

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Ярославский государственный университет им. П.Г. Демидова»  
Кафедра компьютерных сетей

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 24 » \_\_\_\_\_ мая \_\_\_\_\_ 2022 г.

**Рабочая программа дисциплины**  
«Параллельное программирование»

**Научная специальность**

2.3.5 Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей

**Форма обучения**

очная

Программа рассмотрена  
на заседании кафедры  
от 15 март 2022г.,  
протокол № 8

Ярославль

### 1. Цели освоения дисциплины

Целями дисциплины «Параллельное программирование» являются освоение технологий параллельного программирования, разбор архитектуры параллельных вычислительных систем, ознакомление с принципами распараллеливания программ, получение навыков программирования с использованием технологий OpenMP, MPI, CUDA.

### 2. Место дисциплины в структуре ОП аспирантуры

Дисциплина «Параллельное программирование» относится к вариативной части (дисциплина по выбору) ОП аспирантуры.

Для освоения данной дисциплиной студенты должны обладать знаниями по математике и информатике, в т.ч.: основы программирования на языках С и/или С++, алгоритмов и структур данных, алгебре, геометрии, дифференциальными уравнениями, уравнениям в частных производных и численным методам; а также проявлять настойчивость, целеустремленность и инициативу в процессе обучения.

### 3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП аспирантуры

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

<b>Формулировка компетенции</b>	<b>Перечень планируемых результатов обучения</b>
способностью выполнять теоретические исследования процессов создания, накопления и обработки информации, включая анализ и создание моделей данных и знаний, языков их описания и манипулирования, разработку новых математических методов и средств поддержки интеллектуальной обработки данных	Знать: – архитектуру параллельных вычислительных систем; - области применимости параллельных вычислений; Уметь: – разбивать программу на независимые процессы ; – интерпретировать результаты полученных вычислений; Владеть навыками: – разработки программ с использованием технологий параллельного программирования OpenMP, MPI и CUDA. – обработки полученных вычислений для формирования соответствующих выводов об эффективности применяемых параллельных вычислений.

#### 4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зач. ед., 108 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)	
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа		
			<b>Контактная работа</b>							
1.	Раздел 1. История параллельных вычислительных систем. Параллелизм и его использование	2	2						14	
2.	Раздел 2. Вычислительный кластер. API для управления потоками, их синхронизации и планирования Pthreads	2	2						16	Самостоятельная работа
3.	Раздел 3. Технология программирования OpenMP	2	2						16	Самостоятельная работа
4.	Раздел 4. Технология программирования MPI	2	2						16	Самостоятельная работа
5.	Раздел 5. Введение в технологию CUDA	2	2						16	Самостоятельная работа
6.	Раздел 6. Гибридная модель параллельного программирования	2	2			2			16	
	<b>Всего</b>		<b>12</b>			<b>2</b>			<b>94</b>	<b>Зачет</b>

#### Содержание разделов дисциплины:

Раздел 1. История параллельных вычислительных систем. Параллелизм и его использование.

- 1.1. История развития параллельных вычислительных систем.
- 1.2. Обзор современных вычислительных систем для параллельных вычислений.
- 1.3. Способы параллельной обработки данных.
- 1.4. Компьютеры с общей и распределенной памятью.
- 1.5. Графы информационных зависимостей.
- 1.6. Концепция неограниченного параллелизма.
- 1.7. Крупноблочное распараллеливание.
- 1.8. Низкоуровневое распараллеливание.
- 1.9. Оценка эффективности параллельных вычислений.

Раздел 2. Вычислительный кластер. API для управления потоками, их синхронизации и планирования Pthreads.

- 2.1. Основы работы с гибридным вычислительным кластером.
- 2.2. Стандарт POSIX-реализации потоков (нитей) выполнения: типы данных, функции управления потоками, функции синхронизации потоков.

Раздел 3. Технология программирования OpenMP.

- 3.1. Основные конструкции.
- 3.2. Работа с переменными.
- 3.3. Распараллеливание циклов.
- 3.4. Параллельные секции.
- 3.5. Критические секции.
- 3.6. Атомарные операции.
- 3.7. Операции синхронизации.

Раздел 4. Технология программирования MPI.

- 4.1. Общие функции.
- 4.2. Функции приема/передачи сообщений между процессами.
- 4.3. Функции коллективного взаимодействия процессов, создания пользовательских операций, работа с группами процессов.
- 4.4. Пересылка разнотипных данных, производные типы данных, упаковка данных.

Раздел 5. Введение в технологию CUDA.

- 5.1. Архитектура GPU.
- 5.2. Программная модель CUDA.
- 5.3. Иерархия памяти в CUDA.
- 5.4. Библиотека Thrust.

Раздел 6. Гибридная модель параллельного программирования.

- 6.1. Совместное использование технологий программирования OpenMP, MPI и CUDA.

## **5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине**

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

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

В процессе осуществления образовательного процесса используются:

– для формирования текстов материалов для промежуточной и текущей аттестации, для разработки документов, презентаций, для работы с электронными таблицами - программы OfficeStd 2013 RUS OLP NL Acdmc 021-10232, LibreOffice (свободное), издательская система LaTeX;

- компиляторы с высокоуровневых языков программирования;

- вычислительный гибридный кластер ЯрГУ.

- программные библиотеки OpenMP, MPI и NvidiaCUDA.

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

## **7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

а) основная:

1. Д.С. Глызин, Д.С. Кащенко. Параллельное и функциональное программирование. Методические указания. Яросл. гос. ун-т им. П.Г. Демидова. — Ярославль: ЯрГУ, 2009.

2. Топорков В.В. Модели распределенных вычислений. - М.: Физматлит, 2011 - 320с.

3. Карманов В.Г. Математическое программирование: Учеб. пособие. - 5-е изд., стереотип. - М.: Физматлит, 2005. - 264с.

4. Программирование на языке высокого уровня. Программир. на языке C++: Уч. пос. / Т.И.Немцова и др.; Под ред. Л.Г.Гагариной - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 512 с.

б) дополнительная:

1. Г.Р. Эндрюс. Основы многопоточного, параллельного и распределенного программирования. : Пер. с англ. — М.: Издательский дом «Вильямс», 2003.

2. В.В. Воеводин, Вл.В. Воеводин. Параллельные вычисления. С.-Петербург, "БХВ-Петербург", 2002.

3. А.С. Антонов. Параллельное программирование с использованием технологии OpenMP: учеб. пособие — М.: Изд-во МГУ, 2009.

4. Корнеев В.Д. Параллельное программирование в MPI. — Москва-Ижевск: Институт компьютерных исследований, 2003.

5. Д. Кнут, Искусство программирования, том 1. Основные алгоритмы. : Пер. с англ. — М.: Издательский дом «Вильямс», 2008.

6. Д. Уоткинс, Основы матричных вычислений. Пер. с англ. — М.: БИНОМ. Лаборатория знаний, 2006.

7. Алгоритмизация и программирование : Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 352 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0355-1, 1000 экз.

8. Богачёв, К.Ю. Основы параллельного программирования: учебное пособие [Электронный ресурс] : учебное пособие. - Электрон. дан. - М. : "Лаборатория знаний" (ранее "БИНОМ. Лаборатория знаний"), 2013. - 343 с.

в) ресурсы сети «Интернет»

- CUDAdeveloperzone - <http://developer.nvidia.com/category/zone/cuda-zone>

MPI Documents, user's guide - <http://www.mpi-forum.org/docs/docs.html>

OpenMP Specifications - <http://openmp.org/>

The Message Passing Interface (MPI) standard - <http://www.mcs.anl.gov/mpi/index.html>

Портал по параллельным вычислениям - <http://parallel.ru/>

Суперкомпьютеры - <http://supercomputers.ru/>

**8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Компьютерный класс, оборудованный для проведения лекций и практических занятий и консультаций, фонд библиотеки, компьютерная техника с доступом к вычислительному гибриднему кластеру ЯрГУ.

**Автор(ы) :**

Зав. кафедрой

компьютерных сетей, профессор, д.ф.-м.н. \_\_\_\_\_ С.Д. Глызин

**Приложение №1 к рабочей программе дисциплины  
«Параллельное программирование»  
Фонд оценочных средств  
для проведения текущей и промежуточной аттестации студентов  
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для оценки  
знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы  
формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе текущей  
аттестации**

**Проверка сформированности компетенции выполняется всеми контрольными и  
самостоятельными работами**

**Задания для самостоятельной работы**

*Задания по Разделу 2. Вычислительный кластер. API для управления потоками, их синхронизации и планирования Pthreads.*

Написать программу численного интегрирования функции на отрезке с использованием технологий Pthreads. Функция, отрезок интегрирования и количество потоков вычислений задаются индивидуально.

Построить график зависимости скорости подсчета интеграла от количества потоков.

Критерии оценивая:

Оценка	Критерий
зачтено	(компетенция сформирована не ниже порогового уровня): Написана программа на языке C++ численного интегрирования функции на отрезке с использованием технологий Pthreads, программа компилируется без ошибок и предупреждений на вычислительном кластере, в результате работы программы выдаются корректные данные. Проведен комплекс вычислений с использованием написанной программы, построен график зависимости скорости подсчета интеграла от количества потоков, сделаны корректные выводы о зависимости кол-ва потоков на скорость выполнения вычислений.
незачтено	(компетенция не сформирована): Программа не написана или написана некорректная программа. Не проведен комплекс вычислений, не построен график зависимости скорости подсчета интеграла от количества потоков

*Задания по Разделу 3. Технология программирования OpenMP.*

Написать программу численного решения дифференциального уравнения в частных производных с использованием технологий OpenMP. Уравнение, начальные и граничные условия, а также количество потоков вычислений задаются индивидуально.

Построить график зависимости скорости численного решения от количества потоков.

Критерии оценивая:

Оценка	Критерий
зачтено	(компетенция сформирована не ниже порогового уровня):: Написана программа на языке C++ численного решения

	дифференциального уравнения в частных производных с использованием технологий OpenMP, программа компилируется без ошибок и предупреждений на вычислительном кластере, в результате работы программы выдаются корректные данные. Проведен комплекс вычислений с использованием написанной программы, построен график зависимости скорости численного решения от количества потоков, сделаны корректные выводы о зависимости кол-ва потоков на скорость выполнения вычислений.
незачтено	(компетенция не сформирована): Программа не написана или написана некорректная программа. Не проведен комплекс вычислений, не построен график зависимости скорости численного решения от количества потоков

*Задания по Разделу 4. Технология программирования MPI.*

Написать программу численного решения дифференциального уравнения в частных производных с использованием технологий MPI. Уравнение, начальные и граничные условия, а также количество потоков вычислений задаются индивидуально.

Построить график зависимости скорости численного решения от количества потоков.

Критерии оценивая:

Оценка	Критерий
зачтено	(компетенция сформирована не ниже порогового уровня):: Написана программа на языке C++ численного решения дифференциального уравнения в частных производных с использованием технологий MPI, программа компилируется без ошибок и предупреждений на вычислительном кластере, в результате работы программы выдаются корректные данные. Проведен комплекс вычислений с использованием написанной программы, построен график зависимости скорости численного решения от количества потоков, сделаны корректные выводы о зависимости кол-ва потоков на скорость выполнения вычислений.
незачтено	(компетенция не сформирована): Программа не написана или написана некорректная программа. Не проведен комплекс вычислений, не построен график зависимости скорости численного решения от количества потоков

*Задания по Разделу 5. Введение в технологию CUDA.*

Написать программу численного решения дифференциального уравнения в частных производных с использованием технологий CUDA. Уравнение, начальные и граничные условия, а также количество потоков вычислений задаются индивидуально.

Построить график зависимости скорости численного решения от количества потоков.

Критерии оценивая:

Оценка	Критерий
зачтено	(компетенция сформирована не ниже порогового уровня): Написана программа на языке C++ численного решения дифференциального уравнения в частных производных с использованием



	технологий CUDA, программа компилируется без ошибок и предупреждений на вычислительном кластере, в результате работы программы выдаются корректные данные. Проведен комплекс вычислений с использованием написанной программы, построен график зависимости скорости численного решения от количества потоков, сделаны корректные выводы о зависимости кол-ва потоков на скорость выполнения вычислений.
незачтено	(компетенция не сформирована): Программа не написана или написана некорректная программа. Не проведен комплекс вычислений, не построен график зависимости скорости численного решения от количества потоков

### Список заданий к зачету

Зачет выставляется по результатам выполнения заданий для самостоятельной работы и краткого собеседования со студентом после их проверки.

Показатели и критерии, используемые при выставлении оценки:

Оценка	Критерии	Шкала оценивания
Зачтено (компетенция сформирована не ниже порогового уровня)	Знать: – архитектуру параллельных вычислительных систем; - области применимости параллельных вычислений; Уметь: – разбивать программу на независимые процессы ; – интерпретировать результаты полученных вычислений; Владеть навыками: – разработки программ с использованием технологий параллельного программирования OpenMP, MPI и CUDA.	Получены оценка «зачтено» за выполнение всех четырех заданий для самостоятельной работы
Незачтено (компетенция не сформирована)	– – обработки полученных вычислений для формирования соответствующих выводов об эффективности применяемых параллельных вычислений	Не получены хотя бы одной оценки «зачтено» за одно из четырех заданий для самостоятельной работы

### Тест

**1. Для параллельной программы, написанной с использованием технологии OpenMP характерно:**

- А) состоит из последовательных и параллельных секций.
- Б) в начальный момент времени создается главная нить, выполняющая последовательные секции программы.
- В) при входе в параллельную секцию выполняется операция fork, порождающая семейство нитей. Каждая нить имеет свой уникальный числовой идентификатор (главной нити

соответствует 0). При распараллеливании циклов все параллельные нити исполняют один код. В общем случае нити могут исполнять различные фрагменты кода.  
Г) при выходе из параллельной секции выполняется операция join. Завершается выполнение всех нитей, кроме главной.

**2. Какие директивы есть в OpenMP?**

- А) #pragma omp
- Б) #pragma omp parallel
- В) #pragma omp for
- Г) #pragma omp while

**3. В OpenMP нет следующих типов синхронизации**

- А) barrier
- Б) atomic
- В) domain
- Г) master
- Д) ordered
- Е) flush
- Ж) sort
- З) critical

**4. С помощью вызова функции**

**(void) omp\_set\_num\_threads(int num\_threads)**

**можно**

- А) задать количество потоков в области параллельных вычислений
- Б) определить значение переменной окружения OMP\_NUM\_THREADS

**5. Какие из понятий не относятся к технологии Nvidia Cuda**

- А) #pragma omp
- Б) host
- В) kernel
- Г) device

**6. Какие из спецификаторов функций в Cuda не могут быть использованы вместе**

- А) \_\_host\_\_ и \_\_device\_\_
- Б) \_\_global\_\_ и \_\_host\_\_

**7. Для выделения памяти на GPU можно использовать функции**

- А) cudaMalloc
- Б) cudaMallocPitch
- В) cudaNew
- Г) cudaFree

**8. Функция cudaMemcpy используется для**

- А) копирования памяти между CPU и GPU
- Б) копирования памяти между вычислительными узлами кластера
- В) копирования памяти между различными CPU

**9. Одним из распространённых средств разработки программ, основанных на модели обмена сообщениями, является:**

- А) POSIX Threads
- Б) OpenMP
- В) любая реализация MPI

**10. Неблокирующий обмен позволяет:**

- А) повысить производительность параллельной программы
- Б) повысить надежность передачи сообщений
- В) повысить предсказуемость поведения программы

**11. Подпрограмма MPI\_Bcast:**

- А) пересылает всем остальным процессам разные фрагменты данных
- Б) пересылает одну и ту же порцию данных всем остальным процессам

В) выполняет операцию частичного приведения

**12. Второй этап выполнения неблокирующего обмена это:**

А) создание буфера обмена

Б) проверка выполнения обмена

В) проверка доступности буфера обмена

Вопрос №	Правильный ответ	Вопрос №	Правильный ответ	Вопрос №	Правильный ответ
1	А, Б, В, Г	5	А	9	В
2	А, Б, В	6	Б	10	А
3	В, Ж	7	А, Б	11	Б
4	А, Б	8	А	12	Б

Каждый правильный ответ оценивается в 1 балл.

**Оценка за тест**

**Оценка «зачтено»**

11-12 баллов соответствуют формированию проверяемой компетенции на высоком уровне,

8-10 баллов – на продвинутом уровне,

5-7 балла – на пороговом уровне,

**Оценка «незачтено»**

менее 5 баллов – ниже порогового уровня.

## **2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания**

### **2.1. Шкала оценивания сформированности компетенций и ее описание**

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

*Пороговый уровень* - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

*Продвинутый уровень* - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

*Высокий уровень* - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

**2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования**

Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
			Пороговый уровень	Продвинутый уровень	Высокий уровень
Самостоятельная работа по разделам 2-6. Зачет.	1-6	<p>Знать:</p> <ul style="list-style-type: none"> <li>– архитектуру параллельных вычислительных систем;</li> <li>- области применимости параллельных вычислений;</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>– разбивать программу на независимые процессы ;</li> <li>– интерпретировать результаты полученных вычислений;</li> </ul> <p>Владеть навыками:</p> <ul style="list-style-type: none"> <li>– разработки программ с использованием технологий параллельного программирования</li> </ul>	<p>Знание основы архитектуры параллельных систем. Базовые знания и понимание применимости параллельных вычислений</p> <p>Умение выделять в программе независимые процессы</p> <p>Базовое умение интерпретировать результаты полученных вычислений.</p> <p>Владеть основами работы с использованием технологий параллельного</p>	<p>Знать основные параллельные алгоритмы работы для решения практических задач. Продвинутое знание и понимание применимости параллельных вычислений</p> <p>Продвинутое умение интерпретировать результаты полученных вычислений.</p> <p>Владеть продвинутыми средствами работы для технологий параллельного программирования OpenMP, MPI и CUDA. м</p>	<p>Знать продвинутые параллельные алгоритмы работы для решения практических задач. Продвинутое знание и понимание применимости параллельных вычислений в различных сферах науки и производства</p> <p>Уметь практически решать задачи численного характера с использованием адекватных средств параллельных вычислений. Продвинутое умение интерпретировать результаты полученных вычислений.</p> <p>Владеть продвинутыми средствами работы для совместного использования технологий параллельного программирования OpenMP,</p>

		OpenMP, MPI и CUDA. – обработки полученных вычислений для формирования соответствующих выводов об эффективности применяемых параллельных вычислений.	программирования OpenMP, MPI и CUDA Владение базовыми навыками по обработке вычислений		MPI и CUDA. Владение продвинутыми навыками по обработке вычислений
--	--	---	---	--	---

### **3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

#### **3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций**

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

### **3.2 Описание процедуры выставления оценки**

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка. Для дисциплин, изучаемых в течение нескольких семестров, оценка может выставляться не только по окончании ее освоения, но и в промежуточных семестрах. Вид оценки («зачтено», «незачтено») определяется рабочей программой дисциплины в соответствии с учебным планом.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.



## **Приложение №2 к рабочей программе дисциплины «Параллельное программирование»**

### **Методические указания для студентов по освоению дисциплины**

Основной формой изложения учебного материала по дисциплине «Параллельное программирование» являются лекции и практические занятия. Это связано с тем, что в основе читаемого курса лежит особый математический аппарат, с помощью которого решаются довольно сложные и громоздкие задачи. По большому числу тем предусмотрены практические занятия, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам и отработка практических навыков программирования.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и практических занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков работы в течение обучения проводятся мероприятия текущей аттестации в виде самостоятельных работ.

В конце изучения дисциплины студенты сдают зачет. Зачет выставляется по итогам выполнения самостоятельных заданий и краткого собеседования по его результатам.

Освоить вопросы, излагаемые в процессе изучения дисциплины «Параллельное программирование» самостоятельно студенту достаточно сложно. Это связано со сложностью изучаемого материала, требованиям к техническому оснащению курса и большим объемом материала курса. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра, в том числе самостоятельных, сдать зачет по итогам изучения дисциплины студенту достаточно сложно.

### **Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине**

Для самостоятельной работы особенно рекомендуется использовать учебную литературу.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» ([www.biblioclub.ru](http://www.biblioclub.ru)) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (\*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Информационная система "Единое окно доступа к образовательным ресурсам" (<http://window.edu.ru/library>).

Целью создания информационной системы "Единое окно доступа к образовательным ресурсам" (ИС "Единое окно ") является обеспечение свободного доступа к интегральному каталогу образовательных интернет-ресурсов и к электронной библиотеке учебно-методических материалов для общего и профессионального образования.

Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет ([http://lib.uniyar.ac.ru/opac/bk\\_login.php](http://lib.uniyar.ac.ru/opac/bk_login.php)) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню

«Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ ([http://www.lib.uniyar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php)) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/пароллю.

3. Электронная картотека «Книгообеспеченность» ([http://www.lib.uniyar.ac.ru/opac/bk\\_bookreq\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php)) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «Книгообеспеченность» доступна в сети университета и через Личный кабинет.