

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЛГПУ»)**

Структурное подразделение Институт физико-математического
образования, информационных и обслуживающих технологий
Кафедра информационных образовательных технологий и систем

УТВЕРЖДАЮ

Директор ИФМОИОТ

Е.Е. Горбенко

2023 г.



Приложение к рабочей программе учебной дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля и промежуточной аттестации
обучающихся по дисциплине
«Тестирование и отладка программного обеспечения»

По направлению подготовки 09.03.04 Программная инженерия

Профиль подготовки Программное обеспечение систем и комплексов

Квалификация выпускника – бакалавр

Форма обучения очная, заочная

Курс ОФО – 4 курс, ЗФО – 4 курс

Разработчик

Нечай Т.А.

канд. техн. наук, ассистент кафедры
информационных технологий и
систем

Заведующий кафедрой

Д.А. Капустин

Протокол от «24» ноября 2023 г. №8

Луганск, 2023

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины (модуля) Тестирование и отладка программного обеспечения и предназначен для контроля и оценки образовательных достижений студентов, освоивших программу дисциплины (модуля).

1.2. Цели и задачи фонда оценочных средств

Цель ФОС – установить соответствие уровня подготовки обучающегося требованиям ФГОС ВО бакалавриата по направлению подготовки 09.03.04 Программная инженерия, утвержденным приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 920 и Профессиональным стандартом, утвержденным Приказом Министерства труда и социальной защиты Российской Федерации «Об утверждении профессионального стандарта 06.001 «Программист» от 20.07.2022 № 424н.

1.3. Перечень компетенций, формируемых в процессе освоения основной образовательной программы

Процесс освоения дисциплины направлен на формирование следующих компетенций и индикаторов их достижения:

Код по ФГОС ВО	Индикатор достижения
Профессиональные	
ПК-4. Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества	ПК-4.1. Знать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли человека-оператора, процессов, методов, инструментов и технологий обеспечения качества ПК-4.2. Уметь определять оптимальные методы и технологии разработки компьютерных программ, обеспечивающие достаточный уровень качества программного обеспечения (надежности, безопасности, удобства использования), с учетом роли человека-оператора, процессов, методов, инструментов и технологий обеспечения качества ПК-4.3. Владеть навыками разработки компьютерных программ, с заданными потребительскими свойствами с учётом комплекса условий их эксплуатации

1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. Основные понятия тестирования и обеспечения качества.	ПК-4	Выполнение лабораторных работ
Тема 2. Тестирование на ранних этапах разработки.	ПК-4	Выполнение лабораторных работ
Тема 3. Модульное тестирование.	ПК-4	Выполнение лабораторных работ
Тема 4. Функциональное тестирование.	ПК-4	Выполнение лабораторных работ
Тема 5. Нефункциональное тестирование.	ПК-4	Выполнение лабораторных работ
Текущая аттестация	ПК-4	Контрольная работа
Промежуточная аттестация	ПК-4	Зачет

1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
ПК-4. Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества	<p>ПК-4.1. Знает концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли человека-оператора, процессов, методов, инструментов и технологий обеспечения качества</p> <p>ПК-4.2. Умеет определять оптимальные методы и технологии разработки компьютерных программ, обеспечивающие достаточный уровень качества программного обеспечения (надежности, безопасности, удобства использования), с учетом роли человека-оператора, процессов, методов, инструментов и технологий обеспечения качества</p> <p>ПК-4.3. Владеет навыками разработки компьютерных программ, с заданными потребительскими свойствами с учётом комплекса условий их эксплуатации</p>

1.6. Критерии оценивания компетенций на разных этапах их формирования

Вид учебной работы	Количество баллов		
8 семестр / 10-12 триместр			
	ОФО	О-ЗФО	ЗФО
Оформление отчетов по лабораторным работам	30 баллов	-	30 баллов
Работа на лабораторных занятиях	30 баллов	-	30 баллов
Выполнение тестовых заданий	-	-	-
Выполнение заданий самостоятельной работы	10 баллов	-	10 баллов
зачета	30 баллов	-	30 баллов
Итого за семестр:	100 баллов	-	100 баллов
Всего	100 баллов		

Накопительная система оценивания по 100-балльной шкале

Четырехбал- льная система оценивания экзамена	100- балльная шкала	Буквенная шкала, соответствующая 100- балльной шкале	Система оценивания зачета
Отлично	90–100	A – 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	Зачтено
Хорошо	83–89	B – 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	
Хорошо	75–82	C – хорошо – теоретическое содержание курса освоено полностью; 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	
Удовлетво- рительно	63–74	D – удовлетворительно – теоретическое содержание дисциплины 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, содержат ошибки	
Удовлетво- рительно	50–62	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	
Неудовлетво- рительно	21–49	FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы не сформированы; большинство предусмотренных программой обучения учебных заданий не выполнено либо качество	Не зачтено

		их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	
Неудовлетворительно	0–20	F – 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	

2. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА

2.1. Оценочные средства текущего контроля (типовые)

Вопросы для текущего контроля:

1. Что такое тестирование программного обеспечения, и какова его основная цель?
2. Чем отличается тестирование от обеспечения качества?
3. Опишите основные уровни тестирования (модульное, интеграционное, системное, приемочное).
4. Что такое тест-план, и какую роль он играет в процессе тестирования?
5. Каковы основные принципы тестирования программного обеспечения?
6. В чем заключается различие между тестированием черного и белого ящика?
7. Что такое баг и каковы основные этапы его жизненного цикла?
8. Какие существуют виды тестирования, и чем они различаются?
9. Что такое регрессионное тестирование и зачем его проводят?
10. Опишите, что такое обеспечение качества (QA) и его основные задачи.
11. Какие роли существуют в команде тестирования и обеспечения качества?
12. Что такое критерии входа и выхода в тестировании?
13. Объясните разницу между валидацией и верификацией.
14. Что такое автоматизация тестирования, и в каких случаях она наиболее эффективна?
15. Каковы основные метрики качества, используемые в тестировании и обеспечении качества?
16. В чем заключается цель тестирования на ранних этапах разработки?
17. Какие преимущества приносит раннее выявление дефектов?
18. Что такое статическое тестирование, и когда его следует применять?
19. Какие инструменты и методы используются для статического тестирования?
20. В чем разница между статическим и динамическим тестированием?
21. Какова роль рецензирования (ревью) кода в обеспечении качества на ранних этапах?
22. Что такое анализ требований, и почему он важен для тестирования на ранних этапах?
23. Как тестирование требований помогает предотвратить дефекты?
24. Что такое модель V-цикла, и как она отражает важность тестирования на ранних этапах?
25. Как участие тестировщиков на этапе проектирования улучшает качество продукта?
26. Какие техники применяются для анализа и тестирования требований?
27. Что такое тестируемость требований, и как её можно улучшить?

28. Как проведение статических проверок способствует снижению затрат на исправление дефектов?
29. В чем заключается роль тестировщиков на этапе планирования архитектуры и дизайна?
30. Какие ключевые аспекты необходимо учесть при тестировании на уровне прототипов и макетов?
31. Что такое модульное тестирование, и какова его основная цель?
32. Какие преимущества дает модульное тестирование?
33. Что такое модуль (unit) в программировании, и каковы его основные характеристики?
34. Чем модульное тестирование отличается от интеграционного?
35. Какие методологии разработки программного обеспечения особенно подходят для использования модульного тестирования?
36. Что такое тестирование на основе методов черного и белого ящика в контексте модульного тестирования?
37. Какие инструменты и библиотеки используются для написания и выполнения модульных тестов?
38. Что такое фреймворк для модульного тестирования, и какие примеры таких фреймворков вы знаете?
39. Какова роль мок-объектов в модульном тестировании?
40. Что такое покрытие кода, и какие его виды существуют (например, покрытие операторов, покрытие ветвей)?
41. Какие подходы существуют для написания эффективных модульных тестов?
42. Каковы основные принципы "хорошего" модульного теста?
43. Что такое тестовый оркестр и его роль в модульном тестировании?
44. В чем заключаются преимущества и недостатки автоматизации модульного тестирования?
45. Как интеграция модульного тестирования с CI/CD (непрерывной интеграцией и доставкой) влияет на процесс разработки?
46. Что такое функциональное тестирование, и какова его основная цель?
47. Чем функциональное тестирование отличается от нефункционального?
48. Какие виды функционального тестирования существуют?
49. В чем заключается процесс тест-дизайна для функционального тестирования?
50. Что такое позитивное и негативное тестирование, и как они применяются в функциональном тестировании?
51. Каковы основные этапы функционального тестирования?
52. Какие типы тестовой документации используются для функционального тестирования?
53. Что такое тестовый сценарий (test case), и каковы его основные элементы?
54. Какова роль анализа требований в функциональном тестировании?
55. Чем отличается тестирование на основе спецификаций от тестирования на основе опыта?

56. Какие методики покрытия функциональных требований существуют (например, эквивалентное разделение, анализ граничных значений)?
57. Что такое пользовательский сценарий (user story) и его связь с функциональным тестированием?
58. Какие инструменты применяются для автоматизации функционального тестирования?
59. Как выполняется регрессионное тестирование в рамках функционального тестирования?
60. Как функциональное тестирование интегрируется в различные методологии разработки (например, Waterfall, Agile)?
61. Что такое нефункциональное тестирование, и чем оно отличается от функционального?
62. Какие основные виды нефункционального тестирования существуют?
63. Что такое тестирование производительности, и какие виды его существуют (нагрузочное, стресс-тестирование, объемное)?
64. В чем заключается цель тестирования безопасности, и какие методы его проведения используются?
65. Какова цель тестирования удобства использования (usability testing), и какие аспекты оно оценивает?
66. Что такое тестирование совместимости, и как оно проводится?
67. Чем тестирование надежности отличается от других видов нефункционального тестирования?
68. Какие метрики применяются для оценки результатов нефункционального тестирования?
69. Каковы основные инструменты для проведения нагрузочного и стресс-тестирования?
70. Что такое тестирование на отказоустойчивость, и почему оно важно?
71. Как определить требования к нефункциональному тестированию на этапе планирования?
72. Как выполняется мониторинг системы во время нефункционального тестирования?
73. В чем заключаются особенности автоматизации нефункционального тестирования?
74. Как интеграция нефункционального тестирования с CI/CD помогает в процессе разработки?
75. Какие подходы и техники используются для тестирования локализации и интернационализации?

2.2. Оценочные средства для промежуточной аттестации

Вопросы для проведения аттестации

1. В чем заключается процесс тест-дизайна для функционального тестирования?
2. В чем заключается различие между тестированием черного и белого ящика?

3. В чем заключается роль тестировщиков на этапе планирования архитектуры и дизайна?
4. В чем заключается цель тестирования безопасности, и какие методы его проведения используются?
5. В чем заключается цель тестирования на ранних этапах разработки?
6. В чем заключаются особенности автоматизации нефункционального тестирования?
7. В чем заключаются преимущества и недостатки автоматизации модульного тестирования?
8. В чем разница между статическим и динамическим тестированием?
9. Как выполняется мониторинг системы во время нефункционального тестирования?
10. Как выполняется регрессионное тестирование в рамках функционального тестирования?
11. Как интеграция модульного тестирования с CI/CD (непрерывной интеграцией и доставкой) влияет на процесс разработки?
12. Как интеграция нефункционального тестирования с CI/CD помогает в процессе разработки?
13. Как определить требования к нефункциональному тестированию на этапе планирования?
14. Как проведение статических проверок способствует снижению затрат на исправление дефектов?
15. Как тестирование требований помогает предотвратить дефекты?
16. Как участие тестировщиков на этапе проектирования улучшает качество продукта?
17. Как функциональное тестирование интегрируется в различные методологии разработки (например, Waterfall, Agile)?
18. Какие виды функционального тестирования существуют?
19. Какие инструменты и библиотеки используются для написания и выполнения модульных тестов?
20. Какие инструменты и методы используются для статического тестирования?
21. Какие инструменты применяются для автоматизации функционального тестирования?
22. Какие ключевые аспекты необходимо учесть при тестировании на уровне прототипов и макетов?
23. Какие методики покрытия функциональных требований существуют (например, эквивалентное разделение, анализ граничных значений)?
24. Какие методологии разработки программного обеспечения особенно подходят для использования модульного тестирования?
25. Какие метрики применяются для оценки результатов нефункционального тестирования?
26. Какие основные виды нефункционального тестирования существуют?

27. Какие подходы и техники используются для тестирования локализации и интернационализации?
28. Какие подходы существуют для написания эффективных модульных тестов?
29. Какие преимущества дает модульное тестирование?
30. Какие преимущества приносит раннее выявление дефектов?
31. Какие роли существуют в команде тестирования и обеспечения качества?
32. Какие существуют виды тестирования, и чем они различаются?
33. Какие техники применяются для анализа и тестирования требований?
34. Какие типы тестовой документации используются для функционального тестирования?
35. Какова роль тоск-объектов в модульном тестировании?
36. Какова роль анализа требований в функциональном тестировании?
37. Какова роль рецензирования (ревью) кода в обеспечении качества на ранних этапах?
38. Какова цель тестирования удобства использования (usability testing), и какие аспекты оно оценивает?
39. Каковы основные инструменты для проведения нагрузочного и стресс-тестирования?
40. Каковы основные метрики качества, используемые в тестировании и обеспечении качества?
41. Каковы основные принципы "хорошего" модульного теста?
42. Каковы основные принципы тестирования программного обеспечения?
43. Каковы основные этапы функционального тестирования?
44. Объясните разницу между валидацией и верификацией.
45. Опишите основные уровни тестирования (модульное, интеграционное, системное, приемочное).
46. Опишите, что такое обеспечение качества (QA) и его основные задачи.
47. Чем модульное тестирование отличается от интеграционного?
48. Чем отличается тестирование на основе спецификаций от тестирования на основе опыта?
49. Чем отличается тестирование от обеспечения качества?
50. Чем тестирование надежности отличается от других видов нефункционального тестирования?
51. Чем функциональное тестирование отличается от нефункционального?
52. Что такое автоматизация тестирования, и в каких случаях она наиболее эффективна?
53. Что такое анализ требований, и почему он важен для тестирования на ранних этапах?

54. Что такое баг и каковы основные этапы его жизненного цикла?
55. Что такое критерии входа и выхода в тестировании?
56. Что такое модель V-цикла, и как она отражает важность тестирования на ранних этапах?
57. Что такое модуль (unit) в программировании, и каковы его основные характеристики?
58. Что такое модульное тестирование, и какова его основная цель?
59. Что такое нефункциональное тестирование, и чем оно отличается от функционального?
60. Что такое позитивное и негативное тестирование, и как они применяются в функциональном тестировании?
61. Что такое покрытие кода, и какие его виды существуют (например, покрытие операторов, покрытие ветвей)?
62. Что такое пользовательский сценарий (user story) и его связь с функциональным тестированием?
63. Что такое регрессионное тестирование и зачем его проводят?
64. Что такое статическое тестирование, и когда его следует применять?
65. Что такое тестирование на основе методов черного и белого ящика в контексте модульного тестирования?
66. Что такое тестирование на отказоустойчивость, и почему оно важно?
67. Что такое тестирование программного обеспечения, и какова его основная цель?
68. Что такое тестирование производительности, и какие виды его существуют (нагрузочное, стресс-тестирование, объемное)?
69. Что такое тестирование совместимости, и как оно проводится?
70. Что такое тестируемость требований, и как её можно улучшить?
71. Что такое тестовый оркестр и его роль в модульном тестировании?
72. Что такое тестовый сценарий (test case), и каковы его основные элементы?
73. Что такое тест-план, и какую роль он играет в процессе тестирования?
74. Что такое фреймворк для модульного тестирования, и какие примеры таких фреймворков вы знаете?
75. Что такое функциональное тестирование, и какова его основная цель?