмет соответствия требованиям.

В. Что такое формальное тестирование?

О. Верификация программного обеспечения, согласно тест-плану, тестовым процедурам и соответствующей документации, с учетом пожеланий клиента.

В. Что такое тестирование на основе рисков?

О. Определяются наиболее важные части системы, затем устанавливается порядок их тестирования, затем следует, собственно, тестирование.

В. Что такое раннее тестирование?

О. Тестирование по возможности проводится как можно раньше, чтобы выявить дефекты на ранних этапах SDLC. Это позволяет быстрее обнаружить и устранить дефекты, экономит расходы.

В. Что такое исчерпывающее тестирование?

О. Тестирование функциональности, с использованием неверных и верных данных ввода и входных условий.

В. Что такое скопление дефектов?

О. Даже небольшой модуль или функциональность могут содержать в себе ряд дефектов, поэтому необходимо больше уделять внимания тестированию функциональности.

В. Что такое «парадокс пестицида»?

О. Если с помощью имеющихся тестовых сценариев не получается обнаружить дефекты, возможно, стоит дополнить/пересмотреть тест-кейсы, чтобы можно было находить больше дефектов.

В. Что такое статическое тестирование?

О. Верификация кода вручную без программы. В этом процессе проблемы находятся в коде, во время его проверки и сравнения с требованиями.

В. Что такое позитивное тестирование?

О. Тестирование, которое проводится в приложении с целью определить, насколько система функциональна. Такой подход больше известен как «тест на прохождение».

В. Что такое негативное тестирование?

О. Тестирование негативных сценариев в ПО: высвечивает ли система ошибку, когда она должна это делать, или не должна.

В. Что такое сквозное тестирование (end-to-end)?

О. Тестирование общей функциональности системы, включая интеграцию данных в модулях.

В. Что такое исследовательское тестирование?

О. Это исследование приложения, чтобы составить представление о его функциональности, добавление (или) изменение существующих тест-кейсов для более качественного тестирования.

В. Что такое «обезьянье тестирование» (MonkeyTesting)?

О. Тестирование приложения без какого-либо плана, тестирование выборочных мест, чтобы обнаружить какие-то сложные системные сбои, а затем и дефекты, которые к этому привели.

В. Что такое нефункциональное тестирование?

О. Валидация различных нефункциональных аспектов системы, таких как пользовательские интерфейсы, совместимость, производительность и прочее.

В. Что такое юзабилити-тестирование?

О. Проверка на предмет того, насколько легко конечные пользователи способны понять и управлять приложением.

В. Тестирование безопасности.

О. Проверяется, насколько хорошо реализованы в приложении все условия безопасности.

В. Что такое тестирование производительности?

О. Анализ эффективности различных характеристик системы — времени ответа, общей производительности с целью установить, как быстро система работает под нагрузкой.

В. Что такое нагрузочное тестирование?

О. Анализ функциональности и производительности приложения в разных условиях.

В. Что такое **стресс-тестирование**?

О. Проверка устойчивости системы в условиях превышения пределов обычного функционирования. Или снижение ресурсов системы и сохранение нагрузки на определенном уровне, чтобы проверить, как приложения при этом себя ведет.

В. Что такое процесс?

О. Процесс — это набор практик для достижения определенной цели; может включать инструменты, методы, материалы и людей.

В. Что такое конфигурационное управление?

О. Процесс поиска, организации и контроля изменений в разработке ПО. Или методология контроля и управления проектом разработки ПО.

В. Что такое процесс тестирования/жизненный цикл?

О. Составление:

- Тест-плана
 - Тест-сценариев
 - Тест-кейсов
 - Выполнение тест-кейсов
 - Проверка результатов
 - Составление отчетов о дефектах
 - Дефект-трекинг
 - Заккрытие дефектов
 - Тестовый релиз
-

В. Как расшифровывается CMMI?

О. CapabilityMaturityModelIntegration (Модель зрелости процессов разработки).

В. Что такое разбор программы?

О. Неформальный анализ исходного кода программы с целью выявить дефекты и верифицировать техники программирования.

В. Что такое модульное тестирование?

О. Тестирование отдельных программ, модулей или элементов кода.

В. Что такое тестирование уровня интеграции?

О. Тестирование соответствующих программ, модулей (или) единиц кода.

В. Что такое тестирование на уровне системы?

О. Тестирование всей компьютерной системы по всем модулям. Такая разновидность тестирования может включать функциональное и структурное тестирование.

В. Что такое альфа-тестирование?

О. Тестирование всей компьютерной системы перед этапом пользовательского тестирования (UAT).

В. Что такое UAT?

О. Тестирование компьютерной системы клиентом, чтобы проверить, соответствует ли система требованиям.

В. Что такое тестовый план?

О. Документ, описывающий масштаб, подход, ресурсы и график тестирования, в котором определены тестовые элементы, отдельные части функционала, тестовые задания, специалисты, которые будут проводить конкретные тесты, и любые риски, требующие дополнительного планирования.

В. Что такое сценарий тестирования?

О. Идентификация всех возможных зон тестирования.

В. Что такое ECP (Equivalence Class Partition)?

О. Метод генерации тест-кейсов.

В. Что такое дефект?

О. Любое несовершенство в работе софта. Или когда ожидаемый результат не соответствует фактической работе приложения.

В. Что такое критичность?

О. Определяет уровень дефекта с функциональной точки зрения, т.е. насколько критичен дефект для приложения.

В. Что такое приоритет?

О. Указывает на срочность устранения дефекта.

В. Что такое повторное тестирование?

О. Повторное тестирование приложения с целью узнать, устранены ли дефекты.

В. Что такое регрессионное тестирование?

О. Верификация существующих функциональных и нефункциональных зон после того, как были изменены отдельные части приложения или добавлены новые функциональные возможности.

В. Что такое тестирование восстановления?

О. Проверяется возможность системы справиться с некоторыми неожиданными ситуациями.

В. Что такое тестирование глобализации (Globalization Testing)?

О. Тестируется возможность запуска приложения независимо от его географической и культурной среды. Проверяется возможность смены языка, даты, формата и валюты, если приложение разработано для пользователей из нескольких стран.

В. Что такое тестирование локализации?

О. Проверка на предмет того, подходит ли приложение для отдельной локальной группы пользователей, культурных и географических условий.

В. Что такое тестирование установки?

О. Проверяется возможность успешной установки ПО, в соответствии с документацией по установке.

В. Что такое тестирование удаления?

О. Проверка возможности удаления ПО.

В. Что такое тестирование на совместимость?

О. Проверяется совместимость приложения с другим программным и аппаратным обеспечением.

В. Что такое стратегия тестирования?

О. Это часть тест-плана, описывающая, как проводится тестирование и какие разновидности тестирования необходимо сделать.

В. Что такое тест-кейс?

О. Тест-кейс — набор определенных шагов, по которым проверяется функциональность системы.

В. Что такое тест-кейс для валидации бизнес-процессов?

О. Этот тест-кейс составляется для того, что проверить определенное условие или требование.

В. Как определяется хороший тест?

О. Тест-кейс, у которого высокий приоритет обнаружения дефектов.

В. Что такое тестирование по сценарию использования?

О. Такое тестирование определяет, было ли ПО разработано согласно случаю использования.

В. Что такое возраст дефекта?

О. Время между датой обнаружения и датой закрытия дефекта.

В. Что такое дефект Showstopper?

О. Дефект, который вынуждает остановить ход тестирования.

В. Что такое завершение тестирования?

О. Это последний этап STLC. Руководство составляет отчеты по тестам, разъясняет статистику проекта, исходя из имеющихся данных.

В. Что такое Bucket Testing?

О. Bucket Testing, или А/В-тестирование. Чаще всего исследуется эффект разного дизайна, используется метрика для веб-сайтов. Две версии сайта запускаются на одной или нескольких веб-страницах, чтобы определить разницу в кликах.

В. Что такое критерии запуска и завершения тестирования?

О. Критерии запуска — процесс, который должен быть представлен в начале системы. Это может быть:

- SRS – ПО
- FRS
- Случай использования
- Тест-кейс
- План тестирования

Критерий завершения определяет готовность приложения к релизу. Это может быть:

- Отчет по тестированию
 - Метрики
 - Отчет по анализу теста
-

В. Что такое тестирование валюты?

О. Это комплексное пользовательское тестирование одновременного доступа к приложению, для верификации влияния на код, модуль или базу данных. Главным образом обнаруживает тупиковые ситуации в коде.

В. Что такое тестирование веб-приложения?

О. Тестирование веб-приложения проводится на веб-сайте для проверки загрузки, производительности, безопасности, функциональности, интерфейса, совместимости и других вопросов, относящихся к юзабилити.

В. Что такое функциональное тестирование?

О. Тестирование элементов (или побочное тестирование) позволяет проверить отдельные работу модулей исходного кода.

В. Что такое тестирование интерфейса?

О. Тестирование интерфейса проверяет взаимодействие отдельных модулей. Чаще всего используется для тестирования пользовательского интерфейса приложений с GUI.

В. Что такое гамма-тестирование?

О. Гамма-тестирование проводится когда ПО уже готово к релизу, проверяется соответствие требованиям.

5.5 Типовой вариант Кейс-задача №2.

Тема: Модульное тестирование разрабатываемого программного обеспечения.

Цель заключается в проведении модульного тестирования разработанной программы. Модульное тестирование – это процесс поиска ошибок в программных модулях. Этим видом тестирования занимается автор кода (в отличие от функционального тестирования, которым занимается тестировщик).

Ход работы:

1) Постановка задачи. Провести обзор разработанного программного кода. В случае обнаружения ошибок составить отчет следующего содержания (табл. 2.1).

Ошибки обзора

Номер ошибки	Название модуля/функции	Описание ошибки	Важность ошибки (высокая, средняя, низкая)	Ошибка исправлена Да/Нет
1	Func1()	В первом цикле for не верно указано значение окончания цикла	средняя	Да

2) Для каждого модуля построить графы и вычислить цикломатические числа. Разработать тестовые случаи для каждого графа и представить их в виде табл. 2.2.

Модуль Func1()

G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
G=2	1	a-b-d-f	Field=4, x=-5	Нет
	2	a-b-e-f	Field=-3, x=-5	Да

3) Провести модульное тестирование согласно составленным тестовым случаям. При необходимости разработать заглушки и привести их в табл. 2.3

Описание заглушек

Номер заглушки	Название модуля	Назначение заглушки	Текст заглушки	Тест пройден Да/Нет

4) В случае не прохождения тестового случая составить отчет о каждой найденной ошибке в виде табл. 2.4.

Модульное тестирование

Номер ошибки	Название модуля	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Описание ошибки	Важность ошибки	Ошибка исправлена Да/Нет
1	Func1()	a-b-d-f	Field=4, x=-5	Появляется не обработанное сообщение windows	средняя	да

5) Построить схему взаимодействия модулей.

6) Разработать стратегию тестирования взаимодействия модулей и привести ее в виде табл. 2.5.

Взаимодействие модулей

Номер в последовательности	Описание последовательности	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
1	Func1()->func2()	Argument=0	

7) Отчет о проделанной работе

5.6 Примерный перечень тематик Реферата (максимальный уровень заимствований в системе Антиплагиат 40%).

1. Регрессионное тестирование информационной системы оценки физиологических возможностей человека.
2. Интеграционное тестирование программной платформы «Контроль знаний и умений»
3. Комплексное тестирование Web-ресурсов.
4. Модульное тестирование систем мониторинга социальных медиа.
5. Методы промышленного тестирования ПО.

5.7 Типовой тест промежуточной аттестации по дисциплине **Промышленное тестирование программного обеспечения**

1. Какие существуют способы получения эталонных значений теста?
 - (1) предсказание ожидаемого результата
 - (2) независимое вычисление результата
 - (3) подстановка в тест результата вычисления тестируемой программы
2. Отметьте верные утверждения:
 - (1) нереализуемый путь недоступен при корректном исполнении программы
 - (2) нереализуемый путь доступен при реализации недопустимых состояний переменных программы
 - (3) нереализуемый путь доступен при сбое
3. Отметьте верные утверждения
 - (1) тестирование – процесс поиска ошибок
 - (2) в фазу тестирования входят поиски и исправление ошибок
 - (3) отладка – процесс локализации и исправления ошибок
4. Зачем нужна спецификация тестирования?
 - (1) для формирования команды тестировщиков
 - (2) для разработки тестового набора
 - (3) для понимания смысла программы
5. Какие существуют методы анализа и локализации ошибки?
 - (1) выполнение программы в уме
 - (2) пошаговое выполнение
 - (3) метод контрольных точек и анализа трасс

6. Какие существуют фазы процесса тестирования?

- (1) разработка тестового набора
- (2) прогон программы на тестовом наборе
- (3) анализ результатов тестирования
- (4) доказательство правильности программы

7. Каковы особенности разработки тестового набора?

- (1) определение областей эквивалентности входных параметров
- (2) анализ покрытия тестами всех возможных случаев поведения
- (3) проверка граничных значений

8. Какие существуют способы получения эталонных значений теста?

- (1) предсказание ожидаемого результата
- (2) независимое вычисление результата
- (3) подстановка в тест результата вычисления тестируемой программы

9. Что такое управляющий граф программы (УГП)?

- (1) множество операторов программы.
- (2) граф, вершины которого кодируют операторы программы, а дуги - управления (порядок исполнения) операторов.
- (3) множество операторов управления

10. Что такое путь в УГП?

- (1) последовательность вершин и дуг УГП с фиксированными начальной и конечной вершиной
- (2) последовательность ветвей УГП с фиксированными начальной вершиной первой ветви и конечной вершиной последней ветви пути
- (3) множество связанных дуг УГП

11. Что такое ветвь УГП?

- (1) последовательность вершин и дуг УГП с фиксированными начальной и конечной вершиной, которые кодируют либо условные операторы, либо первый и последний операторы УГП соответственно
- (2) часть пути, в котором все внутренние вершины кодируют линейные операторы
- (3) начальная и конечная вершина пути

12. Какие предъявляются требования к идеальному критерию тестирования?

- (1) достаточность
- (2) достижимость
- (3) полнота
- (4) проверяемость

13. Какие классы частных критериев тестируемости известны?

- (1) структурные критерии
- (2) функциональные критерии
- (3) стохастические критерии

- (4) мутационный критерий
- (5) сценарные критерии

14. Назовите полный и надежный критерий для нетривиальных классов программ.

- (1) такого критерия не существует
- (2) сценарный критерий
- (3) критерий «черного ящика»

15. Какие существуют разновидности структурных критериев?

- (1) критерий тестирования команд
- (2) критерий тестирования ветвей
- (3) критерий тестирования путей
- (4) критерий тестирования циклов

16. Какая оценка мощности покрытия для следующих пар критериев правильна?

- (1) $C0 \leq C1$
- (2) $C1 \leq C2$
- (3) $C1 < C2$

17. Назовите недостатки структурных критериев.

- (1) не проверяется соответствие со спецификацией
- (2) не проверяется соответствие со спецификацией, не зафиксированное в структуре программы
- (3) не проверяются ошибки в структурах данных

18. Какие существуют разновидности функциональных критериев?

- (1) тестирование пунктов спецификации
- (2) тестирование классов входных данных
- (3) тестирование классов выходных данных
- (4) тестирование правил
- (5) тестирование функций

19. Назовите недостатки функциональных критериев.

- (1) не проверяется соответствие со спецификацией
- (2) не проверяются ошибки, требования к которым не зафиксированы в спецификации
- (3) не проверяются ошибки в структурах данных, требования к которым не зафиксированы в спецификации

20. Назовите критерии стохастического тестирования.

- (1) стохастический метод Хи-квадрат
- (2) стохастический метод Стьюдента
- (3) метод оценки скорости выявления ошибок
- (4) метод особых состояний

21. Какая информация должна собираться при тестировании для применения метода оценки скорости выявления ошибок?

- (1) интервалы между моментами обнаружения ошибок
- (2) оценка плотности ошибок в проблемной области
- (3) данные из исторической базы данных проектов

22. Какой подход используется в методе мутационного тестирования?

- (1) оценка числа ошибок в программе на основе искусственно внесенных мелких ошибок
- (2) создание программ-мутантов с функциональными дефектами
- (3) создание программ-мутантов на основе изменения модульной структуры основной программы

23. Перечислите метрики оценки оттестированности программного проекта?

- (1) сложность тестирования программы по заданному критерию
- (2) остаточная сложность тестирования программы
- (3) оценка степени оттестированности программы по заданному критерию

24. Каковы особенности плоской модели УГП?

- (1) не выделяются структурные компоненты в виде отдельных подграфов УГП
- (2) для тестирования требуется осуществить весь перебор трасс
- (3) оценка оттестированности не зависит от ранее собранных оценок оттестированности УГП компонентов

25. Каковы особенности иерархической модели УГП?

- (1) УГП структурных компонентов выделяются и выносятся из общего УГП проекта
- (2) для тестирования требуется осуществить перебор трасс упрощенного УГП
- (3) оценка оттестированности зависит от ранее собранных оценок оттестированности УГП компонентов

26. Перечислите разновидности функциональных критериев.

- (1) тестирование пунктов спецификации
- (2) тестирование классов входных данных
- (3) тестирование классов выходных данных
- (4) тестирование правил
- (5) тестирование функций

27. Чем отличается оценка оттестированности проекта от оценки для модуля?

- (1) оценка проекта интегрирует оценки оттестированности модулей
- (2) оценка проекта может вычисляться инкрементально
- (3) в результате получаем наихудшую оценку оттестированности
- (4) в результате получаем наилучшую оценку оттестированности

28. Какие существуют разновидности тестирования?

- (1) модульное
- (2) интеграционное
- (3) системное
- (4) регрессионное

29. Какие задачи у модульного тестирования?

- (1) выявление локальных ошибок реализации алгоритмов модулей
- (2) выявление ошибок при вызове модулей
- (3) выявление ошибок взаимодействия модуля с окружением

30. На основе каких принципов строятся тесты для модульного тестирования?

- (1) анализ потоков управления модуля
- (2) анализ потоков данных модуля
- (3) анализ покрытия в соответствии с заданным критерием С

31. Каковы фазы процесса построения тестовых путей?

- (1) построение УГП
- (2) выбор тестовых путей
- (3) генерация тестов, соответствующих выбранным тестовым путям

32. Какие существуют методы построения тестовых путей?

- (1) статические
- (2) динамические
- (3) методы реализуемых путей

33. Как реализуются динамические методы построения тестовых путей?

- (1) наращивание начальных отрезков реализованных путей продолжающими их фрагментами, чтобы увеличить покрытие
- (2) построение пути методом удлинения за счет добавления дуг
- (3) поиск всех реализуемых путей

34. Какие существуют разновидности интеграционного тестирования?

- (1) монолитное тестирование
- (2) нисходящее тестирование
- (3) восходящее тестирование
- (4) Регрессионное тестирование

35. Каковы особенности нисходящего тестирования?

- (1) необходимость разработки заглушек
- (2) необходимость разработки среды управления очередностью вызовов модулей
- (3) параллельная разработка эффективных модулей

36. Каковы особенности восходящего тестирования?

- (1) минимизация разработки заглушек
- (2) запаздывание в проверке функциональности реализуемого приложения
- (3) необходимость разработки среды управления очередностью вызовов модулей

37. В чем заключаются особенности интеграционного тестирования для процедурного программирования?

- (1) тестирование программных комплексов, заданных в виде иерархических структур модулей

- (2) использование диаграмм потока управления в качестве модели тестируемого комплекса
- (3) контроль соответствия спецификациям параметров модулей и межмодульных связей
- (4) контроль наследования

38. Каково выражение для оценки сложности интеграционного тестирования?

- (1) $V(P, C1) = q + k_{in}$
- (2) $V(P, C1) = \sum V(Modi, C1) - k_{in} + k_{ext}$
- (3) $(P, C1) = \sum V(Modi, C1)$

39. Каково выражение для оценки сложности графа вызовов?

- (1) $V'(P, C1') = \sum V'(Modi, C1') - k_{in} + k_{ext}$
- (2) $V'(P, C1') = q + k_{ext}$
- (3) $V'(P, C1') = q$

40. Какие существуют особенности интеграционного тестирования для процедурного программирования?

- (1) тестирование программных комплексов, заданных в виде иерархических структур модулей
- (2) использование диаграмм потока управления в качестве модели тестируемого комплекса
- (3) контроль соответствия спецификациям параметров модулей и межмодульных связей
- (4) контроль наследования

41. Какие существуют особенности модели ГМП в случае объектно-ориентированного программирования (ООП)?

- (1) она становится неприменимой
- (2) она требует адаптации по обработке сообщений
- (3) она требует описания поведения программы
- (4) она требует описания не только структуры, но и поведения программы

42. Какие существуют особенности интеграционного тестирования для ООП?

- (1) тестирование дерева классов
- (2) тестирование последовательностей прямых вызовов методов с помощью Р-путей
- (3) тестирование последовательностей обработки сообщений с помощью ММ-путей
- (4) тестирование исключительно цепочек вызовов процедур программы

43. Какие этапы включает методика ООП при тестировании программного комплекса?

- (1) тестирование методов каждого класса программного комплекса
- (2) тестирование отношений между классами с помощью тестов на основе Р-путей или ММ-путей
- (3) тестирование взаимодействия модулей по всей иерархии комплекса

44. Каковы особенности модели объектно-ориентированной программы?

- (1) использование исключительно Р-путей
- (2) использование исключительно ММ-путей

(3) использование Р-путей и ММ-путей

45. Какие возможности используются для построения дуг графовой модели в ООП?

- (1) прямой вызов видимого метода из кода вызывающего метода
- (2) вызов метода через обработку сообщения
- (3) изображение множества видимых методов

46. Какова формула оценки сложности интеграционного тестирования для ООП?

- (1) $V(P, C) = \sum V(Cls_i, C)$
- (2) $V(P, C) = \sum f_i(Kmsg, Kem)$
- (3) $V(P, C) = \sum Kmsg_i + Kem_i$

47. Какие этапы методики тестирования используются в ООП?

- (1) тестирование методов каждого класса программы
- (2) тестирование методов класса, входящих в его контекст
- (3) тестирование дерева классов программного проекта, включающего оттестированный класс

48. Какие этапы методов тестирования класса используются в ООП?

- (1) тестирование класса как модуля по выбранному критерию
- (2) тестирование класса как иерархической структуры
- (3) тестирование классов, входящих в модель проекта

49. Какие возможности переиспользования тестов предоставляет ООП?

- (1) переиспользование тестов методов при модульном тестировании
- (2) переиспользование тестов методов при интеграционном тестировании классов
- (3) переиспользование тестов классов при интеграционном тестировании проекта

50. Каковы особенности системного тестирования?

- (1) тестированию подлежит система в целом
- (2) тесты оперируют пользовательским или другими внешними интерфейсами
- (3) тестирование осуществляется по методу «черного ящика»
- (4) структура проекта тестируется на уровне подсистем

51. Какие задачи решаются на этапе системного тестирования?

- (1) выявление дефектов в функционировании приложения или в работе с ним
- (2) выявление дефектов использования ресурсов
- (3) выявление несовместимости с окружением
- (4) выявление непредусмотренных сценариев применения или использования непредусмотренных комбинаций данных

52. Какие категории тестов разрабатываются для системного тестирования?

- (1) тесты для проверки полноты функциональности
- (2) тесты для проверки корректности использования ресурсов
- (3) тесты для проверки стрессовых режимов и оценки производительности
- (4) тесты для проверки защиты от искаженных данных и некорректных действий

(5) тесты для проверки инсталляции и конфигурации для платформ, предусмотренных спецификацией

(6) тесты для проверки корректности пользовательской документации

53. Каковы особенности регрессионного тестирования?

(1) регрессионное тестирование является подмножеством системного тестирования

(2) выбор между полным и частичным перетестированием и пополнением тестовых наборов

(3) перетестирование предусматривает только контроль частей приложения, связанных с изменениями

54. Какие задачи решает тестировщик, проверяя изменения, внесенные разработчиком в код?

(1) проверка и подтверждение исправления дефекта

(2) проверка возможности воспроизведения ошибки каким-либо другим способом

(3) перетестирование последствий исправлений на предмет обнаружения

индуцированных ошибок

(4) обеспечение гарантий, что качество исправленного кода не ухудшилось

55. Какие типы дефектов выявляются при системном или регрессионном тестировании?

(1) отсутствующая или некорректная функциональность

(2) непредусмотренные данные или неподдерживаемые сценарии использования

(3) ошибки переносимости на другие платформы

(4) ошибки инсталляции и конфигурирования

(5) ошибки пользовательской документации

(6) некорректность проектной документации

56. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия информации об изменениях в программе?

(1) нет

(2) да

57. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функцию А?

(1) тесты, пригодные для повторного использования

(2) тесты, требующие повторного запуска

(3) устаревшие тесты

(4) новые тесты

58. Какие методы регрессионного тестирования применяются в условиях высоких требований к качеству программного продукта?

(1) метод повторного прогона всех тестов

(2) случайные методы

(3) безопасные методы

(4) методы минимизации

(5) методы, основанные на покрытии кода

59. Какими свойствами обладает метод повторного прогона всех тестов?

- (1) полнота
- (2) точность
- (3) эффективность
- (4) универсальность

60. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия тестов, использовавшихся при тестировании предыдущих версий программы?

- (1) нет
- (2) да

61. При создании очередной версии программы была добавлена функция A, функция D была удалена, функция C – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функцию D?

- (1) тесты, пригодные для повторного использования
- (2) тесты, требующие повторного запуска
- (3) устаревшие тесты
- (4) новые тесты

62. Какие методы регрессионного тестирования применяются в условиях отсутствия программных средств поддержки регрессионного тестирования?

- (1) метод повторного прогона всех тестов
- (2) случайные методы
- (3) безопасные методы
- (4) методы минимизации
- (5) методы, основанные на покрытии кода

63. Какими свойствами обладает метод random(50)?

- (1) полнота
- (2) точность
- (3) эффективность
- (4) универсальность

64. Какие существуют особенности тестовых наборов, используемых в промышленных проектах?

- (1) тестовые наборы разрабатываются параллельно с разработкой приложения с момента согласования требований на него
- (2) тестовые наборы покрывают каждое требование, зафиксированное в спецификации требований
- (3) к началу фазы системного тестирования разрабатываются или автоматически генерируются тысячи тестовых наборов
- (4) автоматическое тестирование применяется в промышленных проектах

65. Какие этапы процесса тестирования поддаются автоматизации?

- (1) генерация тестовых наборов
- (2) автоматический прогон тестов
- (3) автоматическое сравнение результатов исполнения теста с эталонными и фиксация результатов в Log-файле
- (4) автоматическое прекращение тестирования при достижении заданного покрытия

66. Какие этапы процесса тестирования выполняются вручную?

- (1) описание тестовых сценариев на языке спецификаций
- (2) настройка на особенности целевой платформы
- (3) настройка системы автоматизации тестирования (testbench) на особенности окружения тестируемого приложения
- (4) настройка системы синхронизации testbench (выдающей воздействия и принимающей реакции) с приложением

67. Какую информацию использует и производит система автоматизации тестирования в цикле тестирования?

- (1) набор тестов, достаточный для покрытия в соответствии с выбранным критерием
- (2) результаты тестового прогона, зафиксированные в Log-файле
- (3) статистика тестового цикла, содержащая сведения о причине прекращения прогона, о достигнутой степени покрытия, о структуре и количестве обнаруженных дефектов
- (4) анализ причин обнаружения дефектов

68. Какие два из перечисленных методов тестирования наиболее затратны?

- (1) статические методы
- (2) модульное тестирование
- (3) интеграционное тестирование
- (4) системное тестирование с моделируемым окружением
- (5) системное тестирование в реальном окружении и реальном времени

69. Какие два из перечисленных методов тестирования дают наиболее надежные результаты?

- (1) статические методы
- (2) модульное тестирование
- (3) интеграционное тестирование
- (4) системное тестирование с моделируемым окружением
- (5) системное тестирование в реальном окружении и реальном времени

70. Какие активности входят в систему обеспечения качества программного продукта?

- (1) тестирование
- (2) анализ дизайна
- (3) обзоры кода
- (4) аудиты процесса разработки
- (5) ревьюирование проектной документации

71. Какова процедура выбора и оценки критериев качества программного продукта?

- (1) определение заинтересованных сторон в качестве проекта
- (2) определение критериев качества, важных для каждого из участников
- (3) приоритезация критериев с учетом веса участников
- (4) определение набора критериев с определением целей по каждому критерию
- (5) определение стратегии тестирования, исходя из критериев и целей

72. Какие основные уровни выделены в процессе тестирования?

- (1) модульное тестирование
- (2) интеграционное тестирование
- (3) системное тестирование
- (4) регрессионное тестирование

73. Как определить цели тестирования программного проекта?

- (1) определить части проекта, подлежащие тестированию
- (2) какие их свойства и характеристики подлежат тестированию
- (3) каков критерий качества тестирования
- (4) каков график выполнения задач тестирования

74. Как осуществить планирование тестирования?

- (1) определение необходимых человеческих, программных и аппаратных ресурсов
- (2) разработка графика тестовых циклов и назначение задач на ресурсы
- (3) согласование графика тестирования с графиком разработки
- (4) планирование графика обнаружения ошибок в процессе тестирования

75. Какие существуют условия проведения тестирования?

- (1) наличие build – исполняемой версии приложения
- (2) наличие набора тестов для автоматического прогона
- (3) наличие тестовых процедур – сценариев ручного выполнения проверок

76. Каковы особенности тестового цикла?

- (1) прогон разработанных циклов на специально зафиксированном разработчиками срезе кода системы (Build)
- (2) подготовка конфигурации тестовой машины в соответствии с требованиями
- (3) подготовка конфигурации тестового набора и build, их фиксация (тегирование) в системе версионного контроля

77. Что такое прогон тестов?

- (1) исполнение тестового набора в соответствии с задокументированными процедурами
- (2) сохранение тестовых протоколов (test-log)
- (3) анализ протоколов тестирования и принятие решения о прохождении или не прохождении (pass/fail) тестов

78. Чем отличается финальный цикл тестирования?

- (1) код последнего перед финальным циклом build не должен включать изменений
- (2) количество и структура дефектов, найденных на финальном цикле, не противоречат критериям приемки продукта

(3) финальный цикл гарантирует заказчику качество продукта, оговоренное в спецификации

79. Какую информацию должен содержать тестовый план?

- (1) тестовые ресурсы
- (2) перечень функций и подсистем, подлежащих тестированию
- (3) тестовую стратегию
- (4) расписание тестовых циклов
- (5) тестовую конфигурацию
- (6) тестовые метрики
- (7) дизайн тестовых наборов

80. Как определяется тестовая стратегия?

- (1) анализ архитектуры тестируемого приложения для выявления мест, по всей вероятности, содержащих дефекты
- (2) определение и обоснование стратегии выбора входных данных, достаточных для достижения требуемого покрытия
- (3) определение потребности в системе автоматизации тестирования

81. Какие метрики обычно входят в список тестовых?

- (1) степень покрытия тестами набора требований
- (2) степень покрытия тестами тестируемого кода
- (3) структура дефектов: количество и уровень серьезности
- (4) объем тестового кода и количество тестов
- (5) плотность дефектов

82. Какие существуют типы тестирования по способу выбора входных данных?

- (1) функциональное, с покрытием функциональных требований и сценариев использования
- (2) стрессовое, с покрытием экстремальных режимов использования приложения
- (3) тестирование граничных значений и производительности
- (4) тестирование соответствия стандартам
- (5) тестирование совместимости с другими программно-аппаратными комплексами/платформами

83. Какова методика разработки сценарных тестов?

- (1) определение модели окружения, с явным выделением объектов, с которыми приложение обменивается информацией
- (2) разработка параметризованных сценариев использования продукта, например, на языке MSC
- (3) разработка или генерация набора тестов, покрывающего сценарии

84. Как разрабатываются тесты для тестирования спецификаций?

- (1) неформальный тест спецификации требования на естественном языке нормализуется, чтобы зафиксировать все условия воздействий и ожидаемых реакций
- (2) текст требования формализуется, если это возможно

(3) для каждого требования пишется или генерируется набор тестов, который при выполнении должен гарантировать проверку этого требования в продукте

85. Каковы преимущества описания тестовых наборов на языке спецификаций?

- (1) графическое представление и связанная с ним наглядность
- (2) символическое обозначение транзакций и параметров
- (3) возможность отображения параллельных процессов
- (4) снижение в несколько раз трудоемкости за счет уменьшения ручного труда

86. Почему MSC спецификация обеспечивает снижение трудоемкости тестирования?

- (1) MSC описывает множество инвариантных сценариев, отличающихся численными значениями символических параметров
- (2) MSC позволяет сгенерировать сотни тестов, а соответствующий testbench автоматически прогнать их
- (3) одна MSC может кодировать множество параллельных или недетерминированных сценариев

87. Каковы преимущества автоматизированного тестирования?

- (1) строгий контроль результатов по эталонам
- (2) высокая скорость выполнения тестового набора и возможность повторяемости
- (3) высокая надежность, не зависящая от времени тестового прогона

88. Какие существуют особенности документа для описания тестовых процедур?

- (1) содержат описание последовательности действий, необходимых для выполнения тестового набора
- (2) процедуры должны быть сформулированы так, чтобы их мог выполнить инженер, незнакомый с данным проектом
- (3) процедуры для автоматизированных тестов должны содержать только информацию для запуска и анализа результатов
- (4) процедуры автоматически выполняют тестовые наборы

89. Каковы особенности документа для описания тестов?

- (1) содержат информацию для анализа и поддержки тестового набора
- (2) содержат информацию, связывающую любую функцию с тестами, в которых она используется или связи любого теста со всеми используемыми функциями и их сочетаниями
- (3) содержат информацию о структуре и взаимосвязях тестовых файлов
- (4) концептуальное описание системы автоматизации тестирования
- (5) описание расписания тестирования

90. Каковы особенности документа для описания дефектов?

- (1) номер теста, обнаруживавшего дефект
- (2) уровень серьезности дефекта
- (3) поле записи содержит номер build, на котором дефект был найден
- (4) описание дефекта и описание процедуры его воспроизведения

91. Какие бывают состояния дефекта?

- (1) New – дефект занесен в базу дефектов
- (2) Open – дефект зафиксирован за разработчиком для исправления
- (3) Resolved – дефект разработчиком исправлен
- (4) Verified – успешное исправление дефекта подтверждено инженером по качеству (SQE)
- (5) Postponed – решение о замораживании активности по исправлению дефекта

92. Какая информация должна сопровождать действие по исправлению ошибки и перевод дефекта в состояние Resolved?

- (1) причину возникновения дефекта
- (2) место исправления дефекта
- (3) краткий комментарий сделанных исправлений

93. Каково содержание тестового отчета?

- (1) перечень функциональности, запланированной на тестирование
- (2) количество выполненных тестов и время тестирования
- (3) количество найденных и повторно открытых дефектов
- (4) фиксацию отклонений от процедуры тестирования
- (5) заключение о корректировках тестового набора перед следующим циклом тестирования

94. Какие тестовые метрики используются при тестировании?

- (1) покрытие функциональных требований и покрытие кода продукта
- (2) покрытие множества сценариев
- (3) количество и плотность найденных дефектов
- (4) скорость нахождения дефектов

95. Каковы цели обзора тестовой стратегии?

- (1) установить достаточность проверок при тестировании
- (2) проанализировать оптимальность покрытия
- (3) проанализировать оптимальность подхода к автоматизации тестирования

96. Каковы цели обзора тестового кода?

- (1) установить соответствие тестового набора тестовой стратегии
- (2) проверить правильность кодирования тестов
- (3) оценить степень качества кода, исходя из требований по стандартам, простоте поддержки, наличию комментариев и т.п.

97. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия информации об изменениях в программе?

- (1) нет
- (2) да

98. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функцию А?

- (1) тесты, пригодные для повторного использования
- (2) тесты, требующие повторного запуска
- (3) устаревшие тесты
- (4) новые тесты

99. Какие методы регрессионного тестирования применяются в условиях высоких требований к качеству программного продукта?

- (1) метод повторного прогона всех тестов
- (2) случайные методы
- (3) безопасные методы
- (4) методы минимизации
- (5) методы, основанные на покрытии кода

100. Какими свойствами обладает метод повторного прогона всех тестов?

- (1) полнота
- (2) точность
- (3) эффективность
- (4) универсальность

101. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия тестов, использовавшихся при тестировании предыдущих версий программы?

- (1) нет
- (2) да

102. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функцию D?

- (1) тесты, пригодные для повторного использования
- (2) тесты, требующие повторного запуска
- (3) устаревшие тесты
- (4) новые тесты

103. Какие методы регрессионного тестирования применяются в условиях отсутствия программных средств поддержки регрессионного тестирования?

- (1) метод повторного прогона всех тестов
- (2) случайные методы
- (3) безопасные методы
- (4) методы минимизации
- (5) методы, основанные на покрытии кода

104. Какими свойствами обладает метод random(50)?

- (1) полнота

- (2) точность
- (3) эффективность
- (4) универсальность

105. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия информации о покрытии кода тестами?

- (1) нет
- (2) да

106. При создании очередной версии программы была добавлена функция A, функция D была удалена, функция C – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функции C и U?

- (1) тесты, пригодные для повторного использования
- (2) тесты, требующие повторного запуска
- (3) устаревшие тесты
- (4) новые тесты

107. Какие методы регрессионного тестирования применяются, если исходный набор состоит из структурных тестов?

- (1) метод повторного прогона всех тестов
- (2) случайные методы
- (3) безопасные методы
- (4) методы минимизации
- (5) методы, основанные на покрытии кода

108. Какими свойствами обладает метод минимизации с использованием средства поддержки регрессионного тестирования, ориентированного на язык Java, если время его работы оценивается как $O(|P|^2)$?

- (1) полнота
- (2) точность
- (3) эффективность
- (4) универсальность

109. Расходы на внедрение метода выборочного регрессионного тестирования...

- (1) равны расходам на внедрение метода повторного прогона всех тестов
- (2) больше расходов на внедрение метода повторного прогона всех тестов
- (3) меньше расходов на внедрение метода повторного прогона всех тестов
- (4) могут быть больше или меньше расходов на внедрение метода повторного прогона всех тестов

110. Время тестирования при использовании метода выборочного регрессионного тестирования (с учетом времени работы самого метода)...

- (1) равно времени тестирования при использовании метода повторного прогона всех тестов
- (2) больше времени тестирования при использовании метода повторного прогона всех тестов

(3) меньше времени тестирования при использовании метода повторного прогона всех тестов

(4) может быть больше или меньше времени тестирования при использовании метода повторного прогона всех тестов

111. В среднем, метод выборочного регрессионного тестирования...

(1) обнаруживает столько же ошибок, сколько и метод повторного прогона всех тестов

(2) обнаруживает больше ошибок, чем метод повторного прогона всех тестов

(3) обнаруживает меньше ошибок, чем метод повторного прогона всех тестов

(4) может обнаруживать больше или меньше ошибок, чем метод повторного прогона всех тестов

112. Сопровождение какого типа соответствует случаю реализации новых требований пользователя?

(1) корректирующее

(2) адаптивное

(3) усовершенствующее (прогрессивное)

113. Сопровождение какого типа соответствует случаю исправления ошибки в коде?

(1) корректирующее

(2) адаптивное

(3) усовершенствующее (прогрессивное)

114. Сопровождение какого типа соответствует случаю улучшения временных характеристик программы?

(1) корректирующее

(2) адаптивное

(3) усовершенствующее (прогрессивное)

115. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функцию А?

(1) тесты, пригодные для повторного использования

(2) тесты, требующие повторного запуска

(3) устаревшие тесты

(4) новые тесты

116. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функцию D?

(1) тесты, пригодные для повторного использования

(2) тесты, требующие повторного запуска

(3) устаревшие тесты

(4) новые тесты

117. При создании очередной версии программы была добавлена функция A, функция D была удалена, функция C – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функции C и U?

Ответ:

- (1) тесты, пригодные для повторного использования
- (2) тесты, требующие повторного запуска
- (3) устаревшие тесты
- (4) новые тесты

118. Функция `intfunction(int number)` была изменена и приобрела вид `int function(char* string)`. На каком уровне возможно повторное использование теста «1»?

- (1) 1
- (2) 2
- (3) 3
- (4) 4

119. Функция `intfunction(intnumber)` была изменена и возвращает $(number + 2)$ вместо $(number + 1)$. На каком уровне возможно повторное использование теста «1»?

- (1) 1
- (2) 2
- (3) 3
- (4) 4

120. Функция `intfunction(intnumber)` была изменена путем вставки вызова функции распечатки перед первым исполняемым оператором. На каком уровне возможно повторное использование теста «1»?

- (1) 1
- (2) 2
- (3) 3
- (4) 4

121. Какими свойствами обладает метод повторного прогона всех тестов?

- (1) полнота
- (2) точность
- (3) эффективность
- (4) универсальность

122. Какими свойствами обладает метод `random(50)`?

- (1) полнота
- (2) точность
- (3) эффективность
- (4) универсальность

123. Какими свойствами обладает метод минимизации с использованием средства поддержки регрессионного тестирования, ориентированного на язык Java, если время его работы оценивается как $O(|P|^2)$?

- (1) полнота
- (2) точность
- (3) эффективность
- (4) универсальность

124. Какие методы регрессионного тестирования применяются в условиях высоких требований к качеству программного продукта?

- (1) метод повторного прогона всех тестов
- (2) случайные методы
- (3) безопасные методы
- (4) методы минимизации
- (5) методы, основанные на покрытии кода

125. Какие методы регрессионного тестирования применяются в условиях отсутствия программных средств поддержки регрессионного тестирования?

- (1) метод повторного прогона всех тестов
- (2) случайные методы
- (3) безопасные методы
- (4) методы минимизации
- (5) методы, основанные на покрытии кода

126. Какие методы регрессионного тестирования применяются, если исходный набор состоит из структурных тестов?

- (1) метод повторного прогона всех тестов
- (2) случайные методы
- (3) безопасные методы
- (4) методы минимизации
- (5) методы, основанные на покрытии кода

127. Дано: функция P, ее измененная версия P' и набор тестов T, разработанный для тестирования P. Требуется, используя безопасный метод, отобрать подмножество T' для тестирования P'.

P	P'	T
int abs(int number)	int abs(int number)	1. -1
{	{	2. 0
if (number >= 0)	if (number >= 0)	3. 1
return -number;	return number;	
else	else	
return -number;	return -number;	
}	}	

- (1) 1
- (2) 2
- (3) 3
- (4) 1, 2
- (5) 1, 3
- (6) 2, 3

(7) 1, 2, 3

128. Дана функция P, ее измененная версия P' и набор тестов T, разработанный для тестирования P. Требуется, используя метод минимизации, отобрать подмножество T' для тестирования P'.

P int abs(int number) { if (number >= 0) return -number; else return -number; }	P' int abs(int number) { if (number >= 0) return number; else return -number; }	T 1. -1 2. 0 3. 1
--	--	--

- (1) 1
- (2) 2
- (3) 3
- (4) 1, 2
- (5) 1, 3
- (6) 2, 3
- (7) 1, 2, 3

129. Дана функция P, ее измененная версия P' и набор тестов T, разработанный для тестирования P. Требуется, используя метод стопроцентного покрытия кода, отобрать подмножество T' для тестирования P'.

P int abs(int number) { if (number >= 0) return -number; else return -number; }	P' int abs(int number) { if (number >= 0) return number; else return -number; }	T 1. -1 2. 0 3. 1
--	--	--

- (1) 1
- (2) 2
- (3) 3
- (4) 1, 2
- (5) 1, 3
- (6) 2, 3
- (7) 1, 2, 3

130. Тестируемая программа состоит из модулей A, B, C и D, взаимодействующих по принципу «каждый с каждым». Модули A и B были изменены. Тестирование каких интерфейсов необходимо обеспечить, если используется брандмауэр?

- (1) A – B
- (2) A – C
- (3) A – D
- (4) B – C

(5) B – D

(6) C – D

131. Тестируемая программа состоит из классов A, B и C, взаимодействующих по принципу «каждый с каждым». Классы A и B были изменены. Тестирование каких классов и интерфейсов необходимо обеспечить, если используется брандмауэр?

(1) A

(2) B

(3) C

(4) A – B

(5) A – C

(6) B – C

132. Класс A является потомком класса B, который, в свою очередь, является потомком класса C. Класс B был изменен. Тестирование каких классов необходимо обеспечить?

(1) A

(2) B

(3) C

133. Какими преимуществами обладает методика уменьшения объема тестируемой программы?

(1) уменьшается время компиляции тестируемой программы

(2) уменьшается время выполнения тестируемой программы

(3) уменьшается время работы метода отбора тестов

(4) уменьшается риск пропуска ошибки

134. Какими преимуществами обладает методика упорядочения тестов?

(1) уменьшается время работы метода отбора тестов

(2) увеличивается частота обнаружения ошибок

(3) увеличивается скорость покрытия кода

135. Какими преимуществами обладает методика предсказания целесообразности отбора тестов?

(1) уменьшается время работы метода отбора тестов, в случае если выборочное регрессионное тестирование целесообразно

(2) уменьшается время работы метода отбора тестов, в случае если выборочное регрессионное тестирование нецелесообразно

(3) точность предсказания от версии к версии повышается

136. Модуль E исходной программы содержал ошибку. Ее исправление потребовало изменения модулей B, D и E. Тест 1 покрывает модули A, B и C. Тест 2 покрывает модули A, C и E. Тест 3 покрывает модули D и E. Каким должен быть порядок прогона тестов, если при упорядочении ставится цель скорейшего достижения наибольшей степени покрытия модулей?

(1) 1, 2, 3

(2) 1, 3, 2

- (3) 2, 1, 3
- (4) 2, 3, 1
- (5) 3, 1, 2
- (6) 3, 2, 1

137. Модуль E исходной программы содержал ошибку. Ее исправление потребовало изменения модулей B, D и E. Тест 1 покрывает модули A, B и C. Тест 2 покрывает модули A, C и E. Тест 3 покрывает модули D и E. Каким должен быть порядок прогона тестов, если при упорядочении ставится цель скорейшего достижения наибольшей степени покрытия измененных модулей?

- (1) 1, 2, 3
- (2) 1, 3, 2
- (3) 2, 1, 3
- (4) 2, 3, 1
- (5) 3, 1, 2
- (6) 3, 2, 1

138. Модуль E исходной программы содержал ошибку. Ее исправление потребовало изменения модулей B, D и E. Тест 1 покрывает модули A, B и C. Тест 2 покрывает модули A, C и E. Тест 3 покрывает модули D и E. Каким должен быть порядок прогона тестов, если при упорядочении ставится цель скорейшего роста вероятности того, что тестируемая система надежна?

- (1) 1, 2, 3
- (2) 1, 3, 2
- (3) 2, 1, 3
- (4) 2, 3, 1
- (5) 3, 1, 2
- (6) 3, 2, 1

139. Ожидаемое время работы метода отбора тестов – 1 час. Среднее время выполнения одного теста – 1 час. Тестовый набор состоит из 8 тестов. Каково значение порога целесообразности?

- (1) 1
- (2) 2
- (3) 6
- (4) 7
- (5) 8

140. Ожидаемое время работы метода отбора тестов – 2 часа. Среднее время выполнения одного теста – 1 час. Тестовый набор состоит из 8 тестов. Каково значение порога целесообразности?

- (1) 1
- (2) 2
- (3) 6
- (4) 7
- (5) 8

141. Ожидаемое время работы метода отбора тестов – 2 часа. Среднее время выполнения одного теста – 2 часа. Тестовый набор состоит из 8 тестов. Каково значение порога целесообразности?

- (1) 1
- (2) 2
- (3) 6
- (4) 7
- (5) 8

142. На предыдущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии В, а тест 3 – в состоянии С. На текущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии С, а тест 3 – в состоянии D. На базе какого теста наиболее целесообразна разработка новых тестов?

- (1) 1
- (2) 2
- (3) 3

143. На предыдущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии В, а тест 3 – в состоянии С. На текущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии С, а тест 3 – в состоянии D. На базе каких состояний возможна разработка новых тестов?

- (1) А
- (2) В
- (3) С
- (4) D

144. На предыдущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии В, а тест 3 – в состоянии С. На текущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии С, а тест 3 – в состоянии D. На базе какого состояния наиболее целесообразна разработка новых тестов?

- (1) А
- (2) В
- (3) С
- (4) D

145. Какой этап регрессионного тестирования не имеет аналога в алгоритме обычного тестирования?

- (1) предсказание целесообразности
- (2) идентификация изменений
- (3) отбор тестов
- (4) выполнение тестов
- (5) создание дополнительных тестов
- (6) обновление базы данных

146. На каком этапе регрессионного тестирования проводится упорядочение тестов?

- (1) предсказание целесообразности
- (2) идентификация изменений

- (3) отбор тестов
- (4) выполнение тестов
- (5) создание дополнительных тестов
- (6) обновление базы данных

147. На каком этапе регрессионного тестирования удаляются устаревшие тесты?

- (1) предсказание целесообразности
- (2) идентификация изменений
- (3) отбор тестов
- (4) выполнение тестов
- (5) создание дополнительных тестов
- (6) обновление базы данных

148. Какие из перечисленных средств использует в своей работе система поддержки регрессионного тестирования?

- (1) систему управления конфигурацией
- (2) базу данных дефектов
- (3) отладчик
- (4) профайлер

149. Какие операции позволяет автоматизировать система поддержки регрессионного тестирования?

- (1) отбор тестов
- (2) уменьшение объема тестируемой программы
- (3) упорядочение тестов
- (4) предсказание целесообразности

150. Какие операции при использовании системы поддержки регрессионного тестирования выполняются вручную?

- (1) идентификация различий между версиями программы
- (2) вычисление списка переменных, определяющих состояние
- (3) исключение нештатных состояний
- (4) вычисление векторов отличий

КЛЮЧ

1	1,2	51	1,2,3,4	101	1
2	1,2,3	52	1,2,3,4,5,6	102	3
3	1,2,3	53	1,2,3	103	1,2
4	2	54	1,2,3,4	104	3,4
5	1,2,3	55	1,2,3,4,5	105	1
6	1,2,3	56	1	106	2
7	1,2,3	57	4	107	4,5
8	1,2	58	1,3	108	2
9	2	59	1,3,4	109	2
10	1,2	60	1	110	4
11	1,2	61	3	111	3

12	1,3,4	62	1,2	112	2
13	1,2,3,4	63	3,4	113	1
14	1	64	1,2,3	114	3
15	1,2,3	65	1,2,3,4	115	4
16	3	66	1,2,3,4	116	3
17	2,3	67	1,2,3	117	2
18	1,2,3,4,5	68	4,5	118	1
19	2,3	69	4,5	119	2
20	1,2,3	70	1,2,3,4,5	120	3
21	1,3	71	1,2,3,4,5	121	1,3,4
22	1	72	1,2,3	122	3,4
23	1,2,3	73	1,2,3	123	2
24	1,2,3	74	1,2,3	124	1,3
25	1,2,3	75	1,2,3	125	1,2
26	1,2,3,4,5	76	1,2,3	126	4,5
27	1,2,3	77	1,2,3	127	6
28	1,2,3,4	78	1,2,3	128	2,3
29	1,2	79	1,2,3,4,5,6	129	4,5
30	1,2,3	80	1,2,3	130	1,2,3,4,5
31	1,2,3	81	1,2,3,4,5	131	1,2,4,5,6
32	1,2,3	82	1,2,3,4,5	132	1,2
33	1,3	83	1,2,3	133	1
34	1,2,3	84	1,2,3	134	2,3
35	1,2	85	1,2,3,4	135	2
36	1,2,3	86	1,2,3	136	2
37	1,2,3	87	1,2,3	137	5
38	1,2	88	1,2,3	138	6
39	1	89	1,2,3,4	139	4
40	1,2,3	90	1,2,3,4	140	3
41	1,2,4	91	1,2,3,4,5	141	4
42	1,2,3	92	1,2,3	142	3
43	1,2,3	93	1,2,3,4,5	143	1,3,4
44	3	94	1,2,3,4	144	4
45	1,2	95	1,2,3	145	1
46	1,2	96	1,2,3	146	3
47	1,2,3	97	1	147	6
48	1,2	98	4	148	1,4
49	2,3	99	1,3	149	1,3
50	1,2,3	100	1,3,4	150	5

5.8 Вопросы к экзамену по дисциплине «Промышленное тестирование программного обеспечения»

1. Определение процесса тестирования, его задачи.
2. Тестирование и отладка, сходство и различия.
3. Место тестирования в жизненном цикле информационной системы.
4. Фазы и технология тестирования.
5. Структурные критерии выбора тестов.
6. Функциональные критерии выбора тестов.
7. Стохастические критерии выбора тестов.
8. Мутационные критерии выбора тестов.

9. Оценки покрытия проекта.
10. Модульное тестирование. Цели и задачи.
11. Подходы к тестированию на основе потока управления.
12. Подходы к тестированию на основе потока данных.
13. Тестирование белого, черного и серого ящика.
14. Тестирование на основе инвариантов.
15. Использование случайных величин в тестировании.
16. Интеграционное тестирование. Цели и задачи.
17. Подходы монолитного, инкрементального, нисходящего и восходящего тестирования.
18. Особенности интеграционного тестирования объектно-ориентированных программ).
19. Системное тестирование. Цели и задачи.
20. Тестирование пользовательского интерфейса.
21. Регрессионное тестирование.
22. Структура и шаблоны тестов для формирования тестового набора для автоматического прогона.
23. Ручные и автоматизированные тесты, описание тестовых наборов и тестовых отчетов.
24. Жизненный цикл дефекта.
25. Метрики, используемые при тестировании.
26. Стандарты документации по тестированию - план тестирования, проект тестирования, тестовые варианты, тестовые процедуры, журнал испытаний и пр.
27. Особенности экстремального программирования.
28. Принципы разработки через тестирования.
29. Основные шаблоны тестирования.
30. Определение затрат на проведение тестирования.