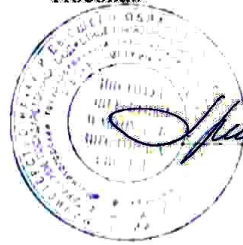




МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И.  
Посова»



УТВЕРЖДАЮ  
Директор ИЭиАС  
В.Р. Храппин

03.02.2026 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**МЕТОДЫ И СРЕДСТВА ВЫСОКОПРОИЗВОДИТЕЛЬНОГО  
ПРОГРАММИРОВАНИЯ**

Направление подготовки (специальность)  
09.04.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы  
Программное обеспечение для цифровизации предприятий и организаций

Уровень высшего образования - магистратура

Форма обучения  
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	1
Семестр	2

Магнитогорск  
2026 год

Рабочая программа составлена на основе ФГОС ВО - магистратура по направлению подготовки 09.04.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 918)

Рабочая программа рассмотрена и одобрена на заседании кафедры  
Вычислительной техники и программирования

29.01.2026, протокол № 7

Зав. кафедрой



О.С. Логунова

Рабочая программа одобрена методической комиссией ИДиАС

03.02.2026 г. протокол № 5

Председатель



В.Р. Храмнин

Рабочая программа составлена:

доцент кафедры кафедры ВТиП, канд. техн. наук



А.П. Калигасв

Рецензент:

директор НИИ «Промбезопасность», д-р техн. наук



М.Ю. Паркевич

## Лист актуализации рабочей программы

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

## 1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Методы и средства высокопроизводительного программирования» являются ознакомление студентов с методами и средствами высокопроизводительного программирования, с основными принципами параллельного программирования, видами современных высокопроизводительных средств вычислительной техники и методиками оценки их производительности.

Для достижения поставленной цели в курсе «Методы и средства высокопроизводительного программирования» решаются задачи:

- изучение методов и средств высокопроизводительного программирования;
- изучение основных принципов параллельного программирования с использованием технологии OpenMP;
- изучение видов современных высокопроизводительных средств вычислительной техники и методик оценки их производительности.

## 2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Методы и средства высокопроизводительного программирования входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Цифровые технологии научных исследований

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Выполнение и защита выпускной квалификационной работы

Технология разработки программного обеспечения

Программное обеспечение для реализации моделей математической физики

## 3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Методы и средства высокопроизводительного программирования» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-7	Обладает способностью к управлению процессом, внутренних правил, методик и регламентов проведения работ по разработке программного обеспечения
ПК-7.1	Оценивает качество управления проведения работ по разработке программного обеспечения



вычислительных систем								
2.1 Отношение стоимость / производительность. Надежность и отказоустойчивость. Масштабируемость. Совместимость и мобильность программного обеспечения.		1			2	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	Устный опрос (собеседование).	ПК-7.1
2.2 Общие замечания. Сравнение компиляторов современных языков программирования высокого уровня и их влияние на производительность ПК.	2	1	2		2	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Проверка индивидуальных заданий.	ПК-7.1
2.3 MIPS. MFLOPS. SPECint92, SPECfp92. SPECrate_int92, SPECrate_fp92. TPC-A, TPC-B, TPC-C. AIM.		1	2		2	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Проверка индивидуальных заданий.	ПК-7.1
Итого по разделу		3	4		6			

3. Методы и средства высокопроизводительного программирования.								
3.1 Основы многопоточного программирования на примере Threads (Win API).	2	1	4		2	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Проверка индивидуальных заданий.	ПК-7.1
3.2 Технологии, ориентированные на кластеры/суперкомпьютеры. Технология MPI.		1			2	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	Устный опрос (собеседование).	ПК-7.1
3.3 Технологии, ориентированные на использование графических процессоров. Технологии NVIDIA CUDA, OpenCL.		1			2	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	Устный опрос (собеседование).	ПК-7.1
3.4 Технологии, ориентированные на многоядерные центральные процессоры. Технологии OpenMP, Intel (Intel Cilk Plus, Intel TBB, Intel ArBB).		1			2	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	Устный опрос (собеседование).	ПК-7.1
Итого по разделу		4	4		8			
4. Параллельное программирование с использованием технологии OpenMP.								
4.1 Базовая модель многопоточного	2	4	5		8	1. Самостоятельно	1. Устный опрос (собеседование).	ПК-7.1

программирования с помощью OpenMP. Структура программы. Последовательные, параллельные и критические секции.						<p>е изучение учебной и научной литературы.</p> <p>2. Работа с электронными библиотеками.</p> <p>3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.</p> <p>4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.</p>	2. Проверка индивидуальных заданий.	
4.2 Разделяемые и локальные потоковые переменные. Барьеры.	2	4	4		5,15	<p>1. Самостоятельное изучение учебной и научной литературы.</p> <p>2. Работа с электронными библиотеками.</p> <p>3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.</p> <p>4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.</p>	<p>1. Устный опрос (собеседование).</p> <p>2. Проверка индивидуальных заданий.</p>	ПК-7.1
Итого по разделу		8	9		13,15			
Итого за семестр		17	17		35,15		экзамен	
Итого по дисциплине		17	17		35,15		экзамен	

## **5 Образовательные технологии**

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающие прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

Формы учебных занятий с использованием технологий проблемного обучения:

Практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.

## **6 Учебно-методическое обеспечение самостоятельной работы обучающихся**

Представлено в приложении 1.

## **7 Оценочные средства для проведения промежуточной аттестации**

Представлены в приложении 2.

## **8 Учебно-методическое и информационное обеспечение дисциплины**

### **а) Основная литература:**

1. Максимов, Н. В. Архитектура ЭВМ и вычислительных систем : учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. — 5-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2024. — 511 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-511-0. - Текст : электронный. - URL:

<https://znanium.ru/catalog/product/2083334> (дата обращения: 21.02.2026). – Режим доступа: по подписке.

2. Сердюк, Ю. П. Параллельное программирование для многоядерных процессоров : краткий учебный курс / Ю. П. Сердюк, А. В. Петров. - Москва : ИНТУИТ, 2016. - 127 с. - Текст : электронный. - URL:

<https://znanium.ru/catalog/product/2155103> (дата обращения: 21.02.2026). – Режим доступа: по подписке.

### **б) Дополнительная литература:**

1. Каропова, Е. Д. Основы многопоточного и параллельного программирования: Учебное пособие / Каропова Е.Д. - Краснояр.:СФУ, 2016. - 356 с.: ISBN 978-5-7638-3385-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/966962> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

2. Федотов, И. Е. Модели параллельного программирования: Практическое пособие / Федотов И.Е. - Москва :СОЛОН-Пр., 2017. - 392 с. (Библиотека профессионала)ISBN 978-5-91359-222-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/858609> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

**в) Методические указания:**

1. Калитаев, А.Н., Егорова, Л.Г., Торчинский, В.Е. Методы и средства высокопроизводительного программирования (практикум): Практикум / Калитаев А.Н., Егорова Л.Г., Торчинский В.Е. - М.: ФГУП НТЦ «Информрегистр», 2022. № гос. рег. 0322203063.

**г) Программное обеспечение и Интернет-ресурсы:****Программное обеспечение**

Наименование ПО	№ договора	Срок действия лицензии
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно
JetBrains IDEA Community Edition	свободно распространяемое ПО	бессрочно

**Профессиональные базы данных и информационные справочные системы**

Название курса	Ссылка
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: <a href="https://elibrary.ru/project_risc.asp">https://elibrary.ru/project_risc.asp</a>
Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: <a href="http://www1.fips.ru/">http://www1.fips.ru/</a>

**9 Материально-техническое обеспечение дисциплины (модуля)**

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.

2. Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ». Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники.

3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.

5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.

6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372.

**Учебно-методическое обеспечение самостоятельной работы обучающихся**

По дисциплине «Методы и средства высокопроизводительного программирования» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает выполнение работ на практических занятиях. Внеаудиторная самостоятельная работа обучающихся осуществляется в виде изучения литературы по соответствующему разделу с проработкой материала при подготовке к сдаче экзамена по данной дисциплине.

Примерные индивидуальные задания (ИДЗ):

**ИДЗ №1. Сравнение компиляторов современных языков программирования высокого уровня и их влияние на производительность ПК.**

Решить задачи на 3 трех языках программирования, оценить качество сделанных программ и провести сравнительную характеристику.

Условие задач:

1. Составить алгоритм и программу, которые вводят границы натуральных  $n$ -значных чисел и из этих чисел печатают только те, цифры которых являются соседними в натуральном ряду. Произвести подсчет напечатанных чисел и определить, сколько в них четных и нечетных.
2. Составить алгоритм и программу, которые в числовую переменную  $N$  вводят большое число, (например,  $N=1000000$ ), и определить все «совершенные» числа. «Совершенным» называется число, равное, сумме всех своих делителей, исключая само число. Например:  $28 = 1+2+4+7+14$ .

*Выполнение работы:* Программы составить на 3 современных языках программирования (например, C++, C#, Object Pascal, Visual Basic, Java).

*Задание:* Оценить качество программ и занести результаты оценки в таблицу:

Компьютер	Язык программирования	Количество найденных чисел	Время выполнения*, сек	
			с учетом оптимизации	без учета оптимизации
<i>n</i> -значность числа	<i>n</i> =	диапазон чисел		
	Object Pascal			
	C++			
	Visual Basic			
	Object Pascal			
	C++			
	Visual Basic			
<i>n</i> -значность числа	<i>n</i> =	диапазон чисел		
	Object Pascal			
	C++			
	Visual Basic			
	Object Pascal			
	C++			
	Visual Basic			

\* - учитывается среднее время выполнения программы по 3 экспериментам, в случае, если время не может быть определено (например, 0 мс), определяем суммарное время за  $m$  – экспериментов (например,  $m = 100$ ).

**ИДЗ №2. Прямое сравнение производительности ПК.**

Для сравнения производительности ПК реализовать тесты на вычисления с вещественными (т.е. дробными) числами и вычисления с целыми числами. Для представления

вещественных чисел можно использовать тип double (8 байт), а для целых – DWORD (unsigned long) (4 байта).

1. Тесты с вычислениями над числами с плавающей точкой.

1.1. Вычисления числа  $\pi$  с помощью ряда  $1 - (1/3) + (1/5) - (1/7) + \dots = (\pi/4)$ .

Компьютер	Время выполнения, с

1.2. Вычисление интеграла функции  $\sin(x)$  при  $0 \leq x \leq \pi$  методом прямоугольников.

Компьютер	Время выполнения, с

2. Тесты с вычислениями над целыми числами. При этом, тесты обрабатывающие массивы данных, необходимо проводить для разных размеров массивов, так как скорость выполнения таких задач зависит не только от вычислительной мощности процессора, но в не меньшей мере и от размера и скорости кэша, скорости памяти, эффективности реализации всей подсистемы памяти в целом.

2.1. Возведение в квадрат единичной матрицы. Чтобы точнее измерить время для небольших матриц, тест выполняет расчеты по несколько раз. Необходимо рассмотреть 3 размерности матрицы, например, 1000x1000 (а); 100x100 (б); 10x10 (в).

Компьютер	Время (а), с	Время (б), с	Время (в), с

2.2. Сортировка псевдослучайной последовательности чисел пузырьковым методом. Необходимо провести операцию для последовательностей разной длины: 50000 (а) элементов; 5000 (б); 100 (в). Чтобы точнее измерить время для небольших последовательностей, тест выполняет расчеты по несколько раз.

Компьютер	Время (а), с	Время (б), с	Время (в), с

2.3. Вычисление суммы всех простых чисел, меньших, чем N, где N, в данном случае равно 50000. Особенность этой задачи состоит в том, что все данные уместятся в регистрах процессора. Таким образом, сравнивается исключительно производительность ядер.

Компьютер	Время выполнения, с

Для сравнения производительности ПК необходимо привести таблицу с «нормированными» значениями. То есть значения буду рассчитывать по формуле

$v = vt \cdot (h/1000)$ , где  $vt$  - время, полученное в тесте,  $h$  - частота процессора. Таким образом, такое значение показывала бы данная система на частоте 1 ГГц, при условии линейной зависимости производительности от частоты.

Компьютер	1.1	1.2	2.1a	2.1б	2.1в	2.2a	2.2б	2.2в	2.3

### ИДЗ №3. Параллельное программирование с использованием WinAPI.

Написать многопоточную программу для вычисления приближенного значения интеграла методами прямоугольников, трапеций и парабол (метод Симпсона). Программа с помощью цикла запускает  $n$  потоков, которые начинают вычислять значение определенного интеграла по заданной формуле, главный поток ожидает завершения вычислений всеми потоками и затем выводит значение вычислений на экран. Потоки, работая параллельно, используют мьютекс для синхронизации вывода в переменную. Параллельные вычисления в потоках производятся с небольшими случайными задержками.

Функция `CreateThread` создает в памяти пространство для запуска процессом еще одной поточной функции.

Функция `CreateMutex` регистрирует в системе объект Мьютекс, нужный для синхронизации параллельных процессов.

Функция `CreateSemaphore` регистрирует в системе объект Семафор, который, также как и мьютекс, помогает синхронизировать параллельные процессы.

Проверить результат вычисления с помощью подпрограммы, состоящей из одного цикла.

Пример приближенного решения интеграла:

$$\int_0^x \sin(x) dx \approx \sum_{k=0}^{n-1} \frac{\pi}{n} * \sin\left(\frac{\pi k}{n}\right).$$

### ИДЗ №4. Параллельное программирование с использованием OpenMP.

*Задание 1. Введение в OpenMP.* Создайте консольное приложение в среде Visual Studio 2010 с поддержкой OpenMP.

*Указания к выполнению задания:*

Для включения поддержки OpenMP установите дополнительные параметры компиляции проекта:

- В главном меню выберите *Project->Имя\_проекта Properties*
- В открывшемся окне выберите *Configuration Properties / C/C++ / Language*. Установите для опции *OpenMP Support* значение *Yes(/openmp)*. Нажмите кнопку *OK*. См. рис.1.

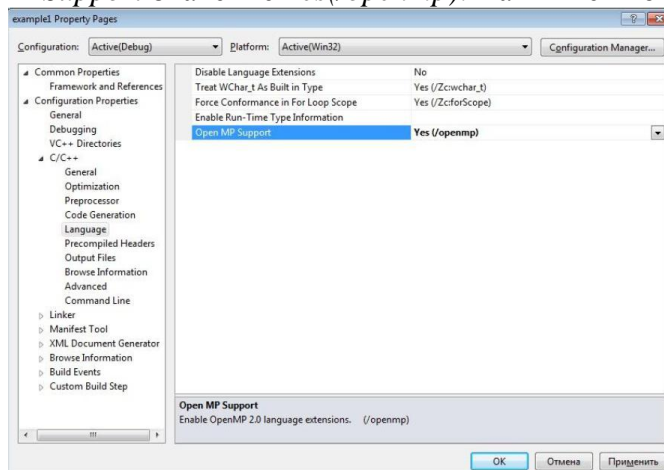


Рис.1. Окно настройки параметров компиляции проекта

**Задание 2.** Напишите программу, в которой создается 4 нити и каждая нить выводит на экран строку "Hello World!".

*Указания к выполнению задания:*

1. Создайте консольное приложение в среде Visual Studio с поддержкой OpenMP.

2. Напишите на языке C/C++ программу, печатающую на экран строку «Hello World!».

3. Подключите заголовочный файл omp.h с функциями и переменными OpenMP.

Строка подключения заголовочного файла:

```
#include <omp.h>
```

4. В функции main создайте параллельную область с помощью OpenMP-директивы parallel. Обратите внимание, что открывающаяся фигурная скобка и название директивы должны находиться в разных строках! Поместите команду вывода строки «Hello World!» внутрь параллельной области.

```
#pragma omp parallel
{
    printf("Hello World!\n");
}
```

5. Задайте количество нитей в параллельной области одним из следующих способов:

*Способ 1.* Вызовите функцию omp\_set\_num\_threads() перед началом параллельной области. В качестве параметра укажите одно целое число – количество нитей в параллельной области:

```
omp_set_num_threads(4);
#pragma omp parallel
{
    printf("Hello World!\n");
}
```

*Способ 2.* Добавьте к директиве parallel параметр num\_threads(). В качестве параметра укажите одно целое число – количество нитей в параллельной области:

```
#pragma omp parallel num_threads(4)
{
    printf("Hello World!\n");
}
```

6. Скомпилируйте и запустите ваше приложение. Убедитесь, что строка «Hello World!» выводится на экран столько раз, сколько нитей вы задали в параллельной области.

**Задание 3.** Использование библиотеки OpenMP для вычисления определенного интеграла функции методами прямоугольников, трапеций, парабол. Выполнить сравнительный анализ результатов работы с результатами, полученными в задании №3.

*Сравнительный анализ результатов*

Метод вычисления значения интеграла функции	Значение и время вычисления интеграла функции $\sin(x)$ ( $0 \leq x \leq \pi$ )			
	средствами WinAPI		средствами OpenMP	
	значение	время	значение	время
Метод прямоугольников				
Метод трапеций				
Метод парабол				

**ИДЗ №5. Распараллеливание вычислений с помощью OpenMP.**

**Задание.** Реализовать и распараллелить с помощью OpenMP заданный алгоритм (согласно варианту).

*Варианты заданий*

№ варианта	Размерность задачи	Тип элемента данных	Алгоритм

1.	120	Длинное целое	Сортировка массива с помощью Шейкер-сортировки
2.	20×60	Знаковый короткий целый	Сортировка столбцов матрицы по возрастанию
3.	70	Беззнаковый целый	Сортировка массива методом пузырька
4.	80	Длинное целое	Сортировка массива методом QuickSort (быстрая сортировка)
5.	50	Целое	Сортировка массива методом Шелла.
6.	90	Беззнаковый целый	Сортировка массива с помощью сортировки вставками
7.	70×90	Длинное целое	Сортировка строк матрицы по возрастанию
8.	80×30	Знаковый короткий целый	Сортировка строк матрицы по убыванию
9.	90×60	Беззнаковый целый	Сортировка столбцов матрицы по убыванию
10.	100×50	Число плавающей запятой	с Перемножение матриц
11.	100	Длинное целое	Сортировка массива с помощью пирамидальной сортировки

### ИДЗ №6. OpenMP. Параллельные методы решения краевых задач.

*Задание.* На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию сеточных алгоритмов решения краевых задач (задача Дирихле для уравнения Пуассона в квадратной области). Сравнить время выполнения и скорость сходимости (т.е. число итераций) параллельного алгоритма с непараллельными алгоритмами для различных значений параметра  $N$  (количество узлов сетки).

Постановка задачи:

Краевая задача Дирихле для уравнения Пуассона:

$$\begin{aligned} \Delta u(x, y) &= f(x, y), \dots (x, y) \in D, \\ u(x, y) &= g(x, y), \dots (x, y) \in \partial D, \end{aligned}$$

Здесь  $D$  – область в  $R^2$ ,  $\partial D$  – ее граница,  $\Delta = \partial^2 / \partial x^2 + \partial^2 / \partial y^2$  – двумерный оператор Лапласа. Функции  $f$  и  $g$  даны, требуется найти функцию  $u$ . Подобная краевая задача возникает при моделировании установившегося течения жидкости, стационарных тепловых полей, процессов теплопередачи с внутренними источниками тепла и т. д.

Для простоты будем рассматривать в качестве области  $D$  единичный квадрат:

$$D = \{(x, y) \in R^2: 0 \leq x \leq 1, 0 \leq y \leq 1\}.$$

Функции  $f$  и  $g$  определим следующим образом:

$$\begin{aligned} (x, y) &= 0, \quad (x, y) \in D, \\ (x, 0) &= 1 - 2x, \\ (0, y) &= 1 - 2y, \\ (x, 1) &= -1 + 2x, \\ (1, y) &= -1 + 2y. \end{aligned}$$

График решения  $u(x, y)$  в данном случае будет представлять собой гиперболический параболоид (седло).

### ИДЗ №7. Решение задач нестационарной теплопроводности с помощью явных и неявных разностных схем.

*Задание.* На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию сеточных алгоритмов решения краевых задач.

*Варианты заданий:*

1. Апельсины в течение короткого времени могут выдерживать отрицательные температуры. Предположим, что апельсин диаметром 0,1 м ( $\lambda=0,47$  Вт/(м·град),  $c=3800$  Дж/(кг·град),  $\rho=940$  кг/м<sup>3</sup>) имеет начальную температуру +5 °С. Температура воздуха внезапно падает до -5 °С. Построить математическую модель для определения момента времени, когда температура поверхности апельсина достигнет 0 °С. Коэффициент теплоотдачи от апельсина к воздуху равен 10 Вт/(м<sup>2</sup>·град). Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование для различной размерности задачи и числа используемых процессов (потоков).
2. Начальная температура хлорвинилового шарика ( $\lambda=0,15$  Вт/(м·град),  $\alpha=8 \cdot 10^{-8}$  м<sup>2</sup>/с) диаметром 5 см равна 90 °С. Он погружается в бак с водой, имеющей температуру 20 °С. Коэффициент теплоотдачи от шарика в воде 20 Вт/(м<sup>2</sup>·град). Построить математическую модель процесса охлаждения шарика. Найти время пребывания шарика в воде, по истечении которого температура в его центре достигнет 40 °С. Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование для различных размеров вычислительной сетки и числа используемых процессов (потоков).
3. Длинный алюминиевый ( $\lambda=236$  Вт/(м·град),  $\alpha=10^{-4}$  м<sup>2</sup>/с) цилиндр диаметром 0,6 м имеет начальную температуру 200 °С. Его внезапно помещают в среду с температурой 70 °С и коэффициентом теплоотдачи 85 Вт/(м<sup>2</sup>·град). Построить математическую модель процесса охлаждения цилиндра. Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование для различных размеров сетки и числа используемых процессов (потоков).
4. Лист оконного стекла имеет толщину 4 мм. Температура одной поверхности 0 °С, а другой – +20 °С. Найти распределение температуры по толщине стекла через заданное время, если температура за окном упала до -10 °С. Записать математическую постановку задачи. Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование для различной плотности узлов вычислительной сетки и числа используемых процессов (потоков).
5. Стенка большой печи толщиной 1,5 см изготовлена из чугуна ( $\lambda=83,5$  Вт/(м·град),  $\alpha=22 \cdot 10^{-6}$  м<sup>2</sup>/с). Температура горячего газа 1100 °С. Коэффициент конвективной теплоотдачи на внутренней поверхности стенки 250 Вт/(м<sup>2</sup>·град). Наружная поверхность печи окружена воздухом с температурой 30 °С и коэффициентом теплоотдачи 20 Вт/(м<sup>2</sup>·град). Записать постановку задачи теплопроводности для стенки и найти распределение температур в ней. Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование для различного числа используемых процессов (потоков).
6. Определить конечное распределение температуры в обручальном кольце с радиусами  $R=2,0$  см и  $r=1,8$  см, если температура в помещении 20 °С, а температуру тела человека можно принять равной 36,6 °С. Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование для различных размеров вычислительной сетки и числа используемых процессов (потоков).  $\alpha=126 \cdot 10^{-6}$  м<sup>2</sup>/с.
7. Бетонный цилиндр диаметром 10 см и длиной 2,5 м имеет начальную температуру 90 °С. Он охлаждается в воздухе при температуре 10 °С. Коэффициент теплоотдачи от бетона к воздуху 18 Вт/(м<sup>2</sup>·град). Найти распределение температур в цилиндре с течением времени. Определить время, за которое температура в центре цилиндра

- достигнет  $30\text{ }^{\circ}\text{C}$ . Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование.
8. Нужно нагреть кусок алюминиевой проволоки, пропуская по ней электрический ток. Диаметр проволоки  $1\text{ мм}$ , длина  $10\text{ см}$ , электрическое сопротивление  $0,2\text{ Ом}$ . По ней пропускается постоянный ток силой  $1\text{ А}$  в течение  $60\text{ с}$ . Начальная температура проволоки  $25\text{ }^{\circ}\text{C}$ . Проволока находится в воздухе с температурой  $25\text{ }^{\circ}\text{C}$  и коэффициентом теплоотдачи  $20\text{ Вт}/(\text{м}^2\cdot\text{град})$ . Найти распределение температуры в проволоке с течением времени. Записать математическую формулировку задачи. Пояснение: объемная интенсивность внутренних источников тепловыделения при пропускании электрического тока = сила тока·сила тока·омическое сопротивление/объем. Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование.
  9. Электродетонатор имеет форму цилиндра диаметром  $0,1\text{ мм}$  и длиной  $5\text{ мм}$ . Он находится в воздухе с температурой  $30\text{ }^{\circ}\text{C}$  и коэффициентом теплоотдачи  $10\text{ Вт}/(\text{м}^2\cdot\text{град})$ . Теплофизические свойства детонатора:  $\lambda=20\text{ Вт}/(\text{м}\cdot\text{град})$ ,  $\alpha =5\cdot 10^{-5}\text{ м}^2/\text{с}$ , электрическое сопротивление  $0,2\text{ Ом}$ . Пренебрегая излучением и утечками тепла в креплениях на концах детонатора, определить время, по истечении которого детонатор взорвется, если по детонатору пропускать постоянный ток силой  $3\text{ А}$ . Пояснение: объемная интенсивность внутренних источников тепловыделения при пропускании электрического тока = сила тока·сила тока·омическое сопротивление/объем. Температура плавления материала детонатора  $900\text{ }^{\circ}\text{C}$ . Записать математическую постановку задачи. Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование для различной размерности задачи и числа используемых процессов (потоков).
  10. Начальная температура длинной стальной балки с сечением квадрата единичной длины равна  $40\text{ }^{\circ}\text{C}$ . Определить распределение температуры тела через  $100\text{ с}$ , если на боковых границах задать следующие граничные условия: при  $X=0:T=20$ ; при  $X=1:T=10$ ; при  $Y=0:T=10$ ; при  $Y=1:T=10$ . Записать математическую постановку задачи. Разработать численный метод решения задачи. Написать параллельную программу и провести ее тестирование.

Оценочные средства для проведения промежуточной аттестации

а) Оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
<b>ПК-7: Обладает способностью к управлению процессом, внутренних правил, методик и регламентов проведения работ по разработке программного обеспечения</b>		
ПК-7.1	Оценивает качество управления проведения работ по разработке программного обеспечения	<p><i>Теоретические вопросы</i></p> <ol style="list-style-type: none"> <li>1. Классификация компьютеров по областям применения. Персональные компьютеры и рабочие станции. X-терминалы. Серверы. Мейнфреймы.</li> <li>2. Современные процессоры. Многопроцессорные системы.</li> <li>3. Системы высокой готовности и отказоустойчивые системы.</li> <li>4. Многопроцессорные системы с общей памятью.</li> <li>5. Многопроцессорные системы с локальной памятью и многомашинные системы.</li> <li>6. Классификация систем параллельной обработки данных. Кластерные решения Sun Microsystems. Высокопроизводительные вычисления на кластерах.</li> <li>7. Понятие – «высокопроизводительные средства вычислительной техники».</li> <li>8. Общие требования, предъявляемые к современным компьютерам.</li> <li>9. Совместимость и мобильность программного обеспечения.</li> <li>10. Оценка производительности вычислительных систем.</li> <li>11. Технологии, ориентированные на кластеры/суперкомпьютеры. Технология MPI.</li> <li>12. Технологии, ориентированные на многоядерные центральные процессоры. Технологии OpenMP, Intel (Intel Cilk Plus, Intel TBB, Intel ArBB).</li> <li>13. Технологии, ориентированные на использование графических процессоров. Технологии NVIDIA CUDA, OpenCL.</li> <li>14. Параллельное программирование с использованием технологии OpenMP.</li> <li>15. Структура программы, использующей средства Win API. Создание, ожидание и завершение работы потоков. Критические секции. Семафоры.</li> </ol>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>16. Алфавитный указатель по директивам, функциям, опциям и переменным окружения OpenMP.</p> <p>17. OpenMP. Параллельные и последовательные области.</p> <p>18. OpenMP. Разработка параллельных алгоритмов и программ для решения задач вычислительной математики.</p> <p>19. Синхронизация в OpenMP.</p> <p><i>Практические задания</i></p> <p>Применять технологии параллельных вычислений для высокопроизводительной реализации решения задач обработки данных:</p> <p><i>Задания:</i></p> <ol style="list-style-type: none"> <li>1. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма, реализующего перемножение двух квадратных матриц <math>A[M][M]</math>.</li> <li>2. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого столбца матрицы <math>A[M][N]</math> по возрастанию.</li> <li>3. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого столбца матрицы <math>A[M][N]</math> по убыванию абсолютных величин.</li> <li>4. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы <math>A[M][N]</math> по возрастанию абсолютных величин.</li> <li>5. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы <math>A[M][N]</math> по убыванию суммы значений цифр элементов матрицы.</li> <li>6. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого столбца матрицы <math>A[M][N]</math> по возрастанию суммы значений цифр элементов матрицы.</li> <li>7. На основе многопоточного распараллеливания (технология OpenMP)</li> </ol>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>выполнить высокопроизводительную реализацию алгоритма сортировки всех нечетных элементов строк матрицы <math>A[M][N]</math> по возрастанию.</p> <p>8. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки всех четных элементов столбцов матрицы <math>A[M][N]</math> по убыванию.</p> <p>9. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого столбца матрицы <math>A[M][N]</math> по возрастанию абсолютных величин.</p> <p>10. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы <math>A[M][N]</math> по убыванию абсолютных величин.</p> <p>11. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой четной строки матрицы <math>A[M][N]</math> по возрастанию, каждого четного столбца по возрастанию.</p> <p>12. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой четной строки матрицы <math>A[M][N]</math> по возрастанию, каждого нечетного столбца по возрастанию абсолютных величин.</p> <p>13. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой нечетной строки матрицы <math>A[M][N]</math> по возрастанию абсолютных величин, каждого четного столбца по возрастанию.</p> <p>14. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы матрицы <math>A[M][N]</math> по возрастанию.</p> <p>15. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого четного столбца матрицы <math>A[M][N]</math> по убыванию, каждой строки по убыванию.</p> <p>16. . На основе многопоточного распараллеливания (технология OpenMP)</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы <math>A[M][N]</math> по возрастанию четных чисел.</p> <p>17. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы <math>A[M][N]</math> по возрастанию, каждого столбца по убыванию.</p> <p>18. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой строки матрицы <math>A[M][N]</math> по возрастанию отрицательных величин.</p> <p>19. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждого четного столбца матрицы <math>A[M][N]</math> по убыванию суммы значений цифр элементов матрицы.</p> <p>20. На основе многопоточного распараллеливания (технология OpenMP) выполнить высокопроизводительную реализацию алгоритма сортировки каждой нечетной строки матрицы <math>A[M][N]</math> по возрастанию суммы значений цифр элементов матрицы.</p> <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>1. Выполнить сравнительный анализ технологий параллельных вычислений при решении задач численного интегрирования.</p> <p>2. Выполнить сравнительный анализ технологий параллельных вычислений при решении краевых задач (задача Дирихле для уравнения Пуассона в квадратной области).</p> <p><i>Тестовые задания:</i></p> <p>1. Какая технология предназначена для создания параллельных программ для систем с распределенной памятью (кластеров)?</p> <p><b>а) MPI</b> б) OpenMP</p> <p>2. Перечислите языки программирования, предназначенные для создания параллельных программ с использованием стандарта OpenMP (несколько</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>вариантов)?</p> <ul style="list-style-type: none"><li>a) Java</li><li><b>б) Fortran</b></li><li>в) C</li><li>г) C++</li><li>д) C#</li><li>е) Visual Basic</li></ul> <p>3. К какому типу директив относится директива parallel стандарта OpenMP?</p> <ul style="list-style-type: none"><li>a) разделение работы</li><li><b>б) определение параллельной области</b></li><li>в) синхронизация</li></ul>

## **б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:**

Промежуточная аттестация по дисциплине «Методы и средства высокопроизводительного программирования» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по данной дисциплине проводится в устной форме по экзаменационным билетам, каждый из которых включает 2 теоретических вопроса и одно практическое задание.

### **Показатели и критерии оценивания экзамена:**

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.