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
«15» декабря 2023 г.

Приложение к рабочей программе учебной дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине
«Статический анализ программного кода»

По направлению подготовки 09.04.04 Программная инженерия

Профиль подготовки Программное обеспечение систем и комплексов
Квалификация выпускника – магистр
Форма обучения очная, заочная
Курс ОФО – 2 курс, ЗФО – 2 курс

Разработчик
Швыров В.В.
канд. физ.-мат. наук, доцент,
доцент кафедры информационных образовательных 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) Статический анализ программного кода 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 бакалавриат / специалитет / магистратура по направлению подготовки 09.04.04 Программная инженерия, утвержденного приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 932 (с изменениями и дополнениями).

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
Универсальные	
Общепрофессиональные	
ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов	ОПК-8.1. Знать методы эффективного управления разработкой программных средств и проектов ОПК-8.2. Уметь применять эффективное управление разработкой программных средств и проектов ОПК-8.3. Владеть навыками эффективного управления разработкой программных средств и проектов
Профессиональные	
ПК-1. Знание методов организации и управления информационными процессами	ПК-1.1. Знать методы управления информационными процессами ПК-1.2. Уметь управлять проектами по информатизации предприятий ПК-1.3. Владеть навыками практического управления проектами по информатизации предприятий

1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания

Тема 1. Введение. Проблема безопасности ПО и статический анализ.	ПК-1; ОПК-8;	Выполнение лабораторных работ
Тема 2. Применение статического анализа для обнаружения потенциальных угроз. Категории уязвимостей	ПК-1; ОПК-8;	Выполнение лабораторных работ
Тема 3. Автоматизированные инструменты статического анализа.	ПК-1; ОПК-8;	Выполнение лабораторных работ
Тема 4. Статический анализ для различных языков программирования.	ПК-1; ОПК-8;	Выполнение лабораторных работ
Тема 5. Особенности статического анализа программ на Python.	ПК-1; ОПК-8;	Выполнение лабораторных работ
Тема 6. Статический анализ проектов на C#.	ПК-1; ОПК-8;	Выполнение лабораторных работ
Тема 7. Каталог CWE и БДУ.	ПК-1; ОПК-8;	Выполнение лабораторных работ
Тема 8. Методы устранения дефектов в коде.	ПК-1; ОПК-8;	Выполнение лабораторных работ
Текущая аттестация	ПК-1; ОПК-8;	Контрольная работа
Промежуточная аттестация	ПК-1; ОПК-8;	Экзамен (письменный)

1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов	ОПК-8.1. Знает методы эффективного управления разработкой программных средств и проектов ОПК-8.2. Умеет применять эффективное управление разработкой программных средств и проектов ОПК-8.3. Владеет навыками эффективного управления разработкой программных средств и проектов
ПК-1. Знание методов организации и управления информационными процессами	ПК-1.1. Знает методы управления информационными процессами ПК-1.2. Умеет управлять проектами по информатизации предприятий ПК-1.3. Владеет навыками практического управления проектами по информатизации предприятий

1.6. Критерии оценивания компетенций на разных этапах их формирования

Вид учебной работы	Количество баллов		
3 семестр / 4-5 триместр			
	ОФО	**О-ЗФО**	**ЗФО**
Оформление отчетов по лабораторным работам	30 баллов		
Работа на лабораторных занятиях	30 баллов		
Выполнение тестовых заданий	-		
Выполнение заданий самостоятельной работы	10 баллов		
экзамена	30 баллов		
Итого за семестр:	100 баллов		
Всего	100 баллов		

Накопительная система оценивания по 100-балльной шкале

Четырехбалльная 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 их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	Не зачтено
Неудовлетво-рительно	**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 представляет собой статический анализ программного кода?
2. Какие основные проблемы безопасности ПО могут возникнуть на ранних этапах разработки?
3. Как статический анализ помогает в решении проблем безопасности программного обеспечения?
4. В чем различие между статическим и динамическим анализом кода?
5. Какие преимущества предоставляет использование статического анализа для выявления угроз безопасности?
6. Какие типы уязвимостей могут быть обнаружены с использованием статического анализа?
7. Какова роль статического анализа в предотвращении атак, связанных с инъекциями?
8. Какие критические уязвимости безопасности может выявить статический анализ?
9. Как статический анализ может содействовать обеспечению соответствия стандартам безопасности кода?
10. Какие инструменты статического анализа могут использоваться для решения проблем безопасности ПО?
11. Какова роль статического анализа в предотвращении атак на аутентификацию и авторизацию?
12. Как статический анализ может помочь в предотвращении атак связанных с межсайтовым скриптингом?
13. Какие вызовы могут возникнуть при внедрении статического анализа в разработку?
14. В чем заключается принцип "безопасность с самого начала" и как его реализовать с использованием статического анализа?
15. Как статический анализ влияет на общую безопасность программных продуктов?
16. Какие аспекты кода могут быть анализированы статически с точки зрения безопасности?
17. Какие меры безопасности предоставляет статический анализ в контексте обработки ввода данных?
18. Какова роль статического анализа в выявлении потенциальных угроз безопасности в структуре приложения?
19. Как статический анализ может быть интегрирован в процесс непрерывной интеграции и развертывания?
20. Какие проблемы безопасности чаще всего возникают из-за недостаточного внимания к безопасности на ранних этапах разработки?
21. Как статический анализ может содействовать снижению числа уязвимостей, связанных с обработкой файлов?
22. Какие стандарты и рекомендации по безопасности могут быть учтены при использовании статического анализа?

23. Как изменения в коде, предложенные статическим анализом, могут повлиять на общую безопасность приложения?
24. Как статический анализ помогает выявлять и предотвращать утечки конфиденциальной информации?
25. Каковы основные вызовы при внедрении статического анализа в большие и сложные проекты?
26. Как статический анализ может помочь в решении проблем, связанных с недостаточной проверкой входных данных?
27. Какие принципы обеспечения безопасности чаще всего нарушаются при разработке программного обеспечения, и как статический анализ может помочь их соблюсти?
28. Какие рекомендации по обеспечению безопасности могут быть предоставлены статическим анализатором?
29. Как статический анализ влияет на общую производительность и эффективность работы разработчиков в контексте безопасности ПО?
30. Как можно оценить эффективность использования статического анализа для решения проблем безопасности программного обеспечения?
31. Какие категории уязвимостей могут быть выявлены с использованием статического анализа?
32. Как статический анализ помогает в выявлении угроз безопасности на ранних этапах разработки?
33. Какие типы атак на безопасность часто обнаруживаются при статическом анализе кода?
34. Как статический анализ справляется с угрозами, связанными с инъекциями, и какие методы применяются?
35. Какие уязвимости безопасности связаны с недостатками аутентификации и авторизации, и как они могут быть выявлены статическим анализом?
36. Как статический анализ помогает предотвращать атаки, связанные с межсайтовым скриптингом?
37. Какова роль статического анализа в обнаружении угроз безопасности, связанных с обработкой файлов?
38. Какие категории уязвимостей можно выявить с использованием статического анализа в контексте работы с базами данных?
39. Как статический анализ может помочь предотвращению атак, связанных с утечками конфиденциальной информации?
40. Какие угрозы безопасности, связанные с многопоточностью, могут быть выявлены при статическом анализе кода?
41. Как статический анализ обнаруживает уязвимости, связанные с недостаточной обработкой ошибок и исключений?
42. Какие методы статического анализа эффективны для выявления угроз, связанных с недостаточной проверкой входных данных?
43. Как статический анализ влияет на обнаружение и предотвращение атак на аутентификацию и сессии?

44. Какие угрозы безопасности связаны с использованием сторонних библиотек и компонентов, и как их можно выявить статическим анализом?
45. Как статический анализ может помочь в выявлении угроз, связанных с недостаточной защитой криптографических решений?
46. Какие категории уязвимостей могут возникнуть из-за недостаточной обработки и валидации внешних данных?
47. Как статический анализ помогает в выявлении проблем безопасности, связанных с недостаточной защитой конфиденциальных данных в хранилищах?
48. Какие угрозы безопасности связаны с использованием небезопасных функций и операций в коде, и как их можно выявить статическим анализом?
49. Как статический анализ воздействует на обнаружение и предотвращение атак на безопасность, связанных с недостаточной защитой сетевого взаимодействия?
50. Какие меры безопасности предоставляет статический анализ для выявления и устранения уязвимостей, связанных с безопасностью операционной системы?
51. Как статический анализ обнаруживает угрозы безопасности, связанные с использованием слабых алгоритмов шифрования и хеширования?
52. Какие категории уязвимостей можно выявить с использованием статического анализа в области безопасности мобильных приложений?
53. Как статический анализ помогает в предотвращении угроз безопасности, связанных с несанкционированным доступом к данным и ресурсам?
54. Какие угрозы безопасности могут возникнуть из-за недостаточного управления сессиями, и как их можно выявить статическим анализом?
55. Как статический анализ обнаруживает уязвимости, связанные с недостаточной обработкой и контролем ошибок при взаимодействии с внешними системами?
56. Какие преимущества предоставляют автоматизированные инструменты статического анализа?
57. Какие основные задачи решают инструменты статического анализа в процессе разработки?
58. Какие виды ошибок и уязвимостей могут быть выявлены при помощи автоматизированных инструментов статического анализа?
59. Каким образом инструменты статического анализа помогают соблюдать стандарты кодирования и рекомендации по стилю?
60. Какова роль автоматизированных инструментов статического анализа в обеспечении безопасности кода?
61. Какие типы языков программирования поддерживаются большинством инструментов статического анализа?
62. Какие проблемы и ограничения могут возникнуть при использовании автоматизированных инструментов статического анализа?
63. Как инструменты статического анализа могут интегрироваться в процесс непрерывной интеграции и развертывания (CI/CD)?

64. Каким образом автоматизированные инструменты статического анализа помогают в управлении качеством кода?
65. Какие сценарии использования инструментов статического анализа наиболее эффективны в условиях командной разработки?
66. Какие возможности предоставляют современные инструменты статического анализа для обработки крупных и сложных проектов?
67. Как автоматизированные инструменты статического анализа могут влиять на производительность разработки?
68. Какие подходы к интеграции инструментов статического анализа существуют для использования в рабочих процессах разработки?
69. Какие критерии выбора инструментов статического анализа наиболее важны при интеграции их в проект?
70. Каковы основные шаги при внедрении нового инструмента статического анализа в процесс разработки?
71. Как инструменты статического анализа могут помочь в выявлении угроз безопасности кода?
72. Каким образом инструменты статического анализа обеспечивают высокую точность выявления ошибок и уязвимостей?
73. Какие средства визуализации и анализа предоставляются инструментами статического анализа для удобства работы разработчиков?
74. Какие аспекты кода могут быть охвачены инструментами статического анализа, помогая улучшить его структуру и стиль?
75. Какие вызовы и трудности могут возникнуть при внедрении инструментов статического анализа в большие и сложные проекты?
76. Как инструменты статического анализа могут помочь в предотвращении идентификации ложных срабатываний?
77. Каким образом автоматизированные инструменты статического анализа могут помочь в решении проблем, связанных с поддержкой кодовой базы?
78. Как статический анализ может быть внедрен в процесс разработки таким образом, чтобы не замедлять темпы работы разработчиков?
79. Как инструменты статического анализа могут помочь в обеспечении согласованности кода в рамках команды разработчиков?
80. Какие критерии оценки эффективности использования автоматизированных инструментов статического анализа в процессе разработки?

2.2. Оценочные средства для промежуточной аттестации

Вопросы для проведения аттестации

1. Какие основные преимущества предоставляет статический анализ кода на языке Python?
2. Какие особенности языка Python могут повлиять на процесс статического анализа?

3. Как статический анализ влияет на выявление потенциальных ошибок в коде на Python?
4. Какие инструменты статического анализа на Python чаще всего используются в разработке?
5. Как статический анализ помогает в поддержке чистоты и структурированности кода на Python?
6. Какие особенности синтаксиса Python могут влиять на точность статического анализа?
7. Как статический анализ на Python может помочь в выявлении уязвимостей безопасности?
8. Какие вызовы могут возникнуть при использовании статического анализа в больших проектах на Python?
9. Как статический анализ справляется с динамической типизацией Python?
10. Каким образом статический анализ может повлиять на процесс оптимизации кода на Python?
11. Какие аспекты архитектуры Python-проектов могут быть выявлены при статическом анализе?
12. Как статический анализ на Python взаимодействует с системами управления версиями, такими как Git?
13. Какие средства визуализации и анализа предоставляются инструментами статического анализа для Python?
14. Каким образом статический анализ Python помогает в соблюдении стандартов кодирования?
15. Как статический анализ влияет на процесс обнаружения и устранения "запахов кода" на Python?
16. Какие типы ошибок и уязвимостей можно выявить с использованием статического анализа кода на Python?
17. Какие вызовы могут возникнуть при интеграции инструментов статического анализа в процесс разработки на Python?
18. Какие аспекты безопасности могут быть улучшены с использованием статического анализа кода на Python?
19. Как статический анализ помогает в выявлении потенциальных проблем при миграции между версиями Python?
20. Какие дополнительные проверки стандартных библиотек Python могут быть внедрены с использованием статического анализа?
21. Какие проблемы могут возникнуть при статическом анализе проектов, использующих множество внешних библиотек и зависимостей на Python?
22. Как статический анализ воздействует на обнаружение и устранение проблем, связанных с асинхронным программированием в Python?
23. Каким образом инструменты статического анализа Python могут быть интегрированы в среды разработки, такие как PyCharm или VSCode?
24. Как статический анализ влияет на процесс обеспечения качества кода в командах разработки на Python?

25. Какие вызовы и трудности могут возникнуть при внедрении статического анализа в проекты на Python с большим объемом legacy-кода?
26. Что представляет собой Каталог уязвимостей по общей классификации (CWE)?
27. Какие основные цели преследует Каталог CWE в области безопасности программного обеспечения?
28. Как устроен классификационный подход CWE для организации уязвимостей?
29. Какие типы уязвимостей охватывает Каталог CWE?
30. Какова роль базы данных уязвимостей (БДУ) в контексте безопасности программного обеспечения?
31. Какие источники данных об уязвимостях включаются в базу данных уязвимостей?
32. Как устроена типизация уязвимостей в Каталоге CWE, и как она соотносится с БДУ?
33. Какие дополнительные атрибуты могут быть присвоены уязвимостям в Каталоге CWE?
34. Какие преимущества предоставляет использование CWE и БДУ при анализе безопасности кода?
35. Какова роль CWE и БДУ в процессе разработки безопасного программного обеспечения?
36. Как осуществляется обновление и поддержка Каталога CWE и базы данных уязвимостей?
37. Какие организации и стандарты активно вовлечены в развитие и поддержку CWE и БДУ?
38. Как Каталог CWE взаимодействует с другими стандартами и инструментами в области безопасности программного обеспечения?
39. Какие сложности могут возникнуть при использовании Каталога CWE и БДУ в процессе обеспечения безопасности ПО?
40. Какие рекомендации предоставляет Каталог CWE по устранению или смягчению уязвимостей?
41. Как БДУ классифицирует и организует информацию об уязвимостях для облегчения поиска и анализа?
42. Какой процесс добавления новых уязвимостей в Каталог CWE и БДУ?
43. Какие инструменты и сервисы могут использовать данные из Каталога CWE и БДУ для обеспечения безопасности программного обеспечения?
44. Как Каталог CWE способствует стандартизации терминологии и понимания уязвимостей в сообществе разработчиков?
45. Какие вызовы могут возникнуть при использовании данных из Каталога CWE и БДУ в процессе тестирования безопасности?
46. Какие меры принимаются для обеспечения конфиденциальности и актуальности информации в БДУ?
47. Как интеграция данных из Каталога CWE и БДУ может помочь в обучении и повышении осведомленности разработчиков о безопасности?

48. Какая роль уязвимостей в Каталоге CWE и БДУ в процессе анализа рисков безопасности программного обеспечения?
49. Как Каталог CWE и БДУ взаимодействуют с программами обучения и сертификацией в области безопасности программного обеспечения?
50. Какова перспектива развития Каталога CWE и БДУ в контексте динамично меняющейся обстановки?
51. Какие методы устранения дефектов в коде существуют на ранних этапах разработки?
52. Как использование стандартов кодирования может помочь в предотвращении дефектов в коде?
53. Какие техники и инструменты используются для статического анализа кода с целью выявления дефектов?
54. Как интеграция статического анализа в процесс разработки помогает в автоматическом выявлении дефектов?
55. Какие методы автоматизированного тестирования используются для выявления дефектов в коде?
56. Как процессы непрерывной интеграции и развертывания (CI/CD) влияют на устранение дефектов в коде?
57. Как ревью кода и пейр-программирование помогают в выявлении и устранении дефектов на ранних этапах разработки?
58. Как использование линтеров и статических анализаторов может улучшить процесс устранения дефектов в коде?
59. Как процессы код-рефакторинга могут помочь в улучшении структуры кода и устранении потенциальных дефектов?
60. Как тестирование безопасности влияет на устранение дефектов, связанных с потенциальными угрозами безопасности?
61. Как использование логирования и мониторинга может помочь в выявлении и устранении дефектов в продакшн-среде?
62. Как управление задачами и баг-трекинговыми системами влияет на процесс устранения дефектов?
63. Какова роль регрессионного тестирования в процессе устранения дефектов и предотвращения их повторения?
64. Как методы отладки (debugging) могут быть использованы для эффективного устранения дефектов в коде?
65. Какие методы и инструменты используются для анализа и устранения дефектов, связанных с производительностью приложения?
66. Как анализ журналов ошибок и исключений помогает в выявлении и устранении дефектов?
67. Какие практики использования тестовых данных и сценариев могут помочь в устранении дефектов в коде?
68. Как интеграция методов "черного ящика" тестирования влияет на устранение дефектов?
69. Какие вызовы могут возникнуть при устранении дефектов в больших и сложных проектах?
70. Как принятие решений об устранении дефектов влияет на приоритеты разработки и релизы?

71. Какие методы обучения и поддержки разработчиков существуют для повышения навыков устранения дефектов?
72. Как использование контрольных точек и аудитов кода может помочь в устранении дефектов в процессе разработки?
73. Как управление версиями кода влияет на процесс устранения дефектов и воспроизводимость ошибок?
74. Как процессы отладки и тестирования влияют на сроки устранения дефектов в коде?
75. Какие методы и метрики используются для оценки эффективности процесса устранения дефектов в коде?