

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Щёкина Вера Витальевна
Должность: Ректор
Дата подписания: 15.09.2024 10:44:21
Уникальный программный идентификатор:
a2232a55157e576557a8999b1190892a55398947b47b1b0b571a474e577789



**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Благовещенский государственный педагогический университет»

**ОСНОВНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
Рабочая программа дисциплины**

УТВЕРЖДАЮ

декан факультета физико-математи-
ческого образования и технологии
ФГБОУ ВО БГПУ

Н.В. Слесаренко
«03» сентября 2024 г.

**Рабочая программа дисциплины
ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ**

Направление подготовки

**02.03.03 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И
АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

Профиль

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

**Уровень высшего образования
БАКАЛАВРИАТ**

**Принята
на заседании кафедры информатики
и методики преподавания информатики
(протокол № 8 от «25» мая 2024 г.)**

Благовещенск 2024

СОДЕРЖАНИЕ

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ	4
3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)	5
4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ	6
5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ	8
6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА.....	8
7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ	15
В ПРОЦЕССЕ ОБУЧЕНИЯ	15
8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	16
9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ	16
10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА	16
11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ	18

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Цель дисциплины: обеспечение базы теоретической и практической подготовки в области параллельного программирования, развитие мышления, связанного с параллельными вычислениями, приобретение знаний, необходимых для изучения последующих дисциплин.

1.2 Место дисциплины в структуре ООП: Дисциплина «Параллельное программирование» относится к дисциплинам обязательной части (части, формируемой участниками образовательных отношений) блока Б1 (Б1.О.36).

Для освоения дисциплины «Параллельное программирование» используются знания, умения и виды деятельности, сформированные в процессе изучения дисциплин «Программирование», «Структуры и алгоритмы компьютерной обработки данных», «Современные технологии программирования», «Технология программирования Java». Дисциплина «Параллельное программирование» в профессиональной подготовке выпускника обеспечивает раскрытие общего круга вопросов разработки программного обеспечения.

1.3 Дисциплина направлена на формирование следующих компетенций: ОПК-3, ПК-7:

- **ОПК-3.** Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения; индикаторами достижения которой является:

- ОПК-3.1. **знает** основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов;

- ОПК-3.2. **умеет** использовать их в профессиональной деятельности;

- ОПК-3.3. **имеет** практические навыки разработки программного обеспечения.

- **ПК-7.** Способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений; индикаторами достижения которой является:

- ПК-7.1. **знает** основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений;

- ПК-7.2. **умеет** программировать в рамках этих направлений;

- ПК-7.3. **имеет** практический опыт разработки программ в рамках этих направлений.

1.4 Перечень планируемых результатов обучения. В результате изучения дисциплины студент должен

- **знать:**

- понятие рекурсивной функции и понятие алгоритма;
- представление алгоритма;
- систему взаимодействующих процессов, средства задания управления взаимодействующими процессами;

- программирование взаимодействующих процессов;

- модель асинхронной программы;

- анализ и отладку параллельных программ;

- распараллеливание численных алгоритмов и методов;

- принципы сборочной технологии параллельного программирования;

- уметь:

- разрабатывать параллельные алгоритмы и программы для решения разного класса задач на компьютерах с распределенной памятью и общей памятью;
- использовать библиотеки для параллельных программ;
- использовать профилировщики и отладчики для параллельных программ;
- ставить и решать задачи, возникающие в процессе конструирования параллельных программ и эксплуатации системных программных средств;

- владеть:

- теоретическими знаниями в области организации взаимодействующих процессов;
- средствами параллельного программирования Java Thred, Java Concurrent и Erlang;
- способами проверки правильности параллельных программ.

1.5 Общая трудоемкость дисциплины «Параллельное программирование» составляет 3 зачетные единицы (далее – ЗЕ) (108 часов):

Программа предусматривает изучение материала на лекциях и лабораторных занятиях. Предусмотрена самостоятельная работа студентов по темам и разделам. Проверка знаний осуществляется фронтально, индивидуально.

1.6 Объем дисциплины и виды учебной деятельности

Объем дисциплины и виды учебной деятельности (очная форма обучения)

Вид учебной работы	Всего часов	Семестр 7
Общая трудоемкость	108	108
Аудиторные занятия	60	60
Лекции	24	24
Лабораторные занятия	36	36
Самостоятельная работа	48	48
Вид итогового контроля		зачет

2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

2.1 Очная форма обучения

Учебно-тематический план

№	Наименование тем (разделов)	Всего часов	Аудиторные занятия		Самостоятельная работа
			Лекции	Лабораторные занятия	
1.	Тема 1. Введение. История и современные технологии параллельного программирования.	4	2	0	2
2.	Тема 2. Задача конструирования параллельной программы.	8	2	2	4
3.	Тема 3. Программирование взаимодействующих процессов.	14	4	4	6

4.	Тема 4. Низкоуровневые технологии программирования Java.	20	4	8	8
5.	Тема 5. Высокоуровневые технологии параллельного программирования Java. Пакет java util Concurrent.	20	6	4	10
6.	Тема 6. Основы языка параллельного программирования Erlang.	26	4	12	10
7.	Тема 7. Гибридные вычислительные системы с использованием технологии CUDA.	16	2	6	8
Зачет					
ИТОГО		108	24	36	48

Интерактивное обучение по дисциплине

№	Наименование тем (разделов)	Вид занятия	Форма интерактивного занятия	Кол-во часов
1.	История и современные технологии параллельного программирования.	ЛЕК	Лекция-дискуссия	2
2.	Низкоуровневые технологии программирования Java.	ЛБ	Работа в малых группах	8
3.	Высокоуровневые технологии параллельного программирования Java. Пакет java util Concurrent.	ЛБ	Выполнение проекта.	10
ИТОГО				20

3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)

Тема 1. Введение. История и современные технологии параллельного программирования.

История создания первых суперкомпьютеров. Появление алгоритмов и языков параллельного программирования. Сферы применения параллельных вычислительных систем. Современные суперкомпьютеры топ 500, тест LINPACK. Классификация параллельных компьютеров и систем. Классификация Флинна (SISD, SIMD, MISD, MIMD) и другие классификации.

Тема 2. Задача конструирования параллельной программы.

Распределенные вычисления. Коллективные взаимодействия процессов; глобальные операции редукции; выделенные группы процессов, объединяемые коммуникационными связями; оценка время емкости операций передачи данных на кластерных системах. Вычисления над общим полем памяти. Области параллельных вычислений. Параллельные алгоритмы для решения простых типовых задач на системах с общей памятью. Глобальная и локальная память потоков. Критические секции. Синхронизация потоков. Редуцированные операции потоков.

Тема 3. Программирование взаимодействующих процессов.

Параллельные алгоритмы матричных умножений на системах с распределенной памятью. Параллелизм по данным (геометрический параллелизм). Ускорение и эффективность параллельных алгоритмов. Масштабируемость алгоритмов. Параллельные методы решения систем линейных алгебраических уравнений итерационными методами на системах с распределенной памятью. Статическая балансировка загрузки процессоров.

Тема 4. Низкоуровневые технологии программирования Java.

Низкоуровневые потоки, класс Thread и его методы: sleep(), yield(), join(). Флаг прерывания класса Thread. Методы для работы с прерываниями. InterruptedException. Пример реализации. Синхронизация потоков. Взаимоблокировки. Методы объектов wait(), notify(), notifyall().

Тема 5. Высокоуровневые технологии параллельного программирования Java. Пакет java.util.concurrent.

Пакет параллельного программирования java.util.concurrent. Набор коллекция для работы в многопоточной среде, неблокирующие и блокирующие очереди с поддержкой многопоточности. Их отличие от стандартных классов. Фреймворки для создания пула потоков (Executors), механизмы синхронизации (Locks), классы с поддержкой атомарных операций над примитивами и ссылками (Atomic).

Тема 6. Основы языка параллельного программирования Erlang.

Язык программирования Erlang, его характеристика. Типизация, парадигма, компиляция и выполнение программ. Сравнение с другими языками. Горячая замена кода. Типы данных. Операции. Встроенные (BIFs) функции. Модули и функции. Препроцессор и макросы. Переменные, сопоставление с образцом. Процессы. Создание процессов, отправка и прием сообщений. Обработка ошибок и завершение процессов. Работа с файлами. Программирование с сокетами. UDP, TCP. Клиент серверная архитектура. Пример простого сервера. Понятие OTP.

Тема 7. Гибридные вычислительные системы с использованием технологии CUDA.

Параллельные вычисления на GPU. Архитектура и программная модель CUDA. Основы работы с технологией CUDA. Вычисления на графических процессорах (GPU) в задачах математической и теоретической физики.

4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

4.1 Общие методические рекомендации

Дисциплина включает 36 часов лабораторных занятий. Перед выполнением работы следует изучить теоретический материал и ответить на контрольные вопросы. К зачету предъявляется с результаты работы, сохраненные в системе контроля версий. Лабораторные работы могут выполняться во время аудиторных занятий в компьютерном классе и самостоятельно.

4.2 Методические рекомендации по подготовке к лекциям

Курс лекций строится на основе четких понятий и формулировок, так, как только при таком подходе студенты приобретают культуру абстрактного мышления, необходимую для высококвалифицированного бакалавра в любой отрасли знаний. Изложение материала

должно быть по возможности простым и базироваться на уровне разумной строгости. Изложение теоретического материала дисциплины должно предшествовать лабораторным занятиям.

4.3 Методические рекомендации по подготовке к лабораторным занятиям

Цель лабораторного практикума – самостоятельная разработка студентами объектно-ориентированной системы с использованием основ современной программной инженерии.

Целью практических занятий является закрепление теоретического материала лекций и выработка умения использования информационных и других ресурсов, предоставляемых университетом.

Для подготовки к занятиям необходимо познакомиться с ресурсами внешнего сайта ФГБОУ ВО БГПУ: <http://www.bgpu.ru/index.jsp>, с ресурсами внутреннего сайта ФГБОУ ВО БГПУ: <http://iskander.bgpu.ru>, с системой электронного обучения вуза <http://moodle.bgpu.ru>.

Для проведения практических занятий используются компьютеры, оснащенные ОС Windows 7 и выше или Linux. Возможно использование проектора или интерактивной доски.

Одной из форм интерактивных форм проведения занятия является дискуссия.

Важно предварительно определить правила ведения дискуссии, а также критерии оценки выступлений ее участников. Лучше всего это сделать предметом обсуждения в группе, а не предлагать преподавателем в готовом виде. Например, обсудить и принять следующие или похожие правила ведения дискуссии:

- не допускать выпадов против личности;
- не допускать излишнюю эмоциональность;
- высказываться четко, кратко и по теме обсуждения;
- выбрать ведущего дискуссии, ответственного за время и правила ее проведения;
- дать высказаться всем желающим, уважительно относиться к любой точке зрения;
- внимательно слушать друг друга, не говорить одновременно;
- постараться проанализировать разные точки зрения;
- не повторяться, продвигать дискуссию дальше через движение новых идей, сообщение новой информации.

Обсудить и принять в группе следующие критерии оценки участия в дискуссии:

- точность аргументов (причинно-следственные связи);
- четкость и понятность аргументации;
- точность контраргументов (причинно-следственные связи);
- четкость понятность контраргументов;
- логичность;
- удачная подача;
- умение выделить главное;
- отделение фактов от субъективного мнения;
- использование ярких поддерживающих фактов;
- видение сути проблемы;
- ориентация меняющейся ситуации.

4.4 Методические указания к самостоятельной работе студентов

Для успешного усвоения дисциплины необходима правильная организация самостоятельной работы студентов. Эта работа должна содержать:

- регулярную (еженедельную) проработку теоретического материала по конспектам лекций и учебникам;
- регулярную (еженедельную) подготовку к лабораторным занятиям, в том числе изучение описания лабораторных работ.

4.5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине:

1. Фонд оценочных средств.
2. Вопросы к экзамену.
3. Список литературы и информационных ресурсов.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

№	Наименование раздела (темы)	Формы/виды самостоятельной работы	Количество часов
1.	Тема 1. Введение. История и современные технологии параллельного программирования.	Проработка теоретического материала по конспектам лекций. Чтение рекомендованной литературы.	2
2.	Тема 2. Задача конструирования параллельной программы.	Проработка теоретического материала по конспектам лекций. Чтение рекомендованной литературы.	4
3.	Тема 3. Программирование взаимодействующих процессов.	Проработка теоретического материала по конспектам лекций. Чтение рекомендованной литературы.	6
4.	Тема 4. Низкоуровневые технологии программирования Java.	Проработка теоретического материала по конспектам лекций. Чтение рекомендованной литературы.	8
5.	Тема 5. Высокоуровневые технологии параллельного программирования Java. Пакет <code>java.util.Concurrent</code> .	Проработка теоретического материала по конспектам лекций. Чтение рекомендованной литературы.	12
6.	Тема 6. Основы языка параллельного программирования Erlang.	Проработка теоретического материала по конспектам лекций. Чтение рекомендованной литературы.	8
7.	Тема 7. Гибридные вычислительные системы с использованием технологии CUDA.	Проработка теоретического материала по конспектам лекций. Чтение рекомендованной литературы.	8
Итого			48

5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ

5.1 План проведения лабораторных занятий по дисциплине

Задача конструирования параллельной программы.

Лабораторная работа №1 «Класс Thread, параллельные потоки Java»: 2 часа.

Программирование взаимодействующих процессов.

Лабораторная работа №2, 3 «Работа с параллельными коллекциями `java.util.Concurrent`»: 4 часа.

Низкоуровневые технологии программирования Java.

Лабораторная работа №4 «Низкоуровневая синхронизация потоков в Java»: 4 часа.

Лабораторная работа №5 «Пулы потоков java.util.concurrent»: 4 часа.

Высокоуровневые технологии параллельного программирования Java. Пакет java.util.concurrent.

Лабораторная работа №6 «Решение систем линейных алгебраических уравнений методами параллельного программирования»: 4 часа.

Основы языка параллельного программирования Erlang.

Лабораторная работа №7 «Синхронизация потоков при помощи пакета java.util.concurrent»: 4 часа.

Лабораторная работа №8 «Последовательное программирование Erlang»: 2 часа.

Лабораторная работа №9 «Потоки в Erlang, работа с бинарными данными.»: 4 часа.

Лабораторная работа №10 «Передача сообщений между процессами в Erlang»: 2 часа.

Гибридные вычислительные системы с использованием технологии CUDA.

Лабораторная работа №11 «Основы работы с технологией CUDA»: 6 часа.

6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА

6.1 Оценочные средства, показатели и критерии оценивания компетенций

Индекс компетенции	Оценочное средство	Показатели оценивания	Критерии оценивания сформированности компетенций
ОПК-3, ПК-7	Собеседование	Низкий (неудовлетворительно)	Студент отвечает неправильно, нечетко и неубедительно, дает неверные формулировки, в ответе отсутствует какое-либо представление о вопросе
		Пороговый (удовлетворительно)	Студент отвечает неконкретно, слабо аргументировано и не убедительно, хотя и имеется какое-то представление о вопросе
		Базовый (хорошо)	Студент отвечает в целом правильно, но недостаточно полно, четко и убедительно
		Высокий (отлично)	Ставится, если продемонстрированы знание вопроса и самостоятельность мышления, ответ соответствует требованиям правильности, полноты и аргументированности.
ОПК-3, ПК-7	Разноуровневые задачи и задания	Низкий (неудовлетворительно)	Ответ студенту не зачитывается если: - Задание выполнено менее, чем на половину; - Студент обнаруживает незнание большей части соответствующего материала, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно излагает материал.

		Пороговый (удовлетворительно)	Задание выполнено более, чем на половину. Студент обнаруживает знание и понимание основных положений задания, но: • Излагает материал неполно и допускает неточности в определении понятий; • Не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; • Излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
		Базовый (хорошо)	Задание в основном выполнено. Ответы правильные, но: • В ответе допущены малозначительные ошибки и недостаточно полно раскрыто содержание вопроса; • Не приведены иллюстрирующие примеры, недостаточно чётко выражено обобщающее мнение студента; • Допущено 1-2 недочёта в последовательности и языковом оформлении излагаемого.
		Высокий (отлично)	Задание выполнено в максимальном объёме. Ответы полные и правильные. • Студент полно излагает материал, даёт правильное определение основных понятий; • Обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры; • Излагает материал последовательно и правильно с точки зрения норм литературного языка.
ОПК-3, ПК-7	Доклад, сообщение	Низкий (неудовлетворительно)	Доклад студенту не зачитывается если: • Студент не усвоил значительной части проблемы; • Допускает существенные ошибки и неточности при рассмотрении ее; • Испытывает трудности в практическом применении знаний; • Не может аргументировать научные положения; • Не формулирует выводов и обобщений; • Не владеет понятийным аппаратом.
		Пороговый (удовлетворительно)	Задание выполнено более чем на половину. Студент обнаруживает знание и понимание основных положений задания, но:

			• Тема раскрыта недостаточно четко и полно, то есть студент освоил проблему, по существу излагает ее, опираясь на знания только основной литературы; • Допускает несущественные ошибки и неточности; • Испытывает затруднения в практическом применении полученных знаний; • Слабо аргументирует научные положения; • Затрудняется в формулировании выводов и обобщений; • Частично владеет системой понятий.
		Базовый (хорошо)	Задание в основном выполнено: • Студент твердо усвоил тему, грамотно и по существу излагает ее, опираясь на знания основной литературы; • Не допускает существенных неточностей; • Увязывает усвоенные знания с практической деятельностью; • Аргументирует научные положения; • Делает выводы и обобщения; • Владеет системой основных понятий.
		Высокий (отлично)	Задание выполнено в максимальном объеме. • Студент глубоко и всесторонне усвоил проблему; • Уверенно, логично, последовательно и грамотно его излагает; • Опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью; • Умело обосновывает и аргументирует выдвигаемые им идеи; • Делает выводы и обобщения; • Свободно владеет понятиями.

6.2 Промежуточная аттестация студентов по дисциплине

Промежуточная аттестация является проверкой всех знаний, навыков и умений студентов, приобретённых в процессе изучения дисциплины. Формой промежуточной аттестации по дисциплине является зачет.

Для оценивания результатов освоения дисциплины применяется следующие критерии оценивания.

Критерии оценивания устного ответа на зачете

1. Оценка «зачтено» выставляется, если студент ответил на все вопросы билета, но:
 - в ответе допустил более 5 недочетов;
 - в ответе допустил более 3 негрубых ошибок;

- в ответе допустил 1-3 грубых ошибки.
- 2. Оценка «незачтено» выставляется, если:
 - студент ответил на все теоретические вопросы билета, но в ответе допустил более 3 грубых ошибок;
 - студент ответил на 2 теоретических вопроса билета, но в ответе допустил грубые ошибки, при этом к решению задачи не приступал, т.е. не объявил необходимые для решения задачи данные и с грубыми ошибками написал управляющие структуры, необходимые для обработки исходных данных, либо использовал не ту управляющую структуру.
 - студент ответил на менее чем 2 вопроса билета.

Оценка «зачтено» может быть выставлена, если в своем ответе студент показал знание монографической и учебной литературы, выходящей за рамки дисциплины, проявил оригинальность и высокую культуру абстрактного мышления.

К грубым ошибкам относятся:

- незнание основных понятий и управляющих структур, алгоритмов обработки данных;
- полное отсутствие детальной проработки излагаемого при наличии основных этапов изложения;
- другие ошибки, равнозначные указанным.

К негрубым ошибкам относятся:

- частичное отсутствие детальной проработки излагаемого при наличии основных этапов изложения;

К недочетам относятся:

- неточности в пояснениях.

6.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов освоения дисциплины

Примеры лабораторных работ с разноуровневыми заданиями

Задача 1 (общее) Низкоуровневые технологии программирования Java.

Написать программу, вычисляющую в $n = 1..10$ потоков интеграл функции $f(x)$ на заданном участке $[a, b]$, с заданной шагом h методом трапеций.

Значения: $f(x)$, a , b , h - заданы для каждого варианта. Программа должна выводить статистику по времени, затраченному для вычисления с заданным количеством потоков.

Пример:

1. 4 - 2375.4 мс.
2. 5 - 2385.4 мс.
3. 6 - 2432.4 мс.
4. ...
5. 10 - 4896.8 мс.

Варианты:

1. $f(x) = \sin(x)$, $[0, \pi]$, $h = 10E-6$
2. $f(x) = \cos(x)$, $[0, \pi]$, $h = 10E-6$
3. $f(x) = \cos(x) + \sin(x)$, $[0, \pi]$, $h = 10E-6$
4. $f(x) = \exp(x)$, $[0, 5]$, $h = 10E-6$
5. $f(x) = \exp(x) + x$, $[0, 5]$, $h = 10E-6$
6. $f(x) = \ln(x)$, $[0, 2]$, $h = 10E-6$
7. $f(x) = \sqrt{x}$, $[\pi, 2\pi]$, $h = 10E-7$

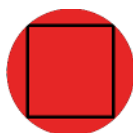
8. $f(x) = 1/\sqrt{x}$, $[\pi, 2\pi]$, $h = 10E-7$
9. $f(x) = \cos(x)$, $[-2\pi, 2\pi]$, $h = 10E-6$
10. $f(x) = \sin(x)+1$, $[5\pi, 7\pi]$, $h = 10E-6$
11. $f(x) = \sqrt{|\sin(x)|}$, $[0, \pi]$, $h = 10E-6$
12. $f(x) = \sqrt{|\cos(x)|}$, $[0, \pi]$, $h = 10E-7$
13. $f(x) = 1/(\pi+\sin(x))$, $[-\pi, \pi]$, $h = 10E-6$
14. $f(x) = \cos(x)*\sin(x)$, $[-\pi, \pi]$, $h = 10E-6$
15. $f(x) = \cos(x)*x$, $[-2\pi, \pi]$, $h = 10E-6$

Задача 2 (повышенная сложность)

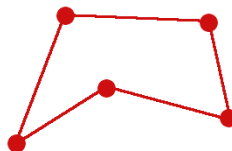
Написать программу, отображающую указанные анимированные графические объекты. Обработка анимации каждого объекта должна происходить в отдельном потоке. Объекты и их потоки можно добавлять, удалять и ставить на паузу.

Примеры объектов:

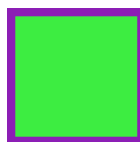
Круг с вписанным n -угольником. Цвет круга и n - устанавливается в зависимости от варианта. n -угольник должен вращаться, скорость и направление вращения задаётся случайным образом при создании объекта. Также случайным образом задается скорость и направление перемещения круга в области отображения. Круг отталкивается от стенок области отображения.



Полигон состоящий из n узлов и линий их соединяющих. n - устанавливается в зависимости от варианта. Полигон обрабатывается одним потоком, но все его узлы должны двигаться независимо. Скорость и направление движения узлов задаются случайно при его создании и для каждого в отдельности. Каждый узел полигона отталкивается от стенок области отображения.



Квадрат с рамкой изменяющей свой цвет. Цвет квадрата устанавливается в зависимости от варианта. Размер, скорость и направление перемещения задаются случайным образом при создании объекта. Квадрат отталкивается от стенок области отображения.



Варианты:

1. круг (синий, 3) полигон (3) квадрат (розовый)
2. круг (красный, 4) полигон (2) квадрат (красный)
3. круг (синий, 5) полигон (4) квадрат (зеленый)
4. круг (красный, 3) полигон (4) квадрат (зеленый)
5. круг (синий, 4) полигон (3) квадрат (зеленый)
6. круг (синий, 5) полигон (2) квадрат (желтый)
7. круг (красный, 3) полигон (3) квадрат (зеленый)
8. круг (зеленый, 4) полигон (5) квадрат (фиолетовый)
9. круг (зеленый, 6) полигон (3) квадрат (розовый)

10. круг (красный, 4) полигон (5) квадрат (зеленый)
11. круг (синий, 3) полигон (3) квадрат (красный)
12. круг (зеленый, 5) полигон (5) квадрат (фиолетовый)
13. круг (синий, 4) полигон (3) квадрат (зеленый)
14. круг (красный, 3) полигон (4) квадрат (фиолетовый)
15. круг (зеленый, 3) полигон (3) квадрат (желтый)

Примеры проектов для самостоятельного решения

Цель – развитие практических навыков создания параллельных программ.

Задача 1.

Написать программу параллельной разметки изображения.

Дано изображение содержащие области произвольной формы и размера (рис. 1.), необходимо написать программу, заливающую каждую область уникальным цветом. Оптимизировать программу для максимальной производительности.

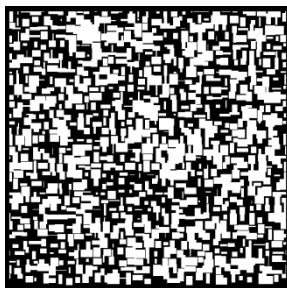


Рис. 1. – Пример исходного изображения

Задача 2.

Реализовать параллельную версию алгоритма векторизации растрового изображения генетическим алгоритмом (рис. 2.).

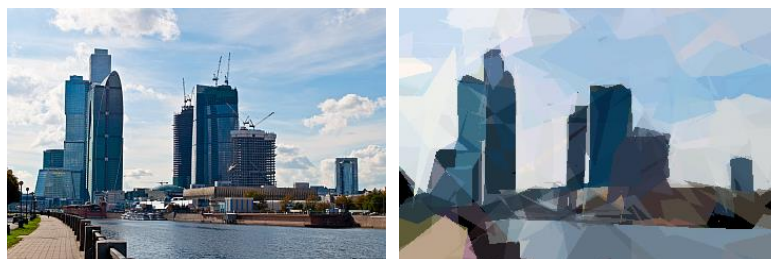


Рис. 2. – Пример работы алгоритма векторизации.

Задача 3.

Написать на языке программирования Erlang программу реализующую алгоритм «Конь Эйлера».

Вопросы к зачету

1. Параллельное программирование и суперкомпьютеры. Области, в которых может возникать потребность в параллельных вычислениях. Особенности параллельных вычислений.

2. Параллельные методы вычислений. Алгоритмы и языки. Современные технологии параллельного программирования.
3. Преобразование последовательных программ в параллельные. Параллельная реализация циклических участков программ. Операции над массивами и множествами.
4. Проблемы создания параллельных программ. Общая память, взаимоблокировки. Стандартные методы синхронизации.
5. Классификация параллельных компьютеров и систем. Классификация Флинна (SISD, SIMD, MISD, MIMD) и другие классификации.
6. Технологии одновременной многопоточности (SSE, Hyper-threading, 3DNow!, MMX)
7. Кластерные, векторно-конвейерные, массивно-параллельные компьютеры и их особенности.
8. Параллельное программирование на Java. Низкоуровневые потоки, класс Thread и его методы: sleep(), yield(), join().
9. Параллельное программирование на Java. Флаг прерывания класса Thread. Методы для работы с прерываниями. InterruptedException. Пример реализации.
10. Параллельное программирование на Java. Синхронизация потоков. Взаимоблокировки. Методы объектов wait(), notify(), notifyall().
11. Пакет параллельного программирования java.util.concurrent. Набор коллекция для работы в многопоточной среде, неблокирующие и блокирующие очереди с поддержкой многопоточности. Их отличие от стандартных классов.
12. Пакет параллельного программирования java.util.concurrent. Фреймворки для создания пула потоков (Executors), механизмы синхронизации (Locks), классы с поддержкой атомарных операций над примитивами и ссылками (Atomic).
13. Язык программирования Erlang, его характеристика. Типизация, парадигма, компиляция и выполнение программ. Сравнение с другими языками. Горячая замена кода.
14. Язык программирования Erlang. Типы данных. Операции. Встроенные (BIFs) функции.
15. Язык программирования Erlang. Модули и функции. Препроцессор и макросы. Переменные, сопоставление с образцом.
16. Язык программирования Erlang. Процессы. Создание процессов, отправка и прием сообщений.
17. Язык программирования Erlang. Обработка ошибок и завершение процессов. Работа с файлами.
18. Язык программирования Erlang. Программирование с сокетами. UDP, TCP.
19. Язык программирования Erlang. Клиент серверная архитектура. Пример простого сервера. Понятие OTP.
20. Параллельные вычисления на графических процессорах. Система распараллеливания CUDA.

7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ В ПРОЦЕССЕ ОБУЧЕНИЯ

Информационные технологии – обучение в электронной образовательной среде с целью расширения доступа к образовательным ресурсам, увеличения контактного взаимодействия с преподавателем, построения индивидуальных траекторий подготовки, объективного контроля и мониторинга знаний студентов.

В образовательном процессе по дисциплине используются следующие информационные технологии, являющиеся компонентами Электронной информационно-образовательной среды БГПУ:

- официальный сайт БГПУ;
- корпоративная сеть БГПУ;

- система электронного обучения ФГБОУ ВО «БГПУ»;
- электронные библиотечные системы;
- мультимедийное сопровождение лекций и практических занятий;
- интегрированная среда разработки программного обеспечения.

8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

При обучении лиц с ограниченными возможностями здоровья применяются адаптивные образовательные технологии в соответствии с условиями, изложенными в раздел «Особенности организации образовательного процесса по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья» основной образовательной программы (использование специальных учебных пособий и дидактических материалов, специальных технических средств обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь и т.п.) с учётом индивидуальных особенностей обучающихся.

9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ

9.1 Литература

1. Богачев, К.Ю. Основы параллельного программирования : учебное пособие / К.Ю. Богачев ; Лаборатория Знаний. – М. : Бином., 2010. – 342 с. (5 экз.)
2. Иванова, Г. С. Технология программирования : учебник для студ. вузов / Г. С. Иванова. – М. : КНОРУС, 2011. – 333 с. (6 экз.)
3. Малявко, А. А. Параллельное программирование на основе технологий openmp, cuda, opencl, mpi : учебное пособие для вузов / А. А. Малявко. – 3-е изд., испр. и доп. – Москва : Издательство Юрайт, 2022. – 135 с. – (Высшее образование). – ISBN 978-5-534-14116-0. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/492127> (дата обращения: 10.10.2022).

9.2 Базы данных и информационно-справочные системы

1. Федеральный портал «Российское образование» – Режим доступа: <http://www.edu.ru>
2. Информационная система «Единое окно доступа к образовательным ресурсам» – Режим доступа: <http://www.window.edu.ru>
3. Федеральный центр информационно-образовательных ресурсов – Режим доступа: <http://fcior.edu.ru>
4. Сайт Федеральной службы по интеллектуальной собственности, патентам и товарным знакам (Роспатента). – Режим доступа: <http://www.fips.ru/rospatent/index.htm>

9.3 Электронно-библиотечные ресурсы

1. ЭБС «Юрайт». – Режим доступа: <https://urait.ru>
2. Полпред (обзор СМИ). – Режим доступа: <https://polpred.com/news>

10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Для проведения лекционных и лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используются аудитории, оснащённые учебной мебелью, аудиторной доской, компьютером с установленным лицензионным специализированным программным обеспечением, с выходом в электронно-библиотечную систему и электронную информационно-образовательную

среду БГПУ, мультимедийными проекторами, экспозиционными экранами, учебно-наглядными пособиями (мультимедийные презентации).

Для проведения лабораторных занятий также используются компьютерные классы, укомплектованные следующим оборудованием:

- Комплект столов письменных.
- Стол преподавателя.
- Аудиторная доска.
- Компьютеры с установленным лицензионным специализированным программным обеспечением.
- Мультимедийный проектор.
- Экспозиционный экран.

Самостоятельная работа студентов организуется в аудиториях оснащенных компьютерной техникой с выходом в электронную информационно-образовательную среду вуза, в специализированных лабораториях по дисциплине, а также в залах доступа в локальную сеть БГПУ.

Лицензионное программное обеспечение: операционные системы семейства Windows, Linux; офисные программы Microsoft office, Libreoffice, OpenOffice; Adobe Photoshop, Matlab, DrWeb antivirus и т.д.

Разработчик: Антонов А.А. – к.ф.-м.н., доцент кафедры информатики и методики преподавания информатики

11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ

Утверждение изменений и дополнений в РПД для реализации в 2025/2026 уч. г.

РПД обсуждена и одобрена для реализации в 2025/2026 уч. г. без изменений на заседании кафедры информатики и методики преподавания информатики (протокол №6 от 26.05.2025 г.).