

*Федеральное государственное автономное образовательное учреждение высшего образования
«Российский университет дружбы народов»*

Факультет физико-математических и естественных наук

Рекомендовано МССН
*02.00.00 «Компьютерные
и информационные науки»*

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины

Параллельное и распределенной программирование

Рекомендуется для направления подготовки

02.04.02 Фундаментальная информатика и информационные технологии
(указываются код и наименование направления подготовки/специальности)

Направленность программы

«Управление инфокоммуникациями и интеллектуальные системы»
(наименование образовательной программы в соответствии с направленностью (профилем))

1. Цели и задачи дисциплины:

Целью дисциплины является введение учащихся в предметную область современных параллельных вычислений.

В процессе преподавания дисциплины решаются следующие задачи:

- освоение архитектурных принципов реализации параллельной обработки в вычислительных машинах;
- изучение методов и языковых механизмов конструирования параллельных программ;
- овладение параллельными вычислительными методами.

2. Место дисциплины в структуре ОП ВО:

Дисциплина *параллельное и распределенное программирование* относится к *обязательной* части блока 1 учебного плана.

В таблице № 1 приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП ВО.

Таблица № 1

Предшествующие и последующие дисциплины, направленные на формирование компетенций

№ п/п	Шифр и наименование компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
Универсальные компетенции (УК)			
	УК-1, УК-7	-	Научно-исследовательская работа, Преддипломная практика, ВКР
Общепрофессиональные компетенции			
	ОПК-1, ОПК-2, ОПК-3	-	Научно-исследовательская работа, Преддипломная практика, ВКР
Определяемые самостоятельно профессиональные компетенции (ПК). Тип задач профессиональной деятельности: научно-исследовательский			
	ПК-1.3	-	Научно-исследовательская работа
Тип задач профессиональной деятельности: производственно-технологический			
	ПК-2.5	-	Преддипломная практика, ВКР

3. Требования к результатам освоения дисциплины:

Процесс изучения дисциплины направлен на формирование следующих компетенций: УК-1; УК-7; ОПК-1; ОПК-2; ОПК-3; ПК-1 (в части ПК1.3); ПК-2 (в части ПК-2.5)

Расшифровка компетенций

УК-1 Способен осуществлять поиск, критический анализ проблемных ситуаций на основе системного подхода, выработать стратегию действий.

- **УК-1.1** Знает принципы сбора, отбора и обобщения информации.
- **УК-1.2** Умеет соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности.
- **УК-1.3** Имеет практический опыт работы с информационными источниками, опыт научного поиска, создания научных текстов.

УК-7 Способен: искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; проводить оценку информации, ее достоверность, строить логические умозаключения на основании поступающих информации и данных

- **УК-7.1** Знает принципы применения цифровых технологий для сбора, отбора и обобщения информации
- **УК-7.2** Умеет применять цифровые технологии для поиска, обработки, анализа, хранения и представления информации в области фундаментальной информатики и информационных технологий
- **УК-7.3** Владеет навыками применения цифровых технологий и методов поиска, обработки, анализа, хранения и представления информации в области фундаментальной информатики и информационных технологий

ОПК-1 Способен находить, формулировать и решать актуальные проблемы прикладной математики, фундаментальной информатики и информационных технологий

- **ОПК-1.1** Обладает фундаментальными знаниями в области математических и естественных наук, теории коммуникаций
- **ОПК-1.2** Умеет осуществлять первичный сбор и анализ материала, интерпретировать различные математические объекты
- **ОПК-1.3** Имеет практический опыт работы с решением математических задач и применяет его в профессиональной деятельности

ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение (в том числе отечественного производства) для решения задач профессиональной деятельности.

- **ОПК-2.1** Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с перечнем ПО, включенного в Единый Реестр Российских программ.
- **ОПК-2.2** Умеет анализировать типовые языки программирования, составлять программы.
- **ОПК-2.3** Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникации.

ОПК-3 Способен проводить анализ математических моделей, создавать инновационные методы решения прикладных задач профессиональной деятельности в области информатики и математического моделирования.

- **ОПК-3.1** Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей.
- **ОПК-3.2** Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем
- **ОПК-3.3** Имеет практический опыт применения и разработки программного обеспечения, тестирования программных продуктов

ПК-1 Проведение работ по обработке и анализу научно-технической информации и результатов исследований:

- **ПК-1.3** Умеет применять полученные знания в области фундаментальных научных основ математики и информатики, а также решать стандартные задачи собственной научно-исследовательской деятельности; умеет решать научные задачи с

пониманием существующих подходов к верификации моделей по тематике исследований в соответствии с выбранной методикой

ПК-2 Организационное и технологическое обеспечение проектирования и дизайна ИС:

- **ПК-2.5** Знает основы программирования; современные методики тестирования разрабатываемых информационных систем; современные инструменты и методы верификации программного кода.

В результате изучения дисциплины студент должен:

Знать:

- принципы сбора, отбора и обобщения информации (УК-1.1. УК-7.1)
- область математических и естественных наук, теории коммуникаций (ОПК-1.1)
- основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с перечнем ПО, включенного в Единый Реестр Российских программ (ОПК-2.1)
- методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей (ОПК-3.1)
- основы программирования; современные методики тестирования разрабатываемых информационных систем; современные инструменты и методы верификации программного кода (ПК-2.5).

Уметь:

- соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности (УК-1.2, УК-7.2)
- осуществлять первичный сбор и анализ материала, интерпретировать различные математические объекты (ОПК-1.2)
- анализировать типовые языки программирования, составлять программы (ОПК-2.3)
- соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем (ОПК-3.2)
- применять полученные знания в области фундаментальных научных основ математики и информатики, а также решать стандартные задачи собственной научно-исследовательской деятельности; умеет решать научные задачи с пониманием существующих подходов к верификации моделей по тематике исследований в соответствии с выбранной методикой (ПК-1.3).

Иметь (владеть):

- практический опыт работы с информационными источниками, опыт научного поиска, создания научных текстов (УК-1.3, УК-7.3)
- практический опыт работы с решением математических задач и применяет его в профессиональной деятельности (ОПК-1.3)
- практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникации (ОПК-2.3)
- практический опыт применения разработки программного обеспечения и тестирования программных продуктов (ОПК-3.3).

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет **5** зачетных единиц.

Вид учебной работы	Всего часов	Семестры
		Семестр 3 (модуль 5)
Аудиторные занятия (всего)	36	36
В том числе:		
<i>Лекции</i>	18	18
<i>Практические занятия (ПЗ)</i>		
<i>Семинары (С)</i>		
<i>Лабораторные работы (ЛР)</i>	18	18
Самостоятельная работа (всего)	144	144
Общая трудоемкость (часы)	180	180
Зачетных единиц	5	5

5. Содержание дисциплины

5.1. Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины.	Содержание раздела (темы)
1.	Актуальные стандарты языка C++.	<p>История языков C и C++.</p> <ul style="list-style-type: none"> • Стандарты C: K&R C, ANSI C, C99, C11. • Стандарты C++: C++98, C++03, C++11, C++14, C++17. • Идиоматические отличия языков C и C++. • Нововведения стандарта C++11 и более поздних стандартов в плане синтаксиса языка. <p>Авто вывод типов и универсальный инициализатор.</p>
2.	Стандартная библиотека шаблонов (STL) и нововведения современных стандартов	<p>Контейнерные классы, реализующие стандартные структуры данных:</p> <ul style="list-style-type: none"> • одномерный динамический массив <code>std::vector</code>, • кортежи <code>std::tuple</code> и пара <code>std::pair</code>, • отображение (словарь, ассоциативный массив) <code>std::map</code>, • очередь <code>std::queue</code>, • двусторонняя очередь <code>std::dqueue</code>, • множество <code>std::set</code>. <p>Вспомогательные классы</p> <ul style="list-style-type: none"> • Библиотека алгоритмов <code>std::algorithm</code>. • Замеры времени с помощью <code>std::chrono</code>. • Обертка <code>std::function</code> для вызываемых объектов (функций, лямбда-выражений и т.д.) <p>Математические функции из <code><cmath></code>, математические константы из <code><numbers></code> массивы из <code><valarray></code>, численные алгоритмы из <code><numeric></code>.</p>
3	Параллелизм, основанный на многопоточности. Класс <code>std::thread</code>	<p>Параллельное и асинхронное/конкурентное выполнение. Параллелизм на основе потоков для многопроцессорных систем с общей памятью.</p> <p>Нововведения стандарта C++11 и более поздних</p>

		<p>стандартов, касающиеся поддержки многопоточного программирования.</p> <ul style="list-style-type: none"> • Библиотека <code>std::thread</code> для управления потоками независимым от операционной системы образом. • Основные понятия, касающиеся многопоточности. Модели памяти, гонка данных. атомарные операции. <p>Концепция мьютекса/семафора.</p>
4	Распараллеливание стандартных алгоритмов	<ul style="list-style-type: none"> • Нововведения стандарта C++17, касающиеся политики выполнения, позволяющие распараллелить некоторые стандартные алгоритмы из <code>std::algorithm</code>. • Примеры распараллеливания стандартных алгоритмов. <p>Методология замеров времени работы программного кода, дающая статистически значимые результаты. Использование <code>std::chrono</code> для получения замеров времени с учетом многопоточности.</p>
5	Многопоточная генерация равномерно распределенных псевдослучайных чисел	<p>Генерирование равномерно распределенных случайных чисел.</p> <ul style="list-style-type: none"> • Алгоритмы LCG, XORSHIFT, KISS, Mersenne twister, xoroshiro, PCG. • Встроенные в стандарт C++11 и выше средства по генерации случайных чисел (<code>std::random</code>). Целочисленные типы данных из <code><cstdint></code>. • Системные псевдоустройства ОС UNIX <code>/dev/random</code> и <code>/dev/urandom</code>. • Тестирование псевдослучайной последовательности чисел. • Графические тесты и статистические наборы тестов. Построение гистограммы, графика квантиль-квантиль, диаграммы рассеяния, вычисление коэффициента автокорреляции и т.д. <p>Многопоточная генерация псевдослучайных чисел. Специфика инициализации генератора для каждого потока.</p>
6	Моделирование случайных процессов	<ul style="list-style-type: none"> • Метод Монте-Карло и его применение для моделирования случайных процессов. Его использование в многопоточном режиме. • Генераторы случайных распределений, отличных от равномерного (нормальное распределение, экспоненциальное распределение, распределение Вейбулла и т.д.), тесты для генераторов данного типа. <p>Повышение производительности при использовании потоков. Структуры данных, безопасные с точки зрения совместного использования с потоками выполнения.</p>
7	Шаблон	Шаблон (паттерн) программирования потребитель-

программирования производитель- потребитель	производитель (Producer-Consumer). Области применения. Реализация на основе потоков. Моделирование систем массового обслуживания с помощью данного паттерна. Использование очереди и двусторонней очереди Producer-Consumer. Моделирование Пуассоновского потока.
---------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.2. Разделы дисциплин и виды занятий

№ п/п	Наименование раздела дисциплины	Лекц.	Лаб. зан.	СРС	Все-го час.
1	Актуальные стандарты языка C++.	2		12	14
2	Стандартная библиотека шаблонов (STL) и нововведения современных стандартов	2	3	20	25
3	Параллелизм, основанный на многопоточности.	4	3	24	31
4	Распараллеливание стандартных алгоритмов	2	3	24	29
5	Многопоточная генерация равномерно распределенных псевдослучайных чисел	3	3	24	30
6	Моделирование случайных процессов	3	3	20	26
7	Шаблон программирования производитель-потребитель	2	3	20	25
	Итого:	18	18	144	180

6. Лабораторный практикум

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудо-емкость (час.)
1	Стандартная библиотека шаблонов (STL) и нововведения современных стандартов	Использование основных контейнерных классов и вспомогательных классов.	3
2	Параллелизм, основанный на многопоточности.	Основной функционал библиотеки std::thread	3
3	Распараллеливание стандартных алгоритмов	Параллельное выполнение стандартных алгоритмов и замеры времени	3
4	Многопоточная генерация равномерно распределенных псевдослучайных чисел	Генерация псевдослучайных чисел в параллельном режиме	3
5	Моделирование случайных процессов	Параллельный метод Монте-Карло	3
6	Шаблон программирования производитель-потребитель	Шаблон производитель-потребитель на основе потоков	3

7. Практические занятия (семинары)

Семинары не предусмотрены

8. Материально-техническое обеспечение дисциплины:

Мультимедийная учебная аудитория для проведения учебных занятий (в том числе для лекционного типа занятий, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации).

Компьютерные (дисплейные) классы с доступом к сети Интернет и электронно-образовательной среде Университета для выполнения обучающимися лабораторных работ и практических заданий по дисциплине и для проведения обучающимися самостоятельной работы, компьютерного тестирования обучающихся (при необходимости).

Высокопроизводительный информационно-коммуникационный комплекс (ИКК) НРС-01_SO000029-001 кафедры прикладной информатики теории вероятностей.

9. Информационное обеспечение дисциплины

а) программное обеспечение:

- ОС GNU/Linux, набор компиляторов GCC (g++), интерпретатор Python 3 с библиотеками NumPy, SciPy и Matplotlib.

б) базы данных, информационно-справочные и поисковые системы

1. Сайт библиотеки РУДН <http://lib.rudn.ru/>
2. Страница курса на телекоммуникационной учебно-информационной системе РУДН (ТУИС) <https://esystem.rudn.ru/course/view.php?id=85>

10. Учебно-методическое обеспечение дисциплины

а) основная литература

1. Уильямс Энтони С++ Практика многопоточного программирования. — СПб.: Питер, 2020. — 640 с. ISBN 978-5-4461-0831-2
2. Яцек Галовиц С++17 STL. Стандартная библиотека шаблонов. — СПб.: Питер, 2019. — 432 с. ISBN 978-5-4461-0680-6

б) дополнительная литература

1. Параллельные вычисления [Текст] : Учебное пособие / В.В. Воеводин, В.В. Воеводин. - СПб. : БХВ-Петербург, 2004. - 608 с. : ил. - ISBN 5-94157-160-7 : 304.81.
2. Основы параллельного программирования [Текст] / К.Ю. Богачев. - М. : БИНОМ. Лаборатория знаний, 2015. - 342 с. : ил. - ISBN 978-5-94774-037-0
3. Левин, М.П. Параллельное программирование с использованием OpenMP : учебное пособие / М.П. Левин. - Москва : Интернет-Университет Информационных Технологий, 2008. - 120 с. - (Основы информационных технологий). - ISBN 978-5-94774-857-4 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=233111>
4. Гергель, В.П. Теория и практика параллельных вычислений : учебное пособие / В.П. Гергель. - Москва : Интернет-Университет Информационных Технологий, 2007. - 424 с. : ил.,табл. - (Основы информационных технологий). - ISBN 978-5-9556-0096-3 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=233067>

11. Методические указания для обучающихся по освоению дисциплины (модуля)

Учебным планом на изучение дисциплины отводится один модуль. В дисциплине предусмотрены лабораторный практикум из 6 лабораторных работ. Каждую лабораторную работу необходимо защитить, составив отчет о выполнении. Итоговая оценка выставляется по сумме баллов, набранных в течении модуля. В случае, если учащийся не набрал необходимое для зачета количество баллов (51 балл), он может

получить недостающие баллы, выполнив дополнительные задания, предоставленные преподавателем.

11.1 Методические указания по самостоятельному освоению теоретического материала по дисциплине

Рекомендуется по указанным темам изучить литературу, указанную в п. 10 программы дисциплины и учебно-методические материалы в ТУИС (<http://esystem.pfur.ru>).

11.2 Методические указания по выполнению лабораторных работ

Задания по лабораторным работам выполняются индивидуально каждым студентом в соответствии с календарным планом и методическими указаниями по выполнению лабораторных работ по дисциплине.

- Для каждой лабораторной необходимо написать и скомпилировать все требуемые программы, а также провести требуемые тесты производительности и оформить результаты в виде таблиц и графиков.
- Для получения полных баллов необходимо очно защитить лабораторные, ответив на вопросы преподавателя и показав графики/таблицы с результатами работы программ.
- В случае сдачи без защиты необходимо оформить отчет и загрузить в соответствующий модуль на странице курса в ТУИС. В отчете должны быть представлены скриншоты запуска программы, результаты их работы, а также таблицы и графики если они требуются. В случае сдачи отчета без дальнейшей защиты ставится лишь 60% от полного балла за лабораторную.
- Срок сдачи указаны для каждой лабораторной работы отдельно.
- В случае сдачи лабораторной не в срок, но с очной защитой ставится не более 70% от максимального балла. Если сдача не в срок и без защиты, то ставится не более 50% от максимального балла.

12. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

ФОС по дисциплине представлен в приложении к данной программе,

Программа составлена в соответствии с требованиями ОС ВО РУДН.

Разработчики:

доцент кафедры прикладной информатики
и теории вероятностей, к.ф.-м.н.



М. Н. Геворкян

Руководитель программы

Заведующий кафедрой
прикладной информатики
и теории вероятностей, д.т.н., проф.



К.Е. Самуйлов

*Федеральное государственное автономное образовательное учреждение высшего образования
«Российский университет дружбы народов»*

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**

Параллельное и распределенной программирование

(наименование дисциплины)

02.04.02 Фундаментальная информатика и информационные технологии

(код и наименование направления подготовки)

«Управление инфокоммуникациями и интеллектуальные системы»

(наименование профиля подготовки)

Магистр

Квалификация (степень) выпускника

Паспорт фонда оценочных средств по дисциплине Параллельное и распределенной программирование

Направление: 02.04.02 Фундаментальная информатика и информационные технологии.

магистерская программа «Управление инфокоммуникациями и интеллектуальные системы»

Код контролируемой компетенции или ее части	Контролируемый раздел дисциплины	Контролируемая тема дисциплины	Формы контроля уровня освоения ООП		Баллы темы	Баллы раздела
			Аудиторная работа	СРС		
			Выполнение лабораторной работы			
УК-1; УК-7; ОПК-1; ОПК-2; ОПК-3; ПК-1 (в части ПК1.3); ПК-2 (в части ПК-2.5)	Современный стандарт языка C++.	Актуальные стандарты языка C++.	16		16	32
		Стандартная библиотека шаблонов (STL) и нововведения современных стандартов	16		16	
	Многопоточное программирования	Параллелизм, основанный на многопоточности.	16		16	68
		Распараллеливание стандартных алгоритмов	16		16	
		Многопоточная генерация равномерно распределенных псевдослучайных чисел	20		20	
		Моделирование случайных процессов	16		16	
	ИТОГО:			100		100

Процесс изучения дисциплины направлен на формирование следующих компетенций: УК-1; УК-7; ОПК-1; ОПК-2; ОПК-3; ПК-1 (в части ПК1.3); ПК-2 (в части ПК-2.5)

Компетенции и индикаторы их достижения

УК-1 Способен осуществлять поиск, критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий.

- **УК-1.1** Знает принципы сбора, отбора и обобщения информации.
- **УК-1.2** Умеет соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности.
- **УК-1.3** Имеет практический опыт работы с информационными источниками, опыт научного поиска, создания научных текстов.

УК-7 Способен: искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач; проводить оценку информации, ее достоверность, строить логические умозаключения на основании поступающих информации и данных

- **УК-7.1** Знает принципы применения цифровых технологий для сбора, отбора и обобщения информации
- **УК-7.2** Умеет применять цифровые технологии для поиска, обработки, анализа, хранения и представления информации в области фундаментальной информатики и информационных технологий
- **УК-7.3** Владеет навыками применения цифровых технологий и методов поиска, обработки, анализа, хранения и представления информации в области фундаментальной информатики и информационных технологий

ОПК-1 Способен находить, формулировать и решать актуальные проблемы прикладной математики, фундаментальной информатики и информационных технологий

- **ОПК-1.1** Обладает фундаментальными знаниями в области математических и естественных наук, теории коммуникаций
- **ОПК-1.2** Умеет осуществлять первичный сбор и анализ материала, интерпретировать различные математические объекты
- **ОПК-1.3** Имеет практический опыт работы с решением математических задач и применяет его в профессиональной деятельности

ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение (в том числе отечественного производства) для решения задач профессиональной деятельности.

- **ОПК-2.1** Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с перечнем ПО, включенного в Единый Реестр Российских программ.
- **ОПК-2.2** Умеет анализировать типовые языки программирования, составлять программы.
- **ОПК-2.3** Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникации.

ОПК-3 Способен проводить анализ математических моделей, создавать инновационные методы решения прикладных задач профессиональной деятельности в области информатики и математического моделирования.

- **ОПК-3.1** Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей.
- **ОПК-3.2** Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем
- **ОПК-3.3** Имеет практический опыт применения и разработки программного

обеспечения, тестирования программных продуктов

ПК-1 Проведение работ по обработке и анализу научно-технической информации и результатов исследований:

- **ПК-1.3** Умеет применять полученные знания в области фундаментальных научных основ математики и информатики, а также решать стандартные задачи собственной научно-исследовательской деятельности; умеет решать научные задачи с пониманием существующих подходов к верификации моделей по тематике исследований в соответствии с выбранной методикой

ПК-2 Организационное и технологическое обеспечение проектирования и дизайна ИС:

- **ПК-2.5** Знает основы программирования; современные методики тестирования разрабатываемых информационных систем; современные инструменты и методы верификации программного кода.

Балльно-рейтинговая система оценки уровня знаний

Сводная оценочная таблица дисциплины

Раздел	Тема	Формы контроля уровня освоения ООП	Баллы темы	Баллы раздела
		Лабораторные работы		
Современный стандарт языка C++.	Стандартная библиотека шаблонов (STL) и нововведения современных стандартов	16	16	32
	Параллелизм, основанный на многопоточности.	16	16	
Многопоточное программирование	Распараллеливание стандартных алгоритмов	16	16	68
	Многопоточная генерация равномерно распределенных псевдослучайных чисел	16	16	
	Моделирование случайных процессов	20	20	
	Шаблон программирования производитель-потребитель	16	16	
Итого			100	100

Таблица соответствия баллов и оценок

Баллы БРС	Традиционные оценки РФ	Оценки ECTS
95 - 100	5	A
86 - 94		B
69 - 85	4	C
61 - 68	3	D
51 - 60		E
31 - 50	2	FX
0 - 30		F
51-100	Зачет	Passed

Правила применения БРС

1. Раздел (тема) учебной дисциплины считаются освоенными, если студент набрал более 50% от возможного числа баллов по этому разделу (теме).
2. Студент не может быть аттестован по дисциплине, если он не освоил все темы и разделы дисциплины, указанные в сводной оценочной таблице дисциплины.
3. По решению преподавателя и с согласия студентов, не освоивших отдельные разделы (темы) изучаемой дисциплины, в течение учебного семестра могут быть повторно проведены мероприятия текущего контроля успеваемости или выданы дополнительные учебные задания по этим темам или разделам. При этом студентам за данную работу засчитывается минимально возможный положительный балл (51% от максимального балла).

4. Текущий контроль освоения тем осуществляется путем защиты учащимся сделанных лабораторных работ и ответов на вопросы преподавателя во время защиты.
5. Для каждой лабораторной необходимо написать и скомпилировать все требуемые программы, а также провести требуемые тесты производительности и оформить результаты в виде таблиц и графиков.
6. Для получения полных баллов необходимо очно защитить лабораторные, ответив на вопросы преподавателя и показав графики/таблицы с результатами работы программ.
7. В случае сдачи без защиты необходимо оформить отчет и загрузить в соответствующий модуль на странице курса в ТУИС. В отчете должны быть представлены скриншоты запуска программы, результаты их работы, а также таблицы и графики если они требуются. В случае сдачи отчета без дальнейшей защиты ставится лишь 60% от полного балла за лабораторную.
8. Срок сдачи указан для каждой лабораторной работы отдельно.
9. В случае сдачи лабораторной не в срок, но с очной защитой ставится не более 70% от максимального балла. Если сдача не в срок и без защиты, то ставится не более 50% от максимального балла.
10. При выполнении студентом дополнительных учебных заданий или повторного прохождения мероприятий текущего контроля полученные им баллы засчитываются за конкретные темы. Итоговая сумма баллов не может превышать максимального количества баллов, установленного по данным темам
11. График проведения мероприятий текущего контроля успеваемости формируется в соответствии с календарным планом курса. Студенты обязаны сдавать лабораторные работы в сроки, установленные преподавателем. Сроки указываются на страницы курса в ТУИС и объявляются преподавателем во время занятий.
12. Отсрочка в прохождении мероприятий текущего контроля успеваемости считается уважительной только в случае болезни студента, что подтверждается наличием у него медицинской справки, предоставляемой преподавателю не позднее двух недель после выздоровления. В этом случае выполнение контрольных мероприятий осуществляется после выздоровления студента в срок, назначенный преподавателем. В противном случае, отсутствие студента на контрольном мероприятии признается не уважительным.
13. Итоговая оценка выставляется по сумме набранных баллов за лабораторные работы.
14. Если в итоге за семестр (модуль) студент получил менее 51 балла, то он в обязательном порядке должен пройти переаттестацию. Во время переаттестации у него будет возможность добрать недостающие баллы и поднять общую оценку максимум до 51 балла.

Примерный перечень оценочных средств

п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
<i>Аудиторная работа</i>			
1.	Лабораторная работа	Система практических заданий, направленных на формирование практических навыков у обучающихся	Фонд практических заданий
<i>Самостоятельная работа</i>			
1	Выполнение домашних заданий	В качестве домашних заданий предлагаются лабораторные работы. Объем лабораторных работ подразумевает самостоятельную работу студента в отведенные для этого учебным планом часы.	Фонд практических заданий

Учебным планом на изучение дисциплины отводится один семестр. В дисциплине предусмотрены лекции, лабораторный практикум, контрольные мероприятия по проверке отчетов по лабораторным работам.

Оценивание результатов освоения дисциплины производится в соответствии с балльно-рейтинговой системой. По дисциплине предусмотрен дифф. зачет.

Итоговая оценка выставляется по сумме баллов, набранных в течении модуля. В случае, если учащийся не набрал необходимое для зачета количество баллов (51 балл), он может получить недостающие баллы, выполнив дополнительные задания, предоставленные преподавателем.

Критерии оценки по дисциплине

95-100 баллов:

- полное выполнение лабораторных работ;
- высокий уровень культуры исполнения лабораторных работ;
- активное участие в мероприятиях, предусмотренных программой дисциплины;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение программным обеспечением, умение эффективно использовать его в постановке и решении научных и профессиональных задач;
- выраженная способность самостоятельно и творчески решать поставленные задачи;
- полная самостоятельность и творческий подход при изложении материала по программе дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

86- 94 балла:

- полное выполнение лабораторных работ;
- высокий уровень культуры исполнения лабораторных работ;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- хорошее владение программным обеспечением, умение эффективно использовать его в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать поставленные задачи в нестандартных производственных ситуациях;
- усвоение основной и дополнительной литературы, нормативных и законодательных актов, рекомендованных программой дисциплины и преподавателем.

69-85 баллов:

- частичное выполнение лабораторных работ;
- хороший уровень культуры исполнения лабораторных работ;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- владение программным обеспечением, умение использовать его в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать проблемы в рамках программы дисциплины;
- усвоение основной литературы;

51-68 баллов:

- частичное выполнение лабораторных работ;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- удовлетворительное владение программным обеспечением, умение использовать его в постановке и решении научных и профессиональных задач;
- способность решать проблемы в рамках программы дисциплины;
- удовлетворительное усвоение основной литературы;

31 - 50 баллов – НЕ ЗАЧТЕНО:

- частичное выполнение лабораторных работ;
- недостаточно полный объем навыков и компетенции в рамках программы дисциплины;
- неумение использовать в практической деятельности научной терминологии, изложение ответа на вопросы с существенными стилистическими и логическими ошибками;
- слабое владение программным обеспечением по разделам программы дисциплины, некомпетентность в решении стандартных (типовых) производственных задач;
- способность решать проблемы в рамках программы дисциплины;
- удовлетворительное усвоение основной литературы;

0-30 баллов, НЕ ЗАЧТЕНО:

- отсутствие умений, навыков, знаний и компетенции в рамках программы дисциплины;
- невыполнение лабораторных заданий; отказ от ответа по программе дисциплины;
- игнорирование занятий по дисциплине по неуважительной причине.
- Программа составлена в соответствии с требованиями ОС ВО РУДН.

Комплект экзаменационных билетов

Дисциплина Параллельное и распределенной программирование
(наименование дисциплины)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

1. Предпосылки развития параллельных вычислений.
2. Задача параллелизации алгоритма.

Составитель М.Н. Геворкян

Заведующий кафедрой К.Е. Самуйлов

Дисциплина Параллельное и распределенной программирование
(наименование дисциплины)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №2

1. Классификации параллельных систем.
2. Оценка степени параллелизма алгоритма

Составитель М.Н. Геворкян

Заведующий кафедрой К.Е. Самуйлов

Дисциплина Параллельное и распределенной программирование
(наименование дисциплины)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №3

1. Масштабируемость вычислительного алгоритма.
2. Ускорение и эффективность вычислительного алгоритма.

Составитель М.Н. Геворкян

Заведующий кафедрой К.Е. Самуйлов

Дисциплина Параллельное и распределенной программирование
(наименование дисциплины)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №4

1. Сверхлинейное ускорение вычислительного алгоритма.
2. Закона Амдаля и ограничения им налагаемые.

Составитель М.Н. Геворкян

Заведующий кафедрой К.Е. Самуйлов

Дисциплина Параллельное и распределенной программирование
(наименование дисциплины)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №5

1. Отличие технологий MPI и OpenMP. Различия в их области применения.
2. Парадигма MIMD (Multiple Instructions Multiple Data).

Составитель М.Н. Геворкян

Заведующий кафедрой К.Е. Самуйлов

Дисциплина Параллельное и распределенной программирование
(наименование дисциплины)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №6

1. Характеристики технологии MPI.
2. Разные способы обмена сообщениями в MPI. Чем они отличаются?

Составитель М.Н. Геворкян

Заведующий кафедрой К.Е. Самуйлов

Дисциплина Параллельное и распределенной программирование
(наименование дисциплины)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №7

1. Укажите основные различия между синхронным (блокирующим) и асинхронным способами передачи сообщений.
2. Способы идентификации процессов и сообщений в рамках технологии MPI.

Составитель М.Н. Геворкян

Заведующий кафедрой К.Е. Самуйлов

Комплект вопросов для оценки усвоения разделов дисциплины

Примерный перечень вопросов, задаваемых во время защиты лабораторных работ для проверки усвоения материала.

Общие вопросы по параллельным вычислениям

1. Какие механизмы параллельного выполнения программ существуют?
2. Чем параллелизм на основе потоков отличается от параллелизма на основе процессов?
3. Чем асинхронные вычисления отличаются от параллельных?
4. Чем распределенные вычисления отличаются от параллельных?
5. Чем кластерная вычислительная система отличается от GRID системы?
6. Может ли программа использующая параллелизм на основе потоков выполняться на кластере? На GRID системе?
7. Что понимается под неопределенностями параллелизма (состоянием гонки) в параллельных вычислениях?
8. Что такое мьютекс?
9. Чем флаг отличается от мьютекса?

Стандарт C++11 и C++14

1. Какой синтаксис цикла `for` был введен в стандарте C++11?
2. Какие стандарты языка C существуют?
3. Какие стандарты языка C++ существуют?
4. Какой синтаксис инициализации структур и объектов введен в стандарте C++11?
5. Как новый синтаксис для инициализации объектов можно использовать для инициализации векторов?
6. Что такое вывод типов?
7. Как вывод типов применяется в языке C++11?
8. Какие типы могут быть выведены автоматически?
9. Как по умолчанию передаются аргументы примитивных типов в языке C++ функцию: по ссылке или по значению?
10. Как по умолчанию передаются аргументы контейнерных типов (векторы, кортежи, строки) в C++ по ссылке или по значению?
11. Что такое константная ссылка?
12. Как константную ссылку можно применять для обхода итерируемых объектов, используя синтаксис цикла `for` C++11?
13. Что такое анонимные функции?
14. Каков синтаксис анонимных функций в языке C++11?
15. Какие типы целых чисел есть в языке C++11?
16. Какие дополнительные целочисленные типы определены в заголовочном файле `cinttypes`?
17. Каков диапазон беззнакового целого типа?
18. Для чего нужна библиотека `chrono`?
19. Зачем нужны `chrono_literals`?
20. Как измерить время в наносекундах с помощью библиотеки `chrono`? Как перевести результат в целое или вещественное число?
21. Как реализован мьютекс в C++11? Какой заголовочный файл следует подключить для использования мьютекса?

22. Для чего нужна библиотека `thread` C++11?

Генераторы псевдослучайных чисел

1. Возможно ли с помощью компьютера генерировать случайные числа?
2. Какие алгоритмы генерации псевдослучайных чисел существуют?
3. Какому распределению подчиняются генерируемые стандартными алгоритмами псевдослучайные числа?
4. Какого типа числа генерируются стандартными алгоритмами?
5. Как из последовательности беззнаковых целых чисел двойной точности получить последовательность из полуинтервала $[0, 1)$?
6. Какие формулы позволяют из последовательности равномерно распределенных случайных чисел получить нормальное распределение?
7. Какие формулы позволяют из последовательности равномерно распределенных случайных чисел получить пуассоновское распределение?
8. Какие формулы позволяют из последовательности равномерно распределенных случайных чисел получить экспоненциальное распределение?
9. Что означает термин магические (волшебные) числа в алгоритмах генерации псевдослучайных чисел?

Фонд практических (лабораторных) заданий

Предлагаются к выполнению 6 лабораторных работ. Отчеты по лабораторным работам выполняются студентом самостоятельно, на лабораторном занятии студент может получить консультацию и методические указания от преподавателя.

Лабораторная/домашняя работа №1

Используя новые возможности C++11 по работе с потоками (стандартная библиотека `thread`) реализовать следующие тренировочные программы.

1. Программа порождает некоторое число потоков, каждый из которых бездействует определенное время, а затем выводит некоторое сообщение, после чего завершает работу. Число потоков передавать программе в виде аргумента командной строки при запуске.
2. Программа порождает поток-демон, который с определенной периодичностью записывает в специальный файл текущее время. Основная программа ждет некоторое время а затем завершает работу. Как отображается демон в списке процессов? Что с ним случается после завершения программы.
3. Программа порождает определенное число потоков, которые работают с общим ресурсом.

Лабораторная/домашняя работа №2

1. Целые типы данных: знаковые, беззнаковые, одинарной и двойной точности.
2. Работа с `std::chrono`. Доступные единицы измерения времени, преобразование их к целому типу.
3. Обертка `std::function` для вызываемых объектов (функций, лямбда-выражений и т.д.)
4. Некоторые алгоритмы из `std::algorithms` и `std::numeric`

Лабораторная/домашняя работа №3

1. Реализовать по крайней мере два алгоритма генерации псевдослучайных равномерно-распределенных чисел (при этом один из них обязательно должен использовать побитовые операции).
2. В качестве инициализирующих значений для генератора использовать биты из псевдоустройства Unix `/dev/random`.
3. Протестировать генераторы с помощью графического теста, нарисовав гистограмму для разного количества сгенерированных чисел.
4. Протестировать генераторы вычислив для разного количества сгенерированных чисел эмпирическое математическое ожидание и эмпирическую дисперсию. Сравнить их с теоретическими значениями для равномерного распределения.

Лабораторная/домашняя работа №4

1. Написать параллельный генератор псевдослучайных чисел, распределив работу между параллельными процессами. Воспользоваться одним из генераторов, который вы реализовали при выполнении лабораторной работы №2.
2. Для проверки работы генератора сгенерировать вначале небольшое количество псевдослучайных чисел и убедиться, что среди них нет периодических повторов.
3. Учесть, что для каждого потока необходимо отдельно задать начальное значение генератора иначе они могут сгенерировать одинаковую последовательность чисел.

Уточнения:

1. Для проверки того, что при многопоточном режиме программа работает быстрее, необходимо замерить время выполнения. Для этого используйте библиотеку `chrono` (см. видео и файлы, приложенные к данному заданию).
2. Та версия программы, для которой вы оцениваете время работы в многопоточном и однопоточном режиме, не должна ничего распечатывать в консоль или сохранять в файл, иначе вы будете оценивать скорость работы жесткого диска и буфера консоли. Также не нужно сохранять сгенерированные числа в массив (вектор), так как такое сохранение приведет к оценке производительности оперативной памяти и доступа к ней, а не скорости работы алгоритма.
3. Передача ссылок на объект при вызове функции-потока также может существенно замедлить производительность, так как потоки могут конкурировать за ресурс, ссылку на который вы передали.

Лабораторная/домашняя работа №5

1. На основе генераторов из предыдущей лабораторной, реализовать генерацию распределений, отличных от равномерного: нормальное распределение, распределение Пуассона, распределение Вейбула и экспоненциальное распределение.
2. Распределить работу между несколькими нитями.
3. Протестировать программу, нарисовав гистограмму распределений, а также вычислив эмпирическое математическое ожидание и эмпирическую дисперсию, сравнив их с теоретическими значениями для соответствующих распределений.

Лабораторная/домашняя работа №6

Реализовать используя потоки и шаблон производитель-потребитель (`producer-consumer`).
Схема работы программы следующая.

1. При запуске программы создаются минимум два потока.
2. Один из потоков (producer) производит некоторую информацию (например, генерирует случайные числа) и отправляет их в некоторый буфер (например, очередь).
3. Буфер имеет конечную длину и при полном заполнении producer или прекращает посылать данные или данные попросту теряются.
4. Второй из процессов (consumer) обращается к буферу, извлекает имеющуюся там информацию и использует ее в своей работе. Если буфер пустой, то он ждет появление информации, проверяя буфер с некоторой периодичностью.
5. Буфер должен обеспечивать корректную работу с множеством потоков и не допускать гонки данных.

• Порядок сдачи лабораторных работ

1. В каждой лабораторной необходимо реализовать программу на языке C++. Программа должна компилироваться без ошибок и предупреждений и выдавать корректные результаты.
2. Для получения полных баллов необходимо защитить лабораторные, ответив на вопросы преподавателя и показав результаты работы программы. Если в задании требуется, то необходимо построить графики/гистограммы.
3. В случае сдачи без защиты необходимо оформить отчет и загрузить его через портал. В отчете должны быть представлены: исходный текст программы, скриншоты запуска и результатов корректной работы, а также графики/гистограммы если они требуются. В случае сдачи отчета без дальнейшей защиты ставится лишь 70% от полного числа баллов за лабораторную.
4. Срок сдачи указаны для каждой лабораторной работы отдельно.
5. В случае сдачи лабораторной не в срок, но с защитой ставится не более 70% от максимального балла. Если сдача не в срок и без защиты, то ставится не более 50% от максимального балла.
6. На портале к лабораторным работам также доступны видеоматериалы с обучением и пояснениями.

Критерии оценки

Оценивается полнота выполнения работы, оформление результатов расчетов и оценки производительности. Также оцениваются ответы на вопросы преподавателя по исходному тексту программы.