

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

Саровский физико-технический институт -

филиал федерального государственного автономного образовательного учреждения высшего
образования «Национальный исследовательский ядерный университет «МИФИ»
(СарФТИ НИЯУ МИФИ)

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ЭЛЕКТРОНИКИ

Кафедра «Вычислительной и информационной техники»

УТВЕРЖДАЮ

Декан ФИТЭ, к.ф.-м.н., доцент

_____ **В.С. Холушкин**

« ____ » _____ **2023 г.**

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ПРИМЕНЕНИЕ КОМПЬЮТЕРОВ В НАУЧНЫХ ИССЛЕДОВАНИЯХ

наименование дисциплины

Направление подготовки (специальность) _____ **03.03.01 Прикладная математика и физика**

Наименование образовательной программы _____

Квалификация (степень) выпускника _____ **бакалавр**

Форма обучения _____ **очная**

Программа одобрена на заседании кафедры

Зав. кафедрой ВИТ

к.ф.-м.н., доцент

протокол № _____ от _____ **20** г.

_____ **В.С. Холушкин**

« ____ » _____ **2023 г.**

г. Саров, 2023 г.

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ВИТ, к.ф.-м.н., доцент

В.С. Холушкин

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ВИТ, к.ф.-м.н., доцент

В.С. Холушкин

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ВИТ, к.ф.-м.н., доцент

В.С. Холушкин

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ВИТ, к.ф.-м.н., доцент

В.С. Холушкин

Семестр	В форме прак- тической подго- товки	Трудоемкость, кред.	Общий объем курса, час.	Лекции, час.	Практич. заня- тия, час.	Лаборат. работы, час.	СРС, час.	КР/ КП	Форма(ы) кон- троля, экз./зач./ЗсО/	Интерактивные часы
3	32	5	180	32	32	-	80	-	Э	6
ИТОГО	32	5	180	32	32	-	80	-	36	8

АННОТАЦИЯ

Курс посвящен изучению теоретических и практических аспектов и основ алгоритмизации и программирования, а также применение современных вычислительных средств в научных исследованиях. Изучаются способы и методы разработки эффективных алгоритмов для решения различных прикладных задач.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В дисциплинах «Применение компьютеров в научных исследованиях» рассматриваются теоретические и практические аспекты основ алгоритмизации и программирования, а также применение современных вычислительных средств в научных исследованиях. Изучаются способы и методы разработки эффективных алгоритмов для решения различных прикладных задач. Рассматриваются общие принципы создания программ на языках высокого уровня и применения этих программ для научных исследований. Главная цель преподавания дисциплин – подготовка специалиста, владеющего фундаментальными знаниями и практическими навыками в области применения компьютеров в научных исследованиях. Задачи дисциплин - дать основы:

- Понятий алгоритмов, их классификаций, методов описания и представления алгоритмов, предназначенных для написания программ, используемых для решения задач из различных предметных областей;
- Структур языков программирования и инструментальных сред разработки программных продуктов;

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина «Применение компьютеров в научных исследованиях» являются базовыми дисциплинами федерального государственного образовательного стандарта высшего профессионального образования (ФОС ВО).

Изучение дисциплины «Применение компьютеров в научных исследованиях» основывается на базе знаний, умений и компетенций, полученных студентами в ходе освоения курса «Информатика», «Алгоритмические языки», «Программирование»

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Ожидается, что в результате освоения дисциплины студент приобретет следующие компетенции:

Общепрофессиональные компетенции (ОПК)

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности, в том числе в сфере педагогической деятельности	З-ОПК-1 Знать фундаментальные основы, полученные в области информационных технологий, естественных и гуманитарных наук, знать методы анализа информации. У-ОПК-1 Уметь использовать на практике углубленные Фундаментальные знания, полученные в области естественных и гуманитарных наук. В-ОПК-1 Владеть навыками обобщения, синтеза и анализа фундаментальных знаний, полученные в области информационных технологий, естественных и гуманитарных наук, владеть научным мировоззрением
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	З-ОПК-2 Знать современные информационные технологии и программные средства для решения задач профессиональной деятельности У-ОПК-2 Уметь выбирать и использовать современные информационные технологии и программные средства для решения задач профессиональной деятельности В-ОПК-2 Владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	З-ОПК-3 Знать современные средства представления результатов научно-технической деятельности, в том числе в форме отчетов, публикаций, презентаций, докладов. У-ОПК-3 Уметь использовать современные средства для представления результатов деятельности, составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты). В-ОПК-3 Владеть навыками представления результатов научно-технической деятельности с использованием современных средств, ориентируясь на потребности аудитории, в том числе в форме отчетов, публикаций.
ОПК-5 Способен участвовать в проведении фундаментальных и прикладных исследований и разработок, самостоятельно осваивать новые теоретические, в том числе математические, методы исследований и работать на современной экспериментальной научно-исследовательской, измерительно-аналитической и технологической аппаратуре	З-ОПК-5 Знать современные теоретические, в том числе математические, и экспериментальные методы исследований для решения профессиональных задач. У-ОПК-5 Уметь применять знания в области математики, физики и других наук в профессиональной деятельности, в том числе для проведения научных и прикладных исследований, их экспериментального и теоретического изучения, уметь самостоятельно осваивать новые теоретические, в том числе математические, методы исследований. В-ОПК-5 Владеть навыками проведения фундаментальных и прикладных исследований и разработок, работы на со-

	временной экспериментальной научно-исследовательской, измерительно-аналитической и технологической аппаратуре
--	---

4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ*

№ п/п	Наименование раздела /темы дисциплины	№ недели	Виды учебной работы						
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий кон- троль (форма)*	Максимальный балл (см. п. 5.3)	
			32	32	-	80			
Семестр 3									
Раздел 1.									
1.1.	Тема 1. Процесс написания программ.	1-2	5	5		16	УО. Защита ЛР	4	
Раздел 2.									
2.1	Тема 1. Безопасность программного обеспечения	3-4	5	5		16	УО. Защита ЛР	4	
2.2	Тема 2. Аппаратное обеспечение	5-7	5	5		16	УО. Защита ЛР	4	
Рубежный контроль		8	СР						5
Раздел 3.									
3.1	Тема 1. Параллельное программирование	9-11	5	5		16	УО. Защита ЛР	4	
3.2	Тема 2. Объектно-ориентированное программирование	12-15	7	7		16	УО. Защита ЛР	4	
Рубежный контроль		16	СР						10
Промежуточная аттестация			Э				-	50	
Посещаемость									5
Итого:			32	32	-	80	-	100	

*Сокращение наименований форм текущего, рубежного и промежуточного контроля:

УО – устный опрос

ЛР-лабораторная работа

СР – самостоятельная работа(решение задачи на заданную тему)

РГР – расчетно – графическая работа

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс

№	Наименование раздела /темы дисциплины	Содержание
1 семестр		
Раздел 1.		
1.1	Тема 1. Процесс написания программ.	Оформление текстов, требования к текстам и комментариям. Сопровождение программ. Документация на ПО, <i>SDP (software development plan</i> – план разработки ПО). Базовые основы элементарной техники программирования. Технические основы программной реализации формальных структур данных, итераторы. Списки, очереди, стеки, наборы, упорядоченные наборы, массивы, деревья, хранение графов, <i>hash</i> -таблицы; примеры использования структур данных, как выбрать структуру, соответствующую задаче. Применение <i>hash</i> -таблиц.
Раздел 2.		
2.1	Тема 1. Безопасность программного обеспечения	Написание программ без «дырок». Техника кодирования защищенных программ и типичные ошибки. Переполнение буфера , определение уровня доступа, работа с минимально возможными привилегиями, криптография и ее корректное применение, предохранение секретных данных, работа с входными данными, проблемы разных путей доступа к одним и тем же данным, запросов к базам данных и веб-страницам, а также проблемы поддержки интернационального ПО. Моделирование угроз. Классификация опасностей <i>STRIDE</i> – <i>Spoofing</i> (подмена данных), <i>Tampering</i> (подделка и изменение содержания данных), <i>Repudiation</i> (незаконный отказ от проведенной операции), <i>Information disclosure</i> (разглашение информации), <i>Denial of service</i> (отказ в обслуживании), <i>Elevation of privilege</i> (незаконное поднятие привилегий). Методика оценки риска <i>DREAD</i> – <i>Damage potential</i> (что может быть сломано), <i>Reproducibility</i> (повторяемость), <i>Exploitability</i> (пригодность угрозы для использования), <i>Affected users</i> (на каких пользователей повлияет), <i>Discoverability</i> (возможно ли детектировать факт использования).
2.2	Тема 2. Аппаратное обеспечение	Эволюция современного аппаратного обеспечения и ее влияние на программное обеспечение. Гипертрედовые (многопоточ-

		ные) и многоядерные процессоры, универсальные графические (GPU) процессоры и новые возможности по их использованию
Раздел 3.		
3.1	Тема 1. Параллельное программирование	Параллельные программы – от работы с разделяемой памятью, использования массивно-параллельных компьютеров и до распределенных расчетов на многих физических компьютерах. Декомпозиция задач на параллельные куски. Параллелизм данных, параллелизм кода. Паттерны параллельного программирования: параллелизм на уровне задач – декомпозиция задачи, «разделяй и властвуй» – декомпозиция задач и данных, геометрическая декомпозиция – декомпозиция данных, конвейерное исполнение – декомпозиция потока данных, «фронт волны» – декомпозиция данных с «многомерными» зависимостями. Пример типового шаблона программирования – пул нитей. Работа с разделяемой памятью. Синхронизационные примитивы (низкоуровневые атомарные команды, критические секции, взаимоисключающая блокировка – <i>mutex</i>, рекурсивная блокировка – <i>lock</i>, блокировка чтения-записи – <i>read/write lock</i>, многопроцессорная блокировка – <i>spinlock</i>, семафоры, барьеры). Реализация одних примитивов через другие, относительная «мощность» примитивов. Задача о консенсусе как способ оценки примитивов. Техническая специфика параллельных программ – производительность, условия «гонки» – <i>race condition</i>, взаимная блокировка – <i>deadlock</i>, повторно-входимые программы, потоко-безопасные (<i>thread-safe</i>) библиотеки. Неблокирующие примитивы синхронизации, транзакционная память. Организация высоко-производительных параллельных вычислений. Проблемы, специфические для параллельного исполнения многонитевых программ – синхронизация (между несколькими нитями), коммуникация (проблемы полосы пропускания и задержек, связанных с обменом данными между нитями), балансировка нагрузки (между нитями), масштабируемость (эффективность использования многих нитей). Написание высокопроизводительных программ, оценка условий и выбор способов реализации.
3.2	Тема 2. Объектно-ориентированное	Принципы и философия ООП в языках, программных системах и операционных системах. Инкапсуляция, полиморфизм и

	программирование	наследование/агрегирование. Динамический и статический подход в описании классов. Краткий обзор ООП реализации в языках C++. Интерфейсы, полиморфизм и перегрузка операторов в C++. Параметризованные классы. Дружественные функции. Потоки ввода-вывода в C++. Наследование как один из вариантов комбинирования объектов. Краткий сравнительный обзор ООП реализации в языках C++ и ObjectiveC, позднее и раннее связывание. Техническая организация ООП поддержки языков программирования для позднего связывания. Event-driven и message-driven программирование на примере XWindows Widgets, Mac OS X Interface Builder и подсистемы GDI MS Windows. Принципы агрегирования COM-объектов. Принципы поддержки ООП в операционных системах и языках, общее и различное. Реализация исключений в языке C++ и в ОС Window
--	------------------	--

Лабораторные занятия

Лабораторная работа №1 Основы работы и создание проекта для разработки приложений, особенности компиляции, отладки и выполнения программ в среде Microsoft Visual Studio

Лабораторная работа №2 Разработка и реализация программ с использованием операций двухточечных обменов. Исследование эффективности передачи данных при различных режимах обмена.

Лабораторная работа №3 Разработка и реализация программ с использованием коллективных операций обмена

Лабораторная работа №4 Разработка и реализация программ с использованием конструкторов типов данных. Исследование эффективности передачи однородных и разнородных данных для различных конструкторов типов данных.

Лабораторная работа №5 Разработка и реализация программ с использованием конструкторов групп и коммуникаторов.

Лабораторная работа №6 Разработка и реализация программ с использованием конструкторов виртуальных топологий.

Лабораторная работа №7 Разработка и реализация программ работы с контейнерами с использованием библиотек.

4.3. Перечень учебно-методического обеспечения для самостоятельной работы студентов

ОСНОВНАЯ ЛИТЕРАТУРА:

1. Литвиненко Н. А. - Технология программирования на С++ .Год издания: 2010
Издательство: БХВ-Петербург
2. Романов Е.Л. Си++. От дилетанта до профессионала Год издания: 2014
3. Макс Шлее - Qt4.5 Профессиональное программирование на С++. Год издания: 2010
4. Стили и методы программирования: курс лекций: учебное пособие
Автор: Непейвода Н.Н.Издательство: Интернет-Университет Информационных Технологий, 2005 г.
5. Введение в программирование: Учебное пособие Авторы: Баженова И.Ю., Сухомлин В.А. Издательство: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007 г.
6. Программирование на языке С++ Автор: Павловская Т.А. Издательство: Интернет-Университет Информационных Технологий, 2010 г.
7. Шлее М. Qt4.5 Профессиональное программирование на С++. СПб.: БХВ-Петербург, 2010.
8. Травова Н.Н. Лабораторный практикум по Qt часть 2(размещен на Server-129 в папке public/Travova/Qt).
9. Бланшет Ж., Саммерфилд М. Qt4. Программирование GUIна С++. 2-е изд. М.: КУДИЦ-Пресс, 2007.
10. Земсков Ю. Qt4 на примерах. СПб.: БХВ-Петербург, 2008.
11. Шилдт Г. Полный справочник по С++. 4-е изд. М.: Вильямс, 2006
12. Стефенс Д. Р. С++. Сборник рецептов. М.: КУДИЦ-Пресс, 2007.
13. Саммерфилд М. Qt4. Профессиональное программирование. Разработка кросс-платформенных приложений на С++. СПб.: Символ-Плюс 2011.
14. Дж.Макконнелл Анализ алгоритмов. Вводный курс. М.: ТЕХНОСФЕРА, 2002.
15. Буч Г. Объектно-ориентированный анализ и проектирование. – 2-е изд. – М.: БИНОМ, 1998.

5.ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИ- НЫ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

5.1. Паспорт фонда оценочных средств по дисциплине

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

Раз- дел	Темы занятий	Компетенция	Индикаторы освоения	Текущий контроль, неделя
Семестр 3				
Раздел 1	Тема 1. Процесс написания программ.	ОПК-1, ОПК2, ОПК-3, ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО2, Защита ЛР2
Раздел 2	Тема 1. Безопасность программного обеспечения	ОПК-1, ОПК2, ОПК-3, ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО4, Защита ЛР4
	Тема 2. Аппаратное обеспечение		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО7, Защита ЛР7
Рубежный контроль		ОПК-1, ОПК2, ОПК-3, ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	СР 8
Раздел 3	Тема 1. Параллельное программирование	ОПК-1, ОПК2, ОПК-3, ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО11, Защита ЛР11
	Тема 2. Объектно-ориентированное программирование		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО15, Защита ЛР15
Рубежный контроль		ОПК-1, ОПК2, ОПК-3, ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	СР 16
Промежуточная аттестация		ОПК-1, ОПК2, ОПК-3, ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	Экзамен

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы

5.2.1. Оценочные средства для текущего контроля

5.2.1.1. Примерные вопросы для устного опроса (УО)

1. Теоретические основы разработки параллельных приложений.
2. Принципы взаимодействия процессов. Базовые понятия и программная реализация принципа передачи данных.
3. Интерфейс передачи сообщений (MPI). Основные понятия и определения.
4. Причины возникновения технологии разработки параллельных приложений.
5. Основные принципы выполнения параллельной обработки данных. Понятие
6. параллельной программы и процесса.
7. Общие принципы передачи сообщений. Понятие коммуникационного пространства.
8. Особенности взаимодействия параллельных процессов с использованием блокирующих и не блокирующих операций двухточечных обменов MPI. Коммуникационные режимы.
9. Понятие блокирующей и не блокирующей операции. Принцип взаимодействия процессов при использовании различных режимов передачи данных.
10. Управление процессом передачи данных при выполнении не блокирующих обменов. Понятие дескриптора запроса.
11. Реализация интерфейса передачи данных (MPICH), обзор наиболее используемых реализаций, отличительные особенности.
12. Общая характеристика среды выполнения MPI программ.
13. Реализация библиотеки функций передачи сообщений.
14. Основные понятия объектно-ориентированного программирования в C++.
15. Понятие класса.
16. Режимы доступа.
17. Перегруженные функции.
18. Наследование.
19. Конструкторы.
20. Деструкторы.
21. Конструктор с параметрами.
22. Конструктор копирования.

23. Дружественные функции.
24. Дружественные классы.
25. Аргументы функций, задаваемые по умолчанию.
26. Структуры и классы.
27. Объединения и классы.
28. Наследование классов.
29. Режимы доступа при наследовании.

5.2.1.2. Примерные задания для самостоятельной работы (СР)

Поиск ключа узла дерева

Исходные данные: дерево, каждый элемент которого хранит уникальный, случайно сгенерированный ключ; ключ, узел с которым требуется найти, количество используемых вычислителей.

Цель: написать параллельную программу для поиска данного ключа.

Программа должна создать заданное пользователем дерево, считать количество используемых вычислителей и ключ для поиска, найти элемент дерева, содержащий этот ключ, и вывести информацию об этом узле.

2. Поиск максимального элемента дерева

Исходные данные: дерево, каждый элемент которого хранит уникальный, случайно сгенерированный ключ, количество используемых вычислителей.

Цель: написать параллельную программу для поиска максимального ключа.

Программа должна создать заданное пользователем дерево, считать количество используемых вычислителей, найти элемент дерева, содержащий максимальный из ключей и вывести информацию об этом узле.

Переменная, содержащая максимальный элемент, должна быть общей для всех вычислителей. Доступ к ней – синхронизован.

3. Поиск суммы всех элементов дерева

Исходные данные: дерево, каждый элемент которого хранит уникальный, случайно сгенерированный ключ, количество используемых вычислителей.

Цель: написать параллельную программу для поиска суммы всех ключей дерева.

Программа должна создать заданное пользователем дерево, считать количество используемых вычислителей, найти и вывести сумму всех ключей дерева.

Переменная, содержащая сумму ключей, должна быть общей для всех вычислителей. Доступ к ней – синхронизован.

4. Поиск количества узлов дерева

Исходные данные: дерево, количество используемых вычислителей.

Цель: написать параллельную программу для поиска количества узлов в дереве.

Программа должна создать заданное пользователем дерево, считать количество используемых вычислителей, найти и вывести количество всех узлов в дереве.

Переменная, содержащая максимальный элемент, должна быть общей для всех вычислителей. Доступ к ней – синхронизован.

Возможные алгоритмы распараллеливания работы для задач 1–4:

а) Пусть есть N вычислителей. У первого – ссылка на корень дерева. Он берет одно из поддеревьев для поиска себе, остальные раздает свободным вычислителям. Далее вычислители поступают точно так же. Когда свободных вычислителей не остается, каждый выполняет обычный поиск по своему поддереву.

б) Пусть есть N вычислителей и выбран вычислитель-мастер. Он сам проводит поиск ключа по дереву в ширину до тех пор, пока не найдет N независимых поддеревьев, после чего раздает их для поиска остальным вычислителям.

Усложнение для задач 1–3: пусть каждый узел дерева содержит дополнительную информацию – количество узлов под ним. К основной цели задачи добавляется равномерное распределение

загрузки вычислителей: каждый должен получить поддерево (поддеревья) с примерно одинаковым количеством узлов (отличающимся не более чем на единицу).

5.2.2. Оценочные средства для рубежного контроля

5.2.2.1. Примерные задания для решения задач по заданной теме

Вычисление интеграла численными методами

Исходные данные: функция одной переменной, отрезок интегрирования, количество вычислителей.

Цель: написать параллельную программу для вычисления интеграла от данной функции по данному отрезку.

Программа должна считать функцию, отрезок интегрирования, количество вычислителей, посчитать интеграл в несколько вычислителей, вывести результат.

Предполагаемый алгоритм распараллеливания: использовать параллелизм по отрезку. Каждый вычислитель считает интеграл по своей части отрезка с помощью одного из стандартных методов (трапеции, Симпсона, Ньютона–Лейбница и т. д.). Результаты отдельных вычислителей суммируются.

Численный метод выбирается преподавателем.

Вычисление интеграла методом Монте-Карло

Исходные данные: функция одной переменной, отрезок интегрирования, количество вычислителей.

Цель: написать параллельную программу для вычисления интеграла от данной функции по данному отрезку методом Монте-Карло.

Программа должна считать функцию, отрезок интегрирования, количество вычислителей, посчитать интеграл в несколько вычислителей, вывести результат.

Предполагаемый алгоритм распараллеливания:

а) Использовать параллелизм по отрезку. Каждый вычислитель считает интеграл по своей части отрезка. Результаты отдельных вычислителей суммируются.

б) Использовать параллелизм по количеству испытаний в методе Монте-Карло. Необходимое количество испытаний делится между N вычислителями. Результаты испытаний обобщаются.

Параллельное перемножение матриц

Исходные данные: две матрицы $n \times t$ и $t \times n$, количество вычислителей.

Цель: написать программу для перемножения матриц параллельно несколькими вычислителями.

Программа должна считать t , n , две матрицы, количество вычислителей; распараллелить процесс их перемножения.

Предполагаемый алгоритм распараллеливания: поскольку каждый элемент матрицы-результата рассчитывается независимо от других, он может быть определен отдельным вычислителем. Вычислитель-мастер занимается распределением работы между остальными. Он хранит базу свободных вычислителей, куда синхронизованно заносятся не имеющие работы и откуда берутся рабочие для вычисления следующего элемента матрицы.

Решение уравнения $Ax = b$ (A – матрица) методом Крамера: параллелизм по элементам столбца решения

Исходные данные: невырожденная матрица $n \times t$, столбец $n \times 1$, количество вычислителей.

Цель: написать программу для расчета решения уравнения $Ax = b$ параллельно несколькими вычислителями.

Программа должна считать t , n , матрицу A , столбец b , количество вычислителей; распараллелить решение уравнения $Ax = b$ методом Крамера.

Предполагаемый алгоритм решения: методом Крамера каждый элемент столбца-результата вычисляется независимо от остальных. Поэтому его можно передать для расчета отдельному вычислителю.

5.2.3. Оценочные средства для промежуточной аттестации

5.2.3.2. Примерные вопросы к экзамену

1. Теоретические основы разработки параллельных приложений.
2. Принципы взаимодействия процессов. Базовые понятия и программная реализация принципа передачи данных.

3. Интерфейс передачи сообщений (MPI). Основные понятия и определения.
4. Причины возникновения технологии разработки параллельных приложений.
5. Основные принципы выполнения параллельной обработки данных. Понятие
6. параллельной программы и процесса.
7. Общие принципы передачи сообщений. Понятие коммуникационного пространства.
8. Особенности взаимодействия параллельных процессов с использованием блокирующих и не блокирующих операций двухточечных обменов MPI. Коммуникационные режимы.
9. Понятие блокирующей и не блокирующей операции. Принцип взаимодействия процессов при использовании различных режимов передачи данных.
10. Управление процессом передачи данных при выполнении не блокирующих обменов. Понятие дескриптора запроса.
11. Реализация интерфейса передачи данных (MPICH), обзор наиболее используемых реализаций, отличительные особенности.
12. Общая характеристика среды выполнения MPI программ.
13. Реализация библиотеки функций передачи сообщений.
14. Инструменты среды разработки, особенности отладки параллельных программ. Понятие интерфейса абстрактного устройства (ADI).
15. Барьерная синхронизация, коллективная операция рассылки данных.
16. Классификация коллективных операций обмена.
17. Понятие синхронизации процессов передачи данных.
18. Особенности выполнения операций барьерной синхронизации и широковещательной рассылки данных.
19. Принципы взаимодействия процессов при выполнении коллективных операций сборки и рассылки данных.
20. Организация операций сборки и рассылки блоков данных с непрерывным расположением в памяти и данных, расположенных со смещением.
21. Проблема устранения взаимоблокировок процессов при выполнении коллективных операций. Обобщенные операции сборки и рассылки данных.
22. Особенности выполнения операций редукции.
23. Классификация базовых типов операций редукции. Принцип коллективной операции обработки данных.
24. Дополнительные типы данных операций редукции. Обобщенная операция редукции.
25. Производные типы данных. Конструкторы типов.
26. Понятие сигнатуры типа. Карта типа. Оптимизация передачи нестандартных данных при использовании производных типов данных.

27. Классификация конструкторов типов данных. Особенности использования операций конструирования производных типов данных.
28. Упаковка и распаковка данных.
29. Использование механизма передачи данных разной структуры и объема. Оценка эффективности. Проблемы оптимизации обмена разнородными данными.
30. Понятие группы процессов. Конструкторы групп
31. Группа процессов, инициализация процессов в группе. Понятие “пустой” группы, базовой группы, идентификатора группы.
32. Конструкторы на базе одной и двух групп. Особенности использования операций конструирования групп. Деструкторы групп.
33. Взаимосвязь групп и коммуникаторов. Коммуникационное пространство, контекст коммуникатора. Предопределенный коммуникатор.
34. Конструкторы коммуникаторов, особенности операций конструирования коммуникаторов. Деструктор коммуникатора.
35. Декартова топология.
36. Понятие виртуальной топологии, оптимизация параллельных алгоритмов с использованием механизма виртуальных топологий.
37. Конструкторы декартовой топологии.
38. Функции описания декартовой топологии.
39. Топология графа как общий вид виртуальной топологии
40. Конструкторы топологии графа. Функции описания топологии графа.
41. Особенности использования топологии графа для увеличения эффективности параллельных программ, преимущества и недостатки
42. Понятие интеркоммуникатора, конструктор интеркоммуникаторов.
43. Создание и цели использования межгрупповых коммуникаторов.
44. Механизм обмена данными между различными коммуникационными пространствами.
45. Основные понятия объектно-ориентированного программирования в C++.
46. Понятие класса.
47. Режимы доступа.
48. Перегруженные функции.
49. Наследование.
50. Конструкторы.
51. Деструкторы.
52. Конструктор с параметрами.
53. Конструктор копирования.

54. Дружественные функции.
55. Дружественные классы.
56. Аргументы функций, задаваемые по умолчанию.
57. Структуры и классы.
58. Объединения и классы.
59. Наследование классов.
60. Режимы доступа при наследовании.
61. Конструкторы с параметрами при наследовании.
62. Множественное наследование.
63. Перегрузка функций - членов класса.
64. Перегрузка операций.
65. Ключевое слово **this**.
66. Перегрузка операций ввода-вывода. Инсерторы.
67. Перегрузка операций ввода-вывода. Экстракторы.
68. Дружественные функции-операции.
69. Ссылки.
70. Использование ссылочных переменных для перегрузки унарных операций.
71. Виртуальные функции.
72. Использование виртуальных функций.
73. Чистые виртуальные функции.
74. Абстрактные типы.
75. Производные классы и их конструкторы и деструкторы.
76. Конструкторы и деструкторы при множественном наследовании.
77. Виртуальные базовые классы.
78. Операции динамического выделения памяти **new** и **delete**.
79. Виртуальные деструкторы.
80. Шаблоны функций.
81. Шаблоны классов.
82. Статические члены класса.
83. Локальные классы.
84. Вложенные классы.

5.3. Шкалы оценки образовательных достижений

Рейтинговая оценка знаний является интегральным показателем качества теоретических и практических знаний и навыков студентов по дисциплине и складывается из оценок, полученных в ходе текущего контроля и промежуточной аттестации.

Результаты текущего контроля и промежуточной аттестации подводятся по шкале балльно-рейтинговой системы.

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	Оценка по 4-ех балльной шкале	Оценка ECTS	Требования к уровню освоению учебной дисциплины
90-100	5 – <i>«отлично»</i>	A	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.
85-89	4 – <i>«хорошо»</i>	B	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		C	
70-74		D	
65-69		E	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.
60-64	3 – <i>«удовлетворительно»</i>		
Ниже 60	2 – <i>«неудовлетворительно»</i>	F	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

6.УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА:

1. Литвиненко Н. А. - Технология программирования на С++ .Год издания: 2010
Издательство: БХВ-Петербург
2. Романов Е.Л. Си++. От дилетанта до профессионала Год издания: 2014
3. Макс Шлее - Qt4.5 Профессиональное программирование на С++. Год издания:2010
4. Стили и методы программирования: курс лекций: учебное пособие
Автор: Непейвода Н.Н.Издательство: Интернет-Университет Информационных Технологий, 2005 г.
5. Введение в программирование: Учебное пособие Авторы: Баженова И.Ю., Сухомлин В.А. Издательство: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007 г.
6. Программирование на языке С++ Автор: Павловская Т.А. Издательство: Интернет-Университет Информационных Технологий, 2010 г.
7. Шлее М. Qt4.5 Профессиональное программирование на С++. СПб.: БХВ-Петербург, 2010.
8. Травова Н.Н. Лабораторный практикум по Qt часть 2(размещен на Server-129 в папке public/Travova/Qt).
9. Бланшет Ж., Саммерфилд М. Qt4. Программирование GUIна С++. 2-е изд. М.: КУДИЦ-Пресс, 2007.
10. Земсков Ю. Qt4 на примерах. СПб.: БХВ-Петербург, 2008.
11. Шилдт Г. Полный справочник по С++. 4-е изд. М.: Вильямс, 2006
12. Стефенс Д. Р. С++. Сборник рецептов. М.: КУДИЦ-Пресс, 2007.
13. Саммерфилд М. Qt4. Профессиональное программирование. Разработка кросс-платформенных приложений на С++. СПб.: Символ-Плюс 2011.
14. Дж.Макконнелл Анализ алгоритмов. Вводный курс. М.: ТЕХНОСФЕРА, 2002.
15. Буч Г. Объектно-ориентированный анализ и проектирование. – 2-е изд. – М.: БИНОМ, 1998.
16. Страуструп Б. Язык программирования С++. – 3-е изд. – М.: БИНОМ, 1999.
17. Секунов Н. Visual С++.NET. – СПб.: БХВ-Петербург, 2002.
18. Топп У., Форд У. Структуры данных в С++. – М.: Изд. дом «Вильямс», 2000.
19. Майерс С. Эффективное использование STL. – СПб.: ПИТЕР, 2002.
20. Александреску А. Современное проектирование на С++. Обобщенное программирование и прикладные шаблоны проектирования. – М.: Изд. дом «Вильямс», 2002.

21. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: ПИТЕР, 2003.
22. Влиссидес Дж. Применение паттернов проектирования. Дополнительные штрихи. – М.: Изд. дом «Вильямс», 2003.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

1. Н. Вирт. Алгоритмы и структуры данных. СПб.: Невский Диалект, 2001
2. Э. Йодан. Структурное программирование и конструирование программ. М.: Мир, 1979
3. Т. Кормен, Ч. Лейзерсон, Р. Ривест. Алгоритмы: Построение и анализ. – СПб, 2003
4. Д. Э. Кнут. Искусство программирования: Т. 1. Основные алгоритмы. – М.: Вильямс, 2000
5. Д. Э. Кнут. Искусство программирования: Т. 2. Получисленные алгоритмы. М.: Вильямс, 2000
6. Д. Э. Кнут. Искусство программирования: Т. 3. Сортировка и поиск. – М.: Вильямс, 2000
7. Г. С. Уоррен. Алгоритмические трюки для программистов. – СПб, 2003
8. И. В. Виленкин. Высшая математика для студентов экономических, технических, естественно-научных специальностей вузов. Учеб. пособие. Ростов н/Д: Финикс, 2002 г.
9. Дж. Коплиен. Программирование на C++. Классика CS. – СПб.: Питер, 2005
10. Р. Вайнер, Лю Пинсон. C++ изнутри. Пер. с англ. – Киев: «ДиаСофт», 1993
11. М. Эллис, Б. Строуструп. Справочное руководство по языку программирования C++ с комментариями. Пер. с англ. Мю: Мир, 1992
12. А. Мешков, Ю. Тихомиров. Visual C++ и MFC. Программирование для Windows NT и Window 95. Т. 1. – СПб.: BHV – Санкт-Петербург, 1997
13. К. Паппас, У. Мюррей. Visual C++. Руководство для профессионалов: пер. с англ. – СПб.: BHV – Санкт-Петербург, 1996

Учебно-методический материал в библиотеке института, ресурсы Интернета, ресурсы электронной библиотеки.

Электронный ресурс: intuit.ru (Интернет университет информационных технологий)

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Класс ПЭВМ не ниже Intel Pentium 4, 512M RAM, 40G HDD с установленным программным обеспечением: MS WindowsXP, MS Office Pro, Microsoft Visual Studio 13 и др. из расчета одна ПЭВМ на одного человека. Класс ПЭВМ не ниже Intel Pentium 4, 512M RAM, 40G HDD с установленным программным обеспечением: OCLinux с графической оболочкой GNOM, QtCreator.

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При чтении лекционного материала используется электронное сопровождение курса: справочно-иллюстративный материал воспроизводится и озвучивается в аудитории с использованием проектора и переносного компьютера в реальном времени.

На сайте кафедры также находится методический и справочный материал, необходимый для проведения лабораторных работ по курсу.

В соответствии с требованиями ОС ВО по направлению подготовки 03.03.01 «Прикладные математика и физика» реализация компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков студентов. В рамках учебного курса студенты работают с лекциями, рекомендованной литературой, выполняют лабораторные работы, готовятся к экзамену. В процессе подготовки студенты используют программные продукты, инструментальные среды, информационно-справочные системы, информационные источники, размещенные в сети Интернет (официальные сайты, веб-порталы, тематические форумы и телекоммуникации), электронные учебники и учебно-методические пособия.

9.МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОРГАНИЗАЦИИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

- Самостоятельно прорабатывать лекционный материал для более полного усвоения материала;
- В учебном процессе при выполнении лабораторного практикума эффективно использовать методические пособия и методический материал по темам лабораторных работ;
- Активно использовать Интернет-ресурсы для получения актуального материала по изучаемой дисциплине;
- Активно использовать Интернет-ресурсы для обновления инструментальной базы (систем программирования, инструментальных сред и т.д.) при выполнении лабораторных работ.

Программа составлена в соответствии с требованиями ОС ВО НИЯУ МИФИ к обязательному минимуму содержания основной образовательной программы по направлению подготовки 03.03.01 «Прикладные математика и физика»

Автор _____ В.С. Холушкин

Рецензент _____ Н.В. Фролова

Согласовано:

Зав. кафедрой ВИТ _____ В.С. Холушкин

Руководитель ОП _____ Ф.А. Стариков