

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Факультет информационных технологий

СОГЛАСОВАНО

Декан ФИТ НГУ



М.М. Лаврентьев

«03» июля 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Оптимизация производительности Java-программ

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Программная инженерия и компьютерные науки

Форма обучения: очная

Год обучения: 4, семестр: 7

№	Вид деятельности	Семестр
		7
1	Лекции, час.	16
2	Практические занятия, час.	32
3	Лабораторные занятия, час.	
4	Занятий в контактной форме без учета промежуточной аттестации, час, из них	50
5	в электронной форме, час.	
6	из них аудиторных занятий, час.	48
7	из них в активной и интерактивной форме, час.	48
8	консультаций, час.	2
9	Самостоятельная работа, час.	92
10	в том числе на выполнение письменных работ, час	50
11	Форма аттестации (экзамен, зачет, дифференцированный зачет), час	Э 2
12	Всего зачетных единиц ¹	4

Новосибирск 2019

¹ С учетом выделенных часов на промежуточную аттестацию

Рабочая программа дисциплины составлена на основании федерального государственного образовательного стандарта (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА.

Федеральный государственный образовательный стандарт (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА введен в действие приказом Минобрнауки от 19.09.2017 № 929.

Место дисциплины в структуре учебного плана: Блок 1 Дисциплины (модули); часть, формируемая участниками образовательных отношений, дисциплина по выбору

Рабочая программа дисциплины утверждена решением Ученого совета факультета информационных технологий от 02.07.2019, протокол № 75.

Программу разработал:

ст.преподаватель кафедры систем информатики ФИТ



А.В.Адаманский

Заведующий кафедрой систем информатики ФИТ,
доктор физико-математических наук



М.М. Лаврентьев

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,
кандидат технических наук



А.А. Романенко

Аннотация к рабочей программе дисциплины «Оптимизация производительности Java-программ»

Дисциплина «Оптимизация производительности Java-программ» реализуется в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, направленность (профиль): ПРОГРАММНАЯ ИНЖЕНЕРИЯ И КОМПЬЮТЕРНЫЕ НАУКИ по очной форме обучения на русском языке.

Место в образовательной программе: Дисциплина «Оптимизация производительности Java-программ» развивает знания, умения и навыки, сформированные у обучающихся по результатам изучения следующих дисциплин: «Объектно-ориентированное программирование».

Дисциплина «Оптимизация производительности Java-программ» реализуется в 7 семестре в рамках части, формируемой участниками образовательных отношений, дисциплин (модулей) Блока 1 и является дисциплиной по выбору.

Дисциплина «Оптимизация производительности Java-программ» направлена на формирование компетенций:

Способен разрабатывать компоненты системных программных продуктов (ПКС-2), в части следующих индикаторов достижения компетенции:

ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня

ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение

Перечень основных разделов дисциплины:

При освоении дисциплины студенты выполняют следующие виды учебной работы: лекции, практические занятия, консультации, самостоятельная работа.

Содержание дисциплины охватывает круг вопросов, связанных с разработкой программных систем на Java платформе. Излагает методологию оценки эффективности работы программных систем. Описывает факторы, влияющие на производительность программных систем, и механизмы управления этими факторами.

Самостоятельная работа включает: подготовку к практическим занятиям, выполнение заданий по разделам дисциплины, подготовку к текущему контролю успеваемости в форме заданий, устных опросов и контрольных работ и промежуточному контролю в форме экзамена.

Общий объем дисциплины – 4 зачетных единиц (144 часа).

Правила аттестации по дисциплине. Текущий контроль по дисциплине «Оптимизация производительности Java-программ» осуществляется на практических занятиях форме портфолио (задания, устные опросы и контрольные работы по основным разделам дисциплины, по результатам которых выставляется оценка по пятибалльной шкале).

Промежуточная аттестация по дисциплине «Оптимизация производительности Java-программ» проводится по завершению каждого периода ее освоения (семестра). Промежуточная аттестация по дисциплине проводится в форме экзамена.

Положительная оценка по дисциплине выставляется только при наличии всех сданных заданий и положительным оценкам за контрольные работы

Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно», означают успешное прохождение промежуточной аттестации.

Учебно-методическое обеспечение дисциплины.

Учебно-методический комплекс по дисциплине «Оптимизация производительности Java-программ» в электронной информационно-образовательной среде НГУ:

- Лекционные слайды <https://adamansky.bitbucket.io/slides/index.html>

1. Внешние требования к дисциплине

Таблица 1.1

Компетенция ПКС-2 Способен разрабатывать компоненты системных программных продуктов, в части следующих индикаторов достижения компетенции:	
ПКС-2.1	Владеть: навыками разработки программ на языках высокого уровня
ПКС-2.7	Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение

2. Требования к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий		
	Лекции	Практики / семинары	Самостоятел ьная работа
ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня			
1. Знать архитектуру виртуальной Java машины (JVM). Процесс создания программ для JVM. Этапы выполнения программ в JVM.	+	+	+
ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение			
2. Уметь использовать утилиты командной строки для сборки, выполнения и отладки программ. Использовать IDE средства NetBeans и Eclipse для создания и отладки Java программ, использовать базовый спектр инструментальных средств, необходимых для создания, оптимизации и диагностики Java программ.	+	+	+

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час. (входит в общее кол-во часов)	Часы	Ссылки на результаты обучения
Семестр: 7			
Архитектура JVM	3	3	1, 2
Структура байткода класс файлов	3	3	1, 2
Обзор алгоритмов сборки мусора	2	2	1, 2
Методы сборки мусора в JVM	2	2	1, 2
Инструментальные средства мониторинга производительности программ JVM	4	4	1, 2
Лучшие практики оптимизации программного кода	2	2	1, 2
Итого:	16	16	

Таблица 3.2

Темы практических занятий	Активные формы, час. (входит в общее кол-во часов)	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 7				
Архитектура JVM	6	6	1, 2	Разбор теоретической темы, представленной на лекции, решение задач
Структура байткода класс файлов	6	6	1, 2	Разбор теоретической темы, представленной на лекции, решение задач
Обзор алгоритмов сборки мусора	4	4	1, 2	Разбор теоретической темы, представленной на лекции, решение задач
Методы сборки мусора в JVM	4	4	1, 2	Разбор теоретической темы, представленной на лекции, решение задач
Инструментальные средства мониторинга производительности программ JVM Диагностика утечек памяти в программах.	8	8	1, 2	Разбор теоретической темы, представленной на лекции, решение задач
Лучшие практики оптимизации программного кода	4	4	1, 2	Разбор теоретической темы, представленной на лекции, решение задач
Итого:	32	32		

4. Самостоятельная работа студентов

Таблица 4.1

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 7				
1	изучение разделов дисциплины по учебной литературе, в том числе вопросов, не освещаемых на лекциях	1, 2	18	
	Изучение предлагаемых теоретических разделов в соответствии с настоящей Программой.			
2	Подготовка к практическим занятиям, к текущему контролю знаний	1, 2	50	
	Выполнение заданий, подготовка к контрольным работам			
3	Подготовка к экзамену	1, 2	24	2
	Повторение теоретического материала по вопросам, совпадающим с темами лекций			
	Итого		92	2

5. Образовательные технологии

В ходе реализации учебного процесса по дисциплине проводятся лекционные и практические занятия. Темы, рассматриваемые на лекциях и изучаемые самостоятельно, закрепляются на практических занятиях, по вопросам, вызывающим затруднения, проводятся консультации.

В ходе реализации учебного процесса по дисциплине применяются такие формы проведения практических занятий, как дискуссии, обсуждение и защита результатов работы, а также применяются следующие интерактивные формы обучения (таблица 5.1).

Таблица 5.1

1	Лекция в форме дискуссии	ПКС-2.1, ПКС-2.7
Формируемые умения: Знать архитектуру виртуальной Java машины. Процесс сборки и выполнения программы в JVM Уметь использовать утилиты командной строки для сборки, выполнения и отладки программ. Использовать IDE средства Nebeans и Eclipse для создания и отладки Java программ, использовать базовый спектр инструментальных средств, необходимых для создания, оптимизации и диагностики Java программ.		
Краткое описание применения: Представляется теория, проблематика вопросов, связанных с оптимизацией производительности Java программ, обсуждаются идеи и способы решения задач, рекомендованных для практических занятий		
2	Портфолио	ПКС-2.1, ПКС-2.7
Формируемые умения: Знать архитектуру виртуальной Java машины. Процесс сборки и выполнения программы в JVM Уметь использовать утилиты командной строки для сборки, выполнения и отладки программ. Использовать IDE средства Nebeans и Eclipse для создания и отладки Java программ, использовать базовый спектр инструментальных средств, необходимых для создания, оптимизации и диагностики Java программ.		
Краткое описание применения: бакалавры ведут портфолио (оценки за задания, контрольные работы), которое является основой для проведения аттестации по дисциплине		

Для организации и контроля самостоятельной работы студентов, а также проведения консультаций применяются информационно-коммуникационные технологии (таблица 5.2).

Таблица 5.2

Информирование	Адрес почты adamansky@gmail.com сообщается студентам на первом занятии.
Консультирование	Адрес почты adamansky@gmail.com сообщается студентам на первом занятии.
Контроль	Адрес почты adamansky@gmail.com сообщается студентам на первом занятии.
Размещение учебных материалов	https://adamansky.bitbucket.io/slides/index.html

6. Правила аттестации студентов по учебной дисциплине

По дисциплине «Оптимизация производительности Java-программ» проводится текущая и промежуточная аттестация (итоговая по дисциплине).

Правила аттестации по дисциплине. Текущий контроль по дисциплине «Оптимизация производительности Java-программ» осуществляется на практических занятиях в форме портфолио, состоящего из заданий, устных опросов и контрольных работ по основным разделам дисциплины, по результатам которых выставляется оценка по пятибалльной шкале.

Пример контрольных опросов:

1. Кратко опишите принцип действия стековых виртуальных машин.
2. Опишите общий алгоритм действий при оптимизации производительности программных систем
3. Сценарии использования атомарных compare-and-swap (CAS) операций.

Пример вопросов для контрольной работы

1. Проиллюстрируйте в коде spin блокировку (крит. секцию), в каких случаях spin блокировки эффективны?
2. Критическая секция, кратко укажите (как java код) возможные способы определения критической секции в программе.
3. В какие операции байткода стековой виртуальной JVM компилятор преобразует сложение двух строк: `String s = "foo" + "bar";`

Пример задания:

Работа 1.

Сделайте анализ байткода следующих программ:

```
package nsu.fit.javaperf.lab2;
```

```
public class P1 {
    public static void main(String[] args) {
        String s = "";
        for (int i = 0; i < 10000; ++i) {
            s += " " + String.valueOf(i);
        }
        System.out.println(s.length());
    }
}
```

и

```
package nsu.fit.javaperf.lab2;
```

```
public class P2 {
    public static void main(String[] args) {
        StringBuilder s = new StringBuilder();
        for (int i = 0; i < 10000; ++i) {
            s.append(" ").append(String.valueOf(i));
        }
        System.out.println(s.length());
    }
}
```

на основе проведенного анализа байткода программ оцените время сравнительное время выполнения и использование памяти этих программ.

Работа 2.

Для программы:

```
package nsu.fit.javaperf;
```

```
public class TransactionProcessor {
```

```

public void processTransaction(int txNum) throws Exception{
    System.err.println("Processing tx: " + txNum);
    int sleep = (int) (Math.random() * 1000);
    Thread.sleep(sleep);
    System.err.println(String.format("tx: %d completed", txNum));
}

```

```

public static void main(String[] args) throws Exception{
    TransactionProcessor tp = new TransactionProcessor();
    int tx = 0;
    tp.processTransaction(++tx);
    tp.processTransaction(++tx);
    tp.processTransaction(++tx);
}
}

```

Используя технологию агентов JVM и трансформацию класс файлов получите ответы на следующие вопросы:

1. Количество классов загруженных JVM в течении времени выполнения программы.
2. Измерьте время выполнения каждого метода в каждом классе находящимся в пакете nsu.fit.javaperf.

Положительная оценка по дисциплине выставляется только при наличии всех сданных заданий и положительным оценкам за контрольные работы

Промежуточная аттестация по дисциплине «Оптимизация производительности Java-программ» проводится по завершению каждого периода ее освоения (семестра). Промежуточная аттестация по дисциплине проводится в форме экзамена.

Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно», означают успешное прохождение промежуточной аттестации.

В таблице 6.1 представлено соответствие форм аттестации заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.1

Коды компетенций ФГОС	Результаты обучения	Формы аттестации	
		семестр 7	
		портфолио	Экзамен
ПКС-2	ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня	+	+

ПКС-2	ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение	+	+
-------	---	---	---

Требования к структуре и содержанию портфолио, оценочные средства, а также критерии оценки сформированности компетенций и освоения дисциплины в целом, представлены в Фонде оценочных средств, являющемся приложением 1 к настоящей рабочей программе дисциплины.

7. Литература

1. Апанасевич, С. А. Структуры и алгоритмы обработки данных. Линейные структуры : учебное пособие / С. А. Апанасевич. — Санкт-Петербург : Лань, 2019. — 136 с. — ISBN 978-5-8114-3366-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/113934>
2. Городняя, Л. В. Парадигма программирования : учебное пособие / Л. В. Городняя. — Санкт-Петербург : Лань, 2019. — 232 с. — ISBN 978-5-8114-3565-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/>

Интернет-ресурсы

Таблица 7.1

№ п/п	Наименование Интернет-ресурса	Краткое описание
1.	http://preshing.com/20120612/an-introduction-to-lock-free-programming/	An Introduction to Lock-Free Programming
2.	http://www.oracle.com/technetwork/java/index.htm	официальный сайт для разработчиков Java
3.	http://blog.ragozin.info/2011/06/understanding-gc-pauses-in-jvm-hotspots.html	Исследование задержек сборщиков мусора
4.	http://www.cubrid.org/blog/dev-platform/understanding-java-garbage-collection/	Understanding Java Garbage Collection
5.	http://docs.oracle.com/javase/7/docs/technotes/guides/management/toc.html	Java SE Monitoring and Management Guide
6.	http://docs.oracle.com/javase/specs/jvms/se8/html/ http://arhipov.blogspot.ru/2011/01/java-bytecode-fundamentals.html	Java bytecode fundamentals (Основы байткода платформы Java) Спецификация виртуальной Java машины
7.	http://blog.ragozin.info/2011/12/garbage-collection-in-hotspot-jvm.html	Обзор методов сборки мусора в JVM
8.	http://www.brpreiss.com/books/opus5/html/book.html	Data Structures and Algorithms with Object-Oriented Design Patterns in Java

8. Учебно-методическое и программное обеспечение дисциплины

8.1. Учебно-методическое обеспечение

Учебно-методический комплекс по дисциплине «Оптимизация производительности Java-программ» в электронной информационно-образовательной среде НГУ:

- Лекционные слайды <https://adamansky.bitbucket.io/slides/index.html>

8.2. Программное обеспечение

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows и MS Office.

Перечень специализированного программного обеспечения для изучения дисциплины представлен в таблице 8.1.

Специализированное программное обеспечение

Таблица 8.1

№	Наименование ПО	Назначение
1	Eclipse 2019	Среда разработки приложений
2	NetBeans IDE 8.2	Среда разработки приложений

9. Профессиональные базы данных и информационные справочные системы

1. Полнотекстовые журналы Springer Journals за 1997-2015 г., электронные книги (2005-2016 гг.), коллекция научных биомедицинских и биологических протоколов SpringerProtocols, коллекция научных материалов в области физических наук и инжиниринга SpringerMaterials, реферативная БД по чистой и прикладной математике zbMATH.

2. Электронная библиотека диссертаций Российской государственной библиотеки (ЭБД РГБ)

3. Электронные ресурсы Web of Science Core Collection (Thomson Reuters Scientific LLC.), Journal Citation Reports + ESI

4. БД Scopus (Elsevier)

10. Материально-техническое обеспечение

Таблица 10.1

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Для проведения лекционных занятий
2	Компьютерный класс (с выходом в Internet)	Для проведения практических занятий и организации самостоятельной работы

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Факультет информационных технологий

СОГЛАСОВАНО

Декан ФИТ НГУ



М.М. Лаврентьев

«03» июля 2019 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
по дисциплине Оптимизация производительности Java-программ**

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Программная инженерия и компьютерные науки

Квалификация: бакалавр

Форма обучения: очная

Год обучения: 4, семестр 7

Форма аттестации	Семестр
Экзамен	7

Новосибирск 2019

Фонд оценочных средств промежуточной аттестации по дисциплине является **Приложением 1** к рабочей программе дисциплины «Оптимизация производительности Java-программ», реализуемой в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 Информатика и вычислительная техника, направленность (профиль): Программная инженерия и компьютерные науки.

Фонд оценочных средств промежуточной аттестации по дисциплине утвержден решением ученого совета факультета информационных технологий, протокол № 75 от 02.07.2019.

Разработчики:

ст.преподаватель кафедры систем информатики ФИТ

А.В.Адаманский

Заведующий кафедрой систем информатики ФИТ,
доктор физико-математических наук

М.М. Лаврентьев

Ответственный за образовательную программу:
доцент кафедры систем информатики ФИТ,
кандидат технических наук

А.А. Романенко

1. Содержание и порядок проведения промежуточной аттестации по дисциплине

1.1. Общая характеристика содержания промежуточной аттестации

Промежуточная аттестация по дисциплине «Оптимизация производительности Java-программ» проводится по завершению периода освоения образовательной программы (семестра) для оценки сформированности компетенций в части следующих индикаторов достижения компетенции (таблица П1.1).

Таблица П1.1

Код	Компетенции, формируемые в рамках дисциплины «Оптимизация производительности Java-программ»	Семестр 7	
		Портфолио	Экзамен
ПКС-2 Способен разрабатывать компоненты системных программных продуктов			
ПКС-2.1	Владеть: навыками разработки программ на языках высокого уровня	+	+
ПКС-2.7	Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение	+	+

1.2. Порядок проведения промежуточной аттестации по дисциплине

Промежуточная аттестация проводится в форме экзамена и включает 2 этапа: портфолио и экзамен. Необходимым условием для прохождения промежуточной аттестации является оценка «зачтено» по результатам выполненного портфолио. Для оценивания портфолио студенту необходимо сдать все работы, входящие в структуру портфолио.

Экзамен проводится в устной форме, в аудитории, студентам разрешено пользоваться бумагой для записей и авторучкой. Справочной, учебной и другой литературой пользоваться не разрешается. Использование электронных устройств (телефоны, любые виды компьютеров, т.д.) запрещено.

2. Требования к структуре и содержанию фонда оценочных средств промежуточной аттестации по дисциплине

Перечень оценочных средств, применяемых на каждом этапе проведения промежуточной аттестации по дисциплине, представлен в таблице П1.2.

Таблица П1.2

№	Наименование	Краткая характеристика оценочного	Представление
---	--------------	-----------------------------------	---------------

п/п	оценочного средства	средства	оценочного средства в фонде
Семестр 7 1 этап- портфолио			
1	Портфолио	Целевая подборка работ студента, раскрывающая его индивидуальные образовательные достижения в одной или нескольких учебных дисциплинах.	Требования к структуре и содержанию портфолио
Этап 2 – Экзамен			
2	Билет для экзамена	Комплекс вопросов	Список теоретических вопросов

2.1 Требования к структуре и содержанию оценочных средств аттестации

2.1.1 По дисциплине «Оптимизация производительности Java-программ» проводится текущая и промежуточная аттестация (итоговая по дисциплине).

Правила аттестации по дисциплине. Текущий контроль по дисциплине «Оптимизация производительности Java-программ» осуществляется на практических занятиях в форме портфолио

Состав портфолио:

задания, устные опросы и контрольные работы по основным разделам дисциплины, по результатам которых выставляется оценка по пятибалльной шкале.

Пример контрольных опросов:

1. Кратко опишите принцип действия стековых виртуальных машин.
2. Опишите общий алгоритм действий при оптимизации производительности программных систем
3. Сценарии использования атомарных compare-and-swap (CAS) операций.

Пример вопросов для контрольной работы

- Проиллюстрируйте в коде spin блокировку (крит. секцию), в каких случаях spin блокировки эффективны?
- Критическая секция, кратко укажите (как java код) возможные способы определения критической секции в программе.
- В какие операции байткода стековой виртуальной JVM компилятор преобразует сложение двух строк: `String s = "foo" + "bar";`

Пример задания:

Работа 1.

Сделайте анализ байткода следующих программ:

```
package nsu.fit.javaperf.lab2;
```

```

public class P1 {
    public static void main(String[] args) {
        String s = "";
        for (int i = 0; i < 10000; ++i) {
            s += " " + String.valueOf(i);
        }
        System.out.println(s.length());
    }
}

```

и

```

package nsu.fit.javaperf.lab2;

```

```

public class P2 {
    public static void main(String[] args) {
        StringBuilder s = new StringBuilder();
        for (int i = 0; i < 10000; ++i) {
            s.append(" ").append(String.valueOf(i));
        }
        System.out.println(s.length());
    }
}

```

на основе проведенного анализа байткода программ оцените время сравнительное время выполнения и использование памяти этих программ.

Работа 2.

Для программы:

```

package nsu.fit.javaperf;

```

```

public class TransactionProcessor {

    public void processTransaction(int txNum) throws Exception{

        System.err.println("Processing tx: " + txNum);

        int sleep = (int) (Math.random() * 1000);

        Thread.sleep(sleep);

        System.err.println(String.format("tx: %d completed", txNum));

    }
}

```

```
public static void main(String[] args) throws Exception{  
  
    TransactionProcessor tp = new TransactionProcessor();  
  
    int tx = 0;  
  
    tp.processTransaction(++tx);  
  
    tp.processTransaction(++tx);  
  
    tp.processTransaction(++tx);  
  
    } }
```

Используя java agent технологию и трансформацию класс файлов ответьте на следующие вопросы:

1. Количество классов загруженных JVM в течении времени выполнения программы.
2. Измерьте время выполнения каждого метода в каждом классе находящимся в пакете nsu.fit.javaperf.

Положительная оценка по дисциплине выставляется только при наличии всех сданных заданий и положительным оценкам за контрольные работы

Промежуточная аттестация по дисциплине «Оптимизация производительности Java-программ» проводится по завершению периода ее освоения (семестра). Промежуточная аттестация по дисциплине проводится в форме экзамена.

Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно», означают успешное прохождение промежуточной аттестации.

2.2.2 Форма и перечень вопросов билета для экзамена 7 семестра

Форма экзаменационного билета

Таблица П1.3

<p>Новосибирский государственный университет</p> <p>Экзамен</p> <hr/> <p>Оптимизация производительности Java-программ</p> <p style="font-size: small; text-align: center;">наименование дисциплины</p> <hr/> <p>09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА</p> <p style="text-align: center;">Программная инженерия и компьютерные науки</p> <p style="font-size: small; text-align: center;">наименование образовательной программы</p> <hr/> <p style="text-align: center;">ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №</p> <p>1. Вопрос из категории 1</p> <p>2. Вопрос из категории 2</p> <p>Составитель</p> <p>_____</p> <p style="text-align: center; font-size: small;">(подпись)</p> <p>Ответственный за образовательную программу</p> <p style="text-align: right; padding-right: 50px;">А.А. Романенко</p> <p>_____</p> <p style="text-align: center; font-size: small;">(подпись)</p> <p>« ____ » _____ 20 ____ г.</p>

Перечень вопросов экзамена, структурированный по категориям, представлен в таблице П1.4

Таблица П1.4

Семестр 7	Формулировка вопроса
Категория 1 (ПКС-2.1)	Принцип действия стековых вычислительных машин.
	Байт-код JVM, структура .class файлов.
	Инструментальные средства анализа и модификации байт-кода.
	Способы вызова методов в байт-коде. Дескрипторы типов и методов в байт-коде (bytecode descriptors).
	Динамические прокси-классы.
	JIT компилирование кода в JVM, основные шаблоны оптимизации нативного JIT кода в JVM
	Intrinsic (интринсик) функций в виртуальной машине HotSpot
	Общая архитектура JVM, структура памяти JVM.
	Стандартные опции командной строки JVM, управление максимальным/минимальным размером кучи в JVM.

	Получение информации о параметрах работы JVM.
	Понятие «сборщика мусора», характерные достоинства и недостатки GC систем.
	Понятия «динамическая куча», «стек потока». Статическая, динамическая стековая память.
	Граф объектов сборщика мусора, корни графов объектов (GC Roots).
	Mark-and-sweep алгоритм сборки мусора, принцип работы, достоинства, недостатки.
	Tri-color marking алгоритм сборки мусора, принцип работы, достоинства, недостатки.
	Compare-and-swap алгоритм сборки мусора, принцип работы, достоинства, недостатки.
	Сборка мусора основанная на подсчете ссылок на объекты, достоинства, недостатки.
	Concurrent mark-and-sweep алгоритм сборки мусора, принцип работы, достоинства, недостатки.
	Сборка мусора основанная на поколениях объектов. (Generational GC). Связь с классами памяти сборщика мусора JVM.
	«Остановка мира» (STW) фаза сборки мусора.
	Управление Off-heap памятью, сценарии использования.
	Диагностика работы сборщика мусора.
Категория 2 (ПКС-2.7)	Общий алгоритм действий при оптимизации производительности программных систем.
	Взаимодействие JVM с сервисами операционной системы, нативные расширения
	Средства командной строки для диагностики JVM и приложений.
	Средства диагностики мертвых блокировок и потоков выполнения (threads).
	Графические средства диагностики JVM и приложений.
	Диагностика утечек памяти, инспекция heap памяти.
	Диагностика использования CPU java программой. Методы измерения горячих точек JVM процесса: CPU Sampling и CPU Times.
	Предварительная оптимизация (premature optimization). Принцип парето.
	Оптимизация GUI приложений, отличие от подходов к оптимизации не интерактивных приложений.
	Основные источники проблем с производительностью программных систем, их относительный вес.
	Исключения в Java. Механизм обработки исключительных ситуаций, накладные расходы обработки исключений, способы оптимизации обработки исключений.
	Read-Write блокировки.
	Атомарная compare-and-swap (CAS) операция, сценарии использования.

Набор вопросов формируется и утверждается в установленном порядке в начале учебного года при наличии контингента обучающихся, завершающих освоение дисциплины «Оптимизация производительности Java-программ» в текущем учебном году.

3. Критерии оценки сформированности компетенций в рамках промежуточной аттестации по дисциплине

Таблица П1.5

Шифр компетенций	Структурные элементы оценочных средств	Показатель сформированности	Не сформирован	Пороговый уровень	Базовый уровень	Продвинутый уровень
ПКС-2	Портфолио (этап 1), Экзамен (этап 2)	ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня	Не знает архитектуру виртуальной Java машины, процесс сборки и выполнения программы в JVM	Имеет фрагментарные знания набора инструментов средств диагностики Java программ.	Имеет базовые знания набора инструментов диагностики Java программ.	Имеет твердые знания набора инструментальных средств диагностики Java программ.
ПКС-2	Портфолио (этап 1), Экзамен (этап 2)	ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, выработать и обосновать архитектурное решение	Не умеет использовать утилиты командной строки для сборки, выполнения и отладки программ. не использовать IDE средства Nebeans и Eclipse для создания и	Умеет использовать базовые диагностические инструменты командной строки для отладки JVM и IDE для создания Java программ	Умеет использовать базовые диагностические инструменты командной строки для отладки JVM и IDE для создания Java и отладки программ.	Умеет реализовать архитектурно сложные Java программы. Подробно объяснить принципы работы JVM и применять весь спектр средств диагностики Java

			отладки Java про- грамм, ис- пользовать базовый спектр ин- струмен- тальных средств.			
--	--	--	--	--	--	--

4. Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

Результаты промежуточной аттестации в 7 семестре определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Положительная оценка по дисциплине выставляется только при наличии положительной оценки за портфолио.

В 7 семестре результаты промежуточной аттестации определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

Оценка «неудовлетворительно» выставляется, если хотя бы одна компетенция не сформирована.

