

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор физтех-школы  
прикладной математики и  
информатики**

**А.М. Райгородский**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Оптимизация C++ приложений
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Физтех-школа Прикладной Математики и Информатики кафедра корпоративных информационных систем
<b>курс:</b>	4
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Всего часов: 90, всего зач. ед.: 2

Количество контрольных работ, заданий: 1

Программу составил: Н.Ю. Пустовойтов, старший преподаватель

Программа обсуждена на заседании кафедры корпоративных информационных систем 20.02.2020

## Аннотация

Дисциплина “Оптимизация C++ приложений” направлена на изучение способов оптимизации кода приложений на C++ для создания высокопроизводительных приложений. По ходу освоения программы курса студенты изучат основы построения компиляторов, особенности архитектуры процессов, способы повышения производительности приложений на C++. Помимо теоретических знаний студенты будут получать практические задания, которые помогут практиковаться в изученных способах повышения производительности приложений.

В дисциплине рассматриваются следующие темы:

- возможности компилятора по оптимизации;
- методика оптимизаций;
- общие стратегии оптимизаций;
- оптимизация на уровне написания кода;
- особенности архитектур с точки зрения оптимизаций;
- проектирование с учетом оптимизации;
- устройство компилятора;
- финальная оптимизация.

Заключительным этапом всего курса является дифференцированный зачет, целью которого является проверка знаний студентов по теории и выявление практических навыков, полученных при выполнении практических заданий.

## 1. Цели и задачи

### Цель дисциплины

Изучить способы оптимизации кода приложений на C++ для создания высокопроизводительных приложений.

### Задачи дисциплины

- Изучить основы построения компиляторов;
- изучить особенности архитектуры процессоров, дающие возможности для оптимизации;
- изучить способы повышения производительности приложений на C++;
- научиться применять на практике изученные способы повышения производительности приложений.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценить качество разработанной модели

новые научные результаты

ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- Общее устройство парсеров регулярных, контекстно-свободных и контекстно-зависимых грамматик;
- общее устройство компиляторов языка C++;
- общие стратегии оптимизации;
- методику оптимизации;
- возможности компилятора по оптимизации.

уметь:

- Осуществлять и обосновывать выбор оптимальной платформы;
- осуществлять проектирование архитектуры с учетом оптимизации производительности;
- осуществлять оптимизацию производительности при написании кода;
- осуществлять финальную оптимизацию.

владеть:

- Средствами профилирования выполнения программы;
- ключами компиляции.

### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

#### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Возможности компилятора по оптимизации.	4	4		
2	Методика оптимизаций.	4	4		
3	Общие стратегии оптимизаций.	4	4		
4	Оптимизация на уровне написания кода.	4	4		
5	Особенности архитектур с точки зрения оптимизаций.	4	4		
6	Проектирование с учетом оптимизации.	4	4		
7	Устройство компилятора.	4	4		
8	Финальная оптимизация.	2	2		30
Итого часов		30	30		30
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

#### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 8 (Весенний)

## 1. Возможности компилятора по оптимизации.

Мелкозернистые оптимизации. Сбор общих подвыражений, удаление избыточных операций чтения из памяти. Подстановка констант, удаление мертвого кода. Анализ зависимостей в цикловых регионах программы. Цикловые оптимизации и условия их корректности. Оптимизации доступа в память. Векторные инструкции. Анализ и оптимизации на предикатном коде. Оптимизация на основе профилирования. Расширения: OpenMP, Cilk Plus.

## 2. Методика оптимизаций.

Выбор оптимальной платформы. Профилирование и определение наиболее затратных фрагментов кода. Выделение фрагментов кода, способных дать максимальный прирост производительности. Выбор оптимального алгоритма. Динамический и статический профилировщики программ.

## 3. Общие стратегии оптимизаций.

Загрузка программы. Подсистема ввода-вывода. Переключение контекста. Системные вызовы. Обращение к БД. Динамическое связывание. Обращения к памяти. Обращения к сети. Цепочки зависимостей.

## 4. Оптимизация на уровне написания кода.

Массивы структур vs структуры массивов. Векторные операции. Intrinsic. Inlining. Кэширование. Учет кэширования в процессоре. Выравнивание данных. Операции с плавающей точкой. Целочисленные операции. Ассемблерные вставки. Проверка диапазонов. Побитовые операции.

## 5. Особенности архитектур с точки зрения оптимизаций.

Архитектуры современных микропроцессоров. Особенности архитектур с точки зрения оптимизаций. Машинная модель.

## 6. Проектирование с учетом оптимизации.

Использование STL. Метапрограммирование. Указатели vs ссылки. Конструкторы, деструкторы. Возвращаемые значения. Поля и методы. Наследование. Поток. Поток ввода-вывода. Исключения и другие способы обработки ошибок. RTTI. Раскручивание стека. Move-семантика.

## 7. Устройство компилятора.

Введение. Понятие компиляции, виды оптимизирующей компиляции. Внутреннее представление программы и его построение. Управляющий граф, построение и использование.

## 8. Финальная оптимизация.

Межпроцедурный анализ программ. Межпроцедурные оптимизации.

## 5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютерной техникой с подключением к сети «Интернет» и мультимедийным оборудованием (проектор, звуковая система) для проведения занятий лекционного и семинарского типа.

## 6. Перечень рекомендуемой литературы

#### Основная литература

1. Кетков Ю.Л. Введение в языки программирования С и С++ / Ю.Л. Кетков. - Москва : Национальный Открытый Университет ИНТУИТ, 2016. - 291 с. - ISBN intuit144. - URL: <https://ibooks.ru/bookshelf/362851/reading>. - Текст: электронный.

#### Дополнительная литература

1. Стандарты программирования на С++ [Текст] = C++ Coding Standards : 101 правило и рекомендация / Г. Саттер, А. Александреску. - М. : ООО "И. Д. Вильямс", 2017. - 224 с.: ил. - (C++ In-Depth). - Библиогр.: с. 202-205. - Предм. указ.: с. 220-223. - 200 экз. - ISBN 978-5-8459-1933-5.

### **7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

1. Viva64.com
2. <http://www.intuit.ru/studies/courses/660/516/info>
3. <http://calcul.math.cnrs.fr/IMG/pdf/20120207-binet-opt-cxx.pdf>
4. <http://www.aristeia.com/FastwareForC++.html>
5. <http://meetingcpp.com/index.php/tv14/items/23.html>
6. <http://meetingcpp.com/index.php/tv14/items/20.html>
7. <http://www.viva64.com/en/l/>
8. <http://www.agner.org/optimize/>
9. <http://channel9.msdn.com/Shows/Going+Deep/Russell-Hadley-The-Route-to-C-Code-Optimization>

### **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

Для контроля и коррекции знаний, обучающиеся могут использовать компьютерное тестирование.

### **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Методические рекомендации позволяют студенту оптимальным образом организовать процесс обучения. В структуре учебного плана значительное время отводится на самостоятельное изучение данной дисциплины. В рабочей программе приведено примерное распределение часов аудиторной и внеаудиторной нагрузки по различным темам данной дисциплины.

Для успешного освоения данной дисциплины студенту необходимо:

- посещать лекции и семинары, при этом конспектирование материалов не является необходимым, поскольку основные материалы хранятся в кафедральной папке;
- выполнять задания, задаваемые преподавателем на лекциях и семинарах;
- выполнить итоговое письменное задание по дисциплине, которое вносит основной вклад в изучение дисциплины, а также в итоговую оценку по данному курсу.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

**по направлению:** Информатика и вычислительная техника

**профиль подготовки:** Физтех-школа Прикладной Математики и Информатики  
кафедра корпоративных информационных систем

**курс:** 4

**квалификация:** бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Дифференцированный зачет

**Разработчик:** Н.Ю. Пустовойтов, старший преподаватель

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценить качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Оптимизация C++ приложений» обучающийся должен:

### знать:

- Общее устройство парсеров регулярных, контекстно-свободных и контекстно-зависимых грамматик;
- общее устройство компиляторов языка C++;
- общие стратегии оптимизации;
- методику оптимизации;
- возможности компилятора по оптимизации.

### уметь:

- Осуществлять и обосновывать выбор оптимальной платформы;
- осуществлять проектирование архитектуры с учетом оптимизации производительности;
- осуществлять оптимизацию производительности при написании кода;
- осуществлять финальную оптимизацию.

### владеть:

- Средствами профилирования выполнения программы;
- ключами компиляции.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Перечень вопросов для промежуточного контроля:

1. Примеры оптимизаций.
2. Выбор оптимальной платформы.
3. Стратегии оптимизаций.
4. Виды оптимизаций на уровне написания кода.
5. Архитектура с точки зрения оптимизации.
6. Понятие компиляции, виды оптимизирующей компиляции.
7. Межпроцедурные оптимизации.

#### 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Перечень контрольных вопросов:

1. Устройство компилятора. Понятие компиляции, виды оптимизирующей компиляции.
2. Внутреннее представление программы и его построение. Управляющий граф, построение и использование.
3. Особенности архитектур с точки зрения оптимизаций.
4. Архитектуры современных микропроцессоров.
5. Машинная модель.
6. Методика оптимизаций. Выбор оптимальной платформы. Профилирование и определение наиболее затратных фрагментов кода. Выделение фрагментов кода, способных дать максимальный прирост производительности. Выбор оптимального алгоритма.
7. Динамический и статический профилировщики программ.
8. Общие стратегии оптимизаций. Загрузка программы.
9. Общие стратегии оптимизаций Подсистема ввода-вывода.
10. Общие стратегии оптимизаций Переключение контекста.
11. Общие стратегии оптимизаций Системные вызовы.
12. Общие стратегии оптимизаций Обращение к БД.
13. Общие стратегии оптимизаций Динамическое связывание.
14. Общие стратегии оптимизаций Обращения к памяти.
15. Общие стратегии оптимизаций Обращения к сети.
16. Общие стратегии оптимизаций Цепочки зависимостей.
17. Проектирование с учетом оптимизации. Использование STL.
18. Проектирование с учетом оптимизации. Метапрограммирование.
19. Проектирование с учетом оптимизации. Указатели vs ссылки. Конструкторы, деструкторы. Возвращаемые значения. Поля и методы. Наследование.
20. Проектирование с учетом оптимизации. Потoki. Потoki ввода-вывода. Исключения и другие способы обработки ошибок. RTTI. Раскручивание стека. Move-семантика.
21. Оптимизация на уровне написания кода. Массивы структур vs структуры массивов. Векторные операции.
22. Оптимизация на уровне написания кода. Intrinsic. Inlining. Кэширование. Учет кэширования в процессоре. Выравнивание данных. Операции с плавающей точкой. Целочисленные операции. Ассемблерные вставки. Проверка диапазонов. Побитовые операции.
23. Межпроцедурный анализ программ.
24. Межпроцедурные оптимизации.
25. Мелко-зернистые оптимизации.
26. Сбор общих подвыражений, удаление избыточных операций чтения из памяти. Подстановка констант, удаление мертвого кода.
27. Анализ зависимостей в цикловых регионах программы. Цикловые оптимизации и условия их корректности.
28. Оптимизации доступа в память.
29. Векторные инструкции.
30. Анализ и оптимизации на предикатном коде.
31. Оптимизация на основе профилирования.
32. OpenMP.
33. Cilk Plus.

#### Критерии оценивания

отлично (10) - Студент демонстрирует:

- ☐ систематизированные, глубокие и полные знания по всем разделам учебной программы, а также по основным вопросам, выходящим за ее пределы;
- ☐ точное использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- ☐ безупречное владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных и профессиональных задач;



☐ полное и глубокое усвоение основной и дополнительной литературы, рекомендованной учебной программой дисциплины;

☐ высокий уровень самостоятельности и инициативности при выполнении задач в рамках самостоятельных и практических заданий.

отлично (9) - Студент демонстрирует:

☐ систематизированные, глубокие и полные знания по всем разделам учебной программы;

☐ точное использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;

☐ владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных и профессиональных задач;

☐ полное и глубокое усвоение основной и дополнительной литературы, рекомендованной учебной программой дисциплины;

☐ высокий уровень самостоятельности и инициативности при выполнении задач в рамках самостоятельных и практических заданий.

отлично (8) - Студент демонстрирует:

☐ систематизированные, глубокие и полные знания по всем разделам учебной программы;

☐ использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;

☐ владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных и профессиональных задач;

☐ усвоение основной литературы, рекомендованной учебной программой дисциплины;

☐ самостоятельность и инициативность при выполнении задач в рамках самостоятельных и практических заданий.

хорошо (7) - Студент демонстрирует:

☐ систематизированные, глубокие и полные знания по всем поставленным вопросам в объеме учебной программы;

☐ использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;

☐ владение инструментарием учебной дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;

☐ усвоение основной литературы, рекомендованной учебной программой дисциплины;

☐ самостоятельность при выполнении задач в рамках самостоятельных и практических заданий.

хорошо (6) - Студент демонстрирует:

☐ достаточно полные и систематизированные знания по большинству поставленных вопросов в объеме учебной программы;

☐ использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;

☐ владение инструментарием учебной дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;

☐ усвоение основной литературы, рекомендованной учебной программой дисциплины;

☐ самостоятельность при выполнении задач в рамках самостоятельных и практических заданий.

хорошо (5) - Студент демонстрирует:

☐ достаточные знания в объеме учебной программы;

☐ использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;

☐ владение инструментарием учебной дисциплины, умение его использовать в решении научных и профессиональных задач;

☐ усвоение основной литературы, рекомендованной учебной программой дисциплины;

☐ самостоятельность при выполнении задач в рамках самостоятельных и практических заданий.

удовлетворительно (4) - Студент демонстрирует:

☐ достаточный объем знаний в рамках образовательного стандарта;

☐ в целом корректное использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы без существенных ошибок;

- ☐ владение инструментарием учебной дисциплины, умение его использовать в решении стандартных научных и профессиональных задач;
- ☐ усвоение основной литературы, рекомендованной учебной программой дисциплины;
- ☐ способность работать под руководством преподавателя при выполнении задач в рамках самостоятельных и практических заданий.

удовлетворительно (3) - Студент демонстрирует:

- ☐ недостаточно полный объем знаний в рамках образовательного стандарта;
- ☐ частично корректное использование научной терминологии, изложение ответа с существенными стилистическими и логическими ошибками;
- ☐ слабое владение инструментарием учебной дисциплины, некорректное его использование в решении стандартных научных и профессиональных задач;
- ☐ знание части основной литературы, рекомендованной учебной программой дисциплины;
- ☐ пассивность при выполнении задач в рамках самостоятельных и практических заданий.

неудовлетворительно (2) - Студент демонстрирует:

- ☐ фрагментарные знания в рамках образовательного стандарта;
- ☐ неумение использовать научную терминологию дисциплины, изложение ответа с существенными стилистическими и логическими ошибками;
- ☐ фрагментарные знания основной литературы, рекомендованной учебной программой дисциплины;
- ☐ пассивность при выполнении задач в рамках самостоятельных и практических заданий.

неудовлетворительно (1) - Студент демонстрирует:

- ☐ отсутствие знаний в рамках образовательного стандарта или отказ от ответа.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Дифференцированный зачет может проводиться по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.

При проведении устного дифференцированного зачета обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося по билету на дифференцированном зачете не должен превышать одного астрономического часа.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, конспектами лекций или другими материалами.