Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 22.05.2024 17:01:23
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____ / Д.Г.Демидов /
«15» февраля 2024г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Анализ программного кода»

Направление подготовки/специальность
09.03.02 «Информационные системы и технологии»

Профиль/специализация
Информационные технологии в креативных индустриях

Квалификация
Бакалавр

Формы обучения
Очная

Москва, 2024 г.

Разработчик(и):
ст. преподаватель  / М.В. Алпатова /

Согласовано:
Заведующий кафедрой
«Информатики и информационных технологий»,
к.т.н.  / Е.В. Булатников/

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 5
3 Структура и содержание дисциплины 5
3.1 Виды учебной работы и трудоемкость 5
3.2 Тематический план изучения дисциплины 5
3.3 Содержание дисциплины 7
3.4 Тематика семинарских/практических и лабораторных занятий 8
3.5 Тематика курсовых проектов (курсовых работ) 8
4 Учебно-методическое и информационное обеспечение 8
4.1 Нормативные документы и ГОСТы 8
4.2 Основная литература 9
4.3 Дополнительная литература 9
4.4 Электронные образовательные ресурсы 9
4.5 Лицензионное и свободно распространяемое программное обеспечение 9
4.6 Современные профессиональные базы данных и информационные справочные системы 9
5 Материально-техническое обеспечение 9
6 Методические рекомендации 9
6.1 Методические рекомендации для преподавателя по организации обучения 10
6.2 Методические указания для обучающихся по освоению дисциплины 10
7 Фонд оценочных средств 10
7.1 Методы контроля и оценивания результатов обучения 10
7.2 Шкала и критерии оценивания результатов обучения 10
7.3 Оценочные средства 12

1 Цели, задачи и планируемые результаты обучения по дисциплине

Дисциплина «Анализ программного кода» направлена на формирование у студентов компетенций в области анализа, оценки и оптимизации программного кода. Целью является подготовка специалистов, способных применять современные методы и инструменты анализа кода для эффективного проектирования и разработки информационных систем среднего и крупного масштаба в соответствии с планируемыми результатами освоения основной профессиональной образовательной программы высшего образования.

В соответствии с задачами профессиональной деятельности, указанными в ФГОС ВО, дисциплина направлена на решение следующих задач:

- Освоение принципов и методов анализа программного кода, включая статический и динамический анализ, оптимизацию и рефакторинг, а также безопасность кода.
- Применение современных инструментов анализа кода для разработки и оптимизации информационных систем.
- Разработка навыков применения анализа кода в реальных проектах, включая определение уязвимостей и обеспечение качества ПО.

По завершении изучения дисциплины студенты должны обладать следующими компетенциями в соответствии с индикаторами достижения, установленными в ОПОП:

- Студенты знают основные принципы проектирования информационных систем, особенности их реализации на предприятиях среднего и крупного масштаба.
- Студенты умеют проводить концептуальное, функциональное и логическое проектирование информационных систем, используя современные методы анализа кода.
- Студенты обладают практическими навыками в области использования программного обеспечения для концептуального, функционального и логического проектирования информационных систем, умеют анализировать и оптимизировать программный код.

Обучение по дисциплине «Анализ программного кода» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ИОПК-2.1. знает современные информационные технологии и программные средства, основные виды и принципы работы информационных систем и информационных технологий; способы внедрения и интеграции современных информационных систем, способы оценки необходимости использования программных средств ИОПК-2.2. умеет использовать современные информационные технологии и программные средства, как в рамках отдельного предприятия, так и в рамках корпораций, государственных систем; внедрять и настраивать современные информационные системы, проводить интеграцию различных информационных систем и программных средств, оценивать необходимость использования программного средства для решения задач ИОПК-2.3. владеет навыками применения современных информационных технологий и

	программных средств, при решении задач в различных отраслях, внедрения и настройки современных информационных систем, оценки необходимости использования программных средств и информационных систем для решения задач
ПК-6. Способен предотвращать потери и повреждения данных	ИПК-6.1. Знает способы и методы резервного копирования и восстановления данных ИПК-6.2. Умеет производить резервное копирование и восстановление данных ИПК-6.3. Имеет навыки применения программного обеспечения для резервного копирования и восстановления данных

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к категории элективных дисциплин блока Б1 «Дисциплины (модули)».

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Базы данных;
- Объектно-ориентированное программирование;
- Тестирование программного обеспечения;
- Шаблоны проектирования;
- Производственная практика (проектно-технологическая);
- Производственная практика (преддипломная);
- Выполнение и защита выпускной квалификационной работы.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зачетные единицы, т.е. 108 академических часов.

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестр
			6
1	**Аудиторные занятия**	**54**	**54**
	В том числе:		
1.1	Лекции	18	18
1.2	Семинарские/практические занятия		
1.3	Лабораторные занятия	36	36
2	**Самостоятельная работа**	**54**	**54**
3	**Промежуточная аттестация**		
	Зачет/диф.зачет/экзамен	**зачет**	**зачет**
	Итого:	**108**	**108**

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Введение в анализ программного кода	8	2				6
2	Принципы проектирования информационных систем	8	2				6
3	Методы анализа кода	8	2				6
4	Инструменты анализа кода	14	2		6		6
5	Оптимизация и рефакторинг кода	14	2		6		6
6	Безопасность кода:	14	2		6		6
7	Тестирование и анализ кода	14	2		6		6
8	Метрики кода и качество ПО	14	2		6		6
9	Применение анализа кода в реальных проектах	14	2		6		6
Итого		**108**	**18**		**36**		**54**

3.3 Содержание дисциплины

3.3.1 Очная форма обучения

1. **Введение в анализ программного кода:** Обзор истории и значимости анализа кода, его роли в современных информационных системах. Отображение компетенций в контексте анализа кода.
2. **Принципы проектирования информационных систем**: Рассмотрение ключевых аспектов проектирования, включая концептуальное, функциональное и логическое проектирование. Особенности проектирования для систем среднего и крупного масштаба.
3. **Методы анализа кода**: Введение в статический и динамический анализ кода, их применение и ограничения.
4. **Инструменты анализа кода**: Обзор современных инструментов для статического и динамического анализа кода, их возможности и применение в практике.
5. **Оптимизация и рефакторинг кода**: Принципы и методы оптимизации кода, необходимость и методы рефакторинга, практические примеры.
6. **Безопасность кода**: Введение в принципы безопасного программирования, анализ уязвимостей кода, методы и инструменты для обнаружения уязвимостей.
7. **Тестирование и анализ кода**: Взаимосвязь между тестированием и анализом кода, методы тестирования, которые используют анализ кода.

8. **Метрики кода и качество ПО**: Введение в метрики кода, их значение для оценки качества кода, инструменты для сбора и анализа метрик.
9. **Применение анализа кода в реальных проектах**: Кейсы из практики, обзор типичных ошибок и проблем, решаемых с помощью анализа кода.

3.3.2 Заочная форма обучения

1. **Введение в анализ программного кода:** Обзор истории и значимости анализа кода, его роли в современных информационных системах. Отображение компетенций в контексте анализа кода.
2. **Принципы проектирования информационных систем**: Рассмотрение ключевых аспектов проектирования, включая концептуальное, функциональное и логическое проектирование. Особенности проектирования для систем среднего и крупного масштаба.
3. **Методы анализа кода**: Введение в статический и динамический анализ кода, их применение и ограничения.
4. **Инструменты анализа кода**: Обзор современных инструментов для статического и динамического анализа кода, их возможности и применение в практике.
5. **Оптимизация и рефакторинг кода**: Принципы и методы оптимизации кода, необходимость и методы рефакторинга, практические примеры.
6. **Безопасность кода**: Введение в принципы безопасного программирования, анализ уязвимостей кода, методы и инструменты для обнаружения уязвимостей.
7. **Тестирование и анализ кода**: Взаимосвязь между тестированием и анализом кода, методы тестирования, которые используют анализ кода.
8. **Метрики кода и качество ПО**: Введение в метрики кода, их значение для оценки качества кода, инструменты для сбора и анализа метрик.
9. **Применение анализа кода в реальных проектах**: Кейсы из практики, обзор типичных ошибок и проблем, решаемых с помощью анализа кода.

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1 Очная форма обучения

1. **Знакомство с инструментами статического анализа**: Работа с популярными инструментами статического анализа, анализ результатов.
2. **Динамический анализ программного кода**: Применение инструментов для динамического анализа, интерпретация результатов.
3. **Рефакторинг и оптимизация кода**: Практическая работа по определению участков кода, требующих рефакторинга, и выполнение необходимых изменений.
4. **Анализ безопасности кода**: Использование инструментов для обнаружения уязвимостей в коде, анализ результатов и разработка рекомендаций по устранению.
5. **Тестирование на основе анализа кода**: Создание тестовых сценариев на основе результатов анализа кода, применение автоматизированных инструментов тестирования.
6. **Анализ метрик кода**: Сбор и интерпретация метрик кода с использованием специализированных инструментов.

3.5 Тематика курсовых проектов (курсовых работ)

Курсовой проект дисциплиной не предусмотрен.

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный закон от 29 декабря 2012 года No 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);

2. Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.02 Информационные системы и технологии, утвержденный Приказом Министерства образования и науки РФ от 19 сентября 2017 г. N 929 "Об утверждении федерального... Редакция с изменениями N 1456 от 26.11.2020;

3. Приказ Министерства образования и науки РФ от 05 апреля 2017 г. No 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры.

4.2 Основная литература

1. Миронов, А. И. Тестирование и верификация программного обеспечения: Практикум : учебное пособие / А. И. Миронов, С. М. Трушин, А. А. Петренко. — Москва : РТУ МИРЭА, 2022. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/240095 (дата обращения: 27.03.2024). — Режим доступа: для авториз. пользователей.

2. Кузнецов, А. С. Системное программирование : учебное пособие / А. С. Кузнецов, И. А. Якимов, П. В. Пересунько. — Красноярск : СФУ, 2018. — 170 с. — ISBN 978-5-7638-3885-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/157574 (дата обращения: 27.03.2024). — Режим доступа: для авториз. пользователей.

3. Старолетов, С. М. Основы тестирования и верификации программного обеспечения / С. М. Старолетов. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 344 с. — ISBN 978-5-507-46773-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/319445 (дата обращения: 27.03.2024). — Режим доступа: для авториз. пользователей.

4.3 Дополнительная литература

1. Brian Chess, Jacob West: Secure Programming with Static Analysis. Addison-Wesley Professional, 2007.

2. Stéphane Eyskens: Software Architecture for Busy Developers. Packt Publishing, 2021

3. Martin Kleppmann: Designing Data-Intensive Applications. O'Reilly Media, Inc.б 2017

4.4 Электронные образовательные ресурсы

1. Анализ программного кода. LMS Московского политеха. ЭОР разрабатывается.

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. VS Code
2. SourceTree

4.6 Современные профессиональные базы данных и информационные справочные системы

1. ОП "Юрайт" https://urait.ru/
2. IPR Smart https://www.iprbookshop.ru/
3. ЭБС "Лань" https://e.lanbook.com/

5 Материально-техническое обеспечение

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется на зачете в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- Выполнение лабораторных работ
- Промежуточное тестирование (посредством изучения теоретических материалов в системе LMS)
- Итоговое тестирование

7.2 Шкала и критерии оценивания результатов обучения

Итоговая оценка по дисциплине рассчитывается как среднее взвешенное всех оценок в соответствующем курсе LMS Московского политеха с применением весовых коэффициентов, представленных ниже:

- Лабораторные работы → 0,8
- Итоговое тестирование → 0,05
- Ознакомление с теорией → 0,15

Оценка за каждую лабораторную работу выставляется исходя из фактического выполнения всех поставленных задач с учётом сроков исполнения: за каждую 1 неделю просрочки задания из оценки вычитается 10 баллов.

Для получения положительной экзаменационной оценки студенту необходимо набрать всего минимально 55 баллов по дисциплине и завершить итоговый тест с результатом не менее 55%.

Шкала оценивания	Диапазон баллов	Описание
Не зачтено	0-54	Не достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
Зачтено	55-69	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
	70-84	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.

	85-100	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

7.3 Оценочные средства

7.3.1 Вопросы к зачету

1. Что такое анализ программного кода и какова его цель в разработке ПО?
2. В чём разница между статическим и динамическим анализом кода?
3. Какие основные инструменты используются для статического анализа кода?
4. Какие задачи решаются с помощью динамического анализа?
5. Что такое рефакторинг кода и каковы его основные принципы?
6. Какие методики используются для оптимизации программного кода?
7. Какие основные типы уязвимостей могут быть выявлены при анализе безопасности кода?
8. Каковы основные принципы безопасного программирования?
9. Какие практические проблемы могут возникнуть при применении инструментов анализа кода в реальных проектах?
10. Как связаны процесс анализа кода и проектирования информационных систем?
11. В чём заключается значение исследования производительности кода?
12. Что такое технический долг и как он связан с анализом программного кода?
13. Какие задачи решает процесс автоматизации анализа кода?
14. Что такое код-ревью и какова его роль в процессе разработки?
15. Каковы преимущества и недостатки автоматического анализа кода по сравнению с ручным?
16. В чём заключается принцип "безопасности по умолчанию" и как он реализуется в коде?
17. Какие метрики качества кода чаще всего используются при его анализе?
18. Как анализ кода помогает в управлении жизненным циклом программного продукта?
19. Что такое контейнеризация и как она может повлиять на безопасность и производительность кода?
20. Какие современные тенденции и инновации наблюдаются в области анализа программного кода?