

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 02.12.2024 г.

Рабочая программа дисциплины

Методы профилировки программ

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Сопряженная разработка программного и аппаратного обеспечения

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.01.02 Методы профилировки программ относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-2: Способен к применению общенаучных базовых знаний математических и естественных наук, фундаментальной информатики и информационных технологий; применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии	ПК-2.1: Знает основы научно-исследовательской деятельности в области информационных технологий, имеет научные знания в теории информационных систем ПК-2.2: Умеет применять полученные знания в области фундаментальных научных основ теории информации и решать стандартные задачи в собственной научно-исследовательской деятельности ПК-2.3: Имеет практический опыт научно-исследовательской деятельности в области информационных технологий	ПК-2.1: Знать: Основные принципы и методы разработки программного кода. Синтаксис и семантику используемых языков программирования . ПК-2.2: Уметь: Разрабатывать корректный и эффективный программный код. Использовать стандартные библиотеки и фреймворки для разработки. ПК-2.3: Владеть: Навыками написания и документирования кода.	Тест Практическое задание Практическая задача	Зачёт: Контрольные вопросы

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	3
Часов по учебному плану	108
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	32

- занятия семинарского типа (практические занятия / лабораторные работы)	0
- КСР	1
самостоятельная работа	75
Промежуточная аттестация	0 Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
Ф	Ф	Ф	Ф	Ф	
1. Введение в профилирование	14	4	0	4	10
2. Инструменты статического анализа	14	4	0	4	10
3. Динамическое профилирование	18	6	0	6	12
4. Профилирование памяти	18	6	0	6	12
5. Профилирование многопоточных приложений	21	6	0	6	15
6. Оптимизация на основе профилирования	22	6	0	6	16
Аттестация	0				
КСР	1			1	
Итого	108	32	0	33	75

Содержание разделов и тем дисциплины

1. Введение в профилирование

Цели и задачи профилирования

Классификация методов профилирования

2. Инструменты статического анализа

Анализ исходного кода

Метрики сложности кода

3. Динамическое профилирование

Профилирование времени выполнения

Инструменты: gprof, VTune, perf

4. Профилирование памяти (6 часов)

Утечки памяти

Инструменты: Valgrind, Heaptrack

5. Профилирование многопоточных приложений (6 часа)

Анализ гонок данных

Инструменты: ThreadSanitizer, Intel Inspector

6. Оптимизация на основе профилирования (6 часа)

Методы оптимизации "горячих" участков кода

Кейсы из реальных проектов

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

а) основная литература:

1. Гамма Э. и др. "Приемы объектно-ориентированного проектирования". – М.: Питер, 2022.
2. Макконнелл С. "Совершенный код". – М.: БХВ-Петербург, 2021.

б) дополнительная литература:

1. Керниган Б., Пайк Р. "Практика программирования". – М.: Диалектика, 2020.
2. Блох Дж. "Java. Эффективное программирование". – М.: Лори, 2019.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-2:

1. Какой инструмент НЕ используется для статического анализа?
 - a) SonarQube
 - b) Valgrind
 - c) PVS-Studio
 - d) Cppcheck
2. Что измеряет метрика "цикломатическая сложность"?
 - a) Время выполнения
 - b) Количество путей выполнения
 - c) Объем используемой памяти
 - d) Число строк кода
3. Какой флаг компилятора нужен для gprof?
 - a) -O3
 - b) -pg
 - c) -debug
 - d) -profile
4. Что НЕ выявляет Valgrind?
 - a) Утечки памяти
 - b) Гонки данных

- c) Неинициализированные переменные
 - d) Оптимальность алгоритмов
5. Какой инструмент используется для профилирования многопоточных приложений?
- a) gprof
 - b) ThreadSanitizer
 - c) Heaptrack
 - d) perf
6. Что такое "hot spot" в профилировании?
- a) Ошибка компиляции
 - b) Участок кода с максимальным временем выполнения
 - c) Место утечки памяти
 - d) Неиспользуемая функция
7. Какой инструмент Intel используется для профилирования?
- a) VTune
 - b) Valgrind
 - c) gdb
 - d) strace
8. Что анализирует инструмент perf?
- a) Только время выполнения
 - b) Аппаратные события процессора
 - c) Только использование памяти
 - d) Только сетевую активность
9. Какой тип графиков НЕ используется в профилировании?
- a) Flame graph
 - b) Call graph
 - c) Pie chart
 - d) Heat map
10. Что такое "кэш-промах"?
- a) Ошибка в кэше процессора
 - b) Обращение к памяти, отсутствующей в кэше
 - c) Утечка в кэше
 - d) Переполнение кэша
11. Какой инструмент подходит для профилирования Python?
- a) cProfile
 - b) gprof
 - c) VTune
 - d) Valgrind
12. Что измеряет метрика "инструкций на цикл" (IPC)?
- a) Скорость сети
 - b) Эффективность использования процессора
 - c) Размер исполняемого файла
 - d) Количество потоков
13. Какой инструмент используется для профилирования Java?
- a) JProfiler
 - b) gprof

- c) perf
- d) strace

14. Что такое "sample-based profiling"?

- a) Профилирование по шаблонам
- b) Периодический сбор данных о выполнении
- c) Анализ только отдельных функций
- d) Статический анализ кода

15. Какой параметр НЕ влияет на точность профилирования?

- a) Частота семплирования
- b) Размер тестовых данных
- c) Цвет интерфейса инструмента
- d) Уровень оптимизации кода

Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
зачтено	как минимум 80% правильных ответов в тесте
не зачтено	менее 80% правильных ответов в тесте

5.1.2 Типовые задания (оценочное средство - Практическая задача) для оценки сформированности компетенции ПК-2:

1. Провести статический анализ заданного кода с помощью SonarQube.
2. Сравнить отчеты gprof и perf для одной программы.
3. Найти и исправить утечку памяти с помощью Valgrind.
4. Проанализировать "горячие" функции с помощью VTune.
5. Оптимизировать код на основе flame graph.
6. Диагностировать гонку данных в многопоточном приложении.
7. Построить call graph для заданной программы.
8. Провести профилирование сетевого приложения.
9. Сравнить производительность двух алгоритмов.
10. Интегрировать профилирование в CI-конвейер.

Критерии оценивания (оценочное средство - Практическая задача)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Программа и результаты работы представлены преподавателю в срок.

Оценка	Критерии оценивания
не зачтено	Выполнены не все лабораторные работы или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, программа работает некорректно, результаты работы не представлены преподавателю).

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и	Продемонстрирован творческий подход к решению нестандартных задач

	ответа		и недочетами	недочетами		недочетов	
--	--------	--	-----------------	------------	--	-----------	--

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-2

1. Определение и цели профилирования программ.
2. Отличия статического и динамического профилирования.
3. Основные метрики сложности кода.
4. Принципы работы инструмента gprof.
5. Как интерпретировать отчеты VTune?
6. Методы выявления утечек памяти.
7. Принцип работы Valgrind.
8. Типы ошибок, выявляемых ThreadSanitizer.

9. Оптимизация кода на основе профилирования.
10. Инструменты для профилирования GPU.
11. Анализ производительности встроенных функций.
12. Профилирование сетевых приложений.
13. Методы снижения накладных расходов при профилировании.
14. Анализ кэш-промахов.
15. Профилирование JIT-компилируемых языков.
16. Интеграция профилирования в CI/CD.
17. Этические аспекты профилирования.
18. Сравнение инструментов для разных языков программирования.
19. Профилирование распределенных систем.
20. Анализ производительности алгоритмов.
21. Методы профилирования в облачных средах.
22. Визуализация результатов профилирования.
23. Профилирование на уровне инструкций процессора.
24. Анализ энергопотребления при профилировании.
25. Современные тенденции в инструментах профилирования.

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Студент ответил на большую часть вопросов, возможно с незначительными недочетами.
не зачтено	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Гамма Э.,

Хелм Р., Джонсон Р., Влиссидес Д. - Москва : ДМК-пресс, ., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=636387&idb=0>.

Дополнительная литература:

1. Майерс С. Эффективное использование C++. 55 верных способов улучшить структуру и код ваших программ : практическое руководство / Майерс С. - Москва : ДМК-пресс, 2023. - 302 с. - ISBN 978-5-89818-324-0., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=878885&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Операционные системы семейства Microsoft Windows, лицензия по подписке Microsoft Imagine.
2. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
3. Среда разработки семейства Microsoft Visual Studio, лицензия по подписке Microsoft Imagine.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Золотых Николай Юрьевич, доктор физико-математических наук, доцент.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 02.12.2024, протокол № 5.