

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»**

КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
«Рекурсивно-логическое программирование»**

Направления подготовки:  
02.03.03 «Математическое обеспечение  
и администрирование информационных систем»

**Профиль**

Программное обеспечение компьютерных технологий и систем  
искусственного интеллекта

Квалификация выпускника – бакалавр

Форма обучения – очная

Рязань

## **Общие положения**

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общепрофессиональных и профессиональных компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации. Промежуточная аттестация проводится в форме зачета. Форма проведения зачета – тестирование, письменный опрос по теоретическим вопросам и выполнение практических заданий.

### **Описание показателей и критериев оценивания компетенций**

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- эталонный уровень характеризуется максимально возможной выраженной степенью компетенций и является важным качественным ориентиром для самосовершенствования.

### **Уровень освоения компетенций, формируемых дисциплиной:**

#### **Описание критериев и шкалы оценивания тестирования:**

<b>Шкала оценивания</b>	<b>Критерий</b>
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

#### **Описание критериев и шкалы оценивания теоретического вопроса:**

<b>Шкала оценивания</b>	<b>Критерий</b>
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

#### **Описание критериев и шкалы оценивания практического задания:**

<b>Шкала оценивания</b>	<b>Критерий</b>
3 балла	Задача решена верно

(эталонный уровень)	
2 балла (продвинутый уровень)	Задача решена верно, но имеются неточности в логике решения
1 балл (пороговый уровень)	Задача решена верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задача не решена

На промежуточную аттестацию (зачет) выносится тест, два теоретических вопроса и 1 задача. Максимально студент может набрать 12 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено» и «не зачтено».

**Оценки «зачтено»** заслуживает обучающийся, продемонстрировавший полное знание материала изученной дисциплины, усвоивший основную литературу, рекомендованную рабочей программой дисциплины; показавший систематический характер знаний по дисциплине, ответивший на все вопросы билета или допустивший погрешности в ответах на вопросы, но обладающий необходимыми знаниями для их устранения под руководством преподавателя.

Дополнительным условием получения оценки «зачтено» могут стать успехи при выполнении лабораторных работ, систематическая активная работа на лабораторных работах.

Оценка «зачтено» выставляется студенту, набравшему 4 и более баллов при промежуточной аттестации.

**Оценки «не зачтено»** заслуживает обучающийся, продемонстрировавший серьезные пробелы в знаниях основного материала изученной дисциплины, не ответивший на все вопросы билета и дополнительные вопросы. Как правило, оценка «не зачтено» ставится обучающимся, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине (формирования и развития компетенций, закрепленных за данной дисциплиной).

Оценка «не зачтено» выставляется студенту, набравшему менее 4 баллов при промежуточной аттестации.

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ПК-1	Способен проектировать программное обеспечение с использованием современных инструментальных средств

**ПК-1.1. Проектирует и разрабатывает программное обеспечение**

**ПК-1.2. Применяет современные инструментальные средства при разработке программного обеспечения**

## Паспорт оценочных материалов по дисциплине

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
Тема 1. Общая характеристика параллельных вычислительных систем.	ПК-1.1	Зачет
Тема 2. Параллельные вычисления	ПК-1.1	Зачет
Тема 3. Технология параллельного программирования OpenMP	ПК-1.1, ПК-1.2	Зачет
Тема 4. Технология параллельного программирования для графических процессоров CUDA	ПК-1.1, ПК-1.2	Зачет
Тема 5. Технология параллельного программирования MPI	ПК-1.1, ПК-1.2	Зачет
Тема 6. Перспективные направления в параллельном программировании	ПК-1.1	Зачет

## ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

### Промежуточная аттестация в форме зачета

Код компетенции	Результаты освоения ОПОП Содержание компетенций	
ПК-1	Способен проектировать программное обеспечение с использованием современных инструментальных средств	

### **ПК-1.1. Проектирует и разрабатывает программное обеспечение**

#### **Типовые вопросы открытого типа:**

1. Совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенных для автоматизации процессов приема, хранения, обработки и выдачи информации и ориентированных либо на достижение сверхвысокой производительности, либо на повышение надежности и живучести средств вычислительной техники называется \_\_\_\_\_.

**Ответ: Вычислительная система (ВС)**

2. \_\_\_\_\_ называют группу взаимно соединенных вычислительных узлов, работающих совместно и составляющих единый вычислительный ресурс, который создает иллюзию единичной (единственной) вычислительной машины или системы.

**Ответ: Кластерная ВС (кластер)**

3. \_\_\_\_\_ - мера отношения объема вычислений, выполненных в параллельной задаче, к объему коммуникаций.

**Ответ: Гранулярность**

4. \_\_\_\_\_ – это система показателей, позволяющая оценить преимущества, получаемые при параллельном решении задачи на n процессорах, по сравнению с последовательным решением той же задачи на единственном процессоре.

**Ответ: Метрики параллельных вычислений**

5. Отношение времени, затрачиваемого на проведение вычислений на однопроцессорной ВС, ко времени решения той же задачи на параллельной n-процессорной системе называется \_\_\_\_\_.

**Ответ: Ускорение**

6. \_\_\_\_\_ – вычисления с использованием разнородных вычислительных устройств.

**Ответ: Гетерогенные (гибридные) вычисления**

7. Наиболее популярной на текущий момент гетерогенной связкой является \_\_\_\_\_

**Ответ: CPU + GPU**

8. При каком законе параллельных вычислений объем вычислений не изменяется, а главная цель расширения ВС — сократить время вычислений?

**Ответ: закон Амдала.**

9. При каком законе параллельных вычислений время вычислений с расширением системы не меняется, но при этом увеличивается объем решаемой задачи?

**Ответ: закон Густафсона.**

10. При каком уровне параллелизма выполнение команды разделяется на фазы, а фазы нескольких соседних команд могут быть перекрыты за счет конвейеризации?

**Ответ: Микроуровень**

### Типовые тестовые вопросы:

#### Вопрос 1

**Вычислительная система называется многопроцессорной, если:**

- а) она содержит несколько процессоров, работающих с общей оперативной памятью, и управляет разными операционными системами;
- б) содержит несколько процессоров;
- +в) она содержит несколько процессоров, работающих с общей оперативной памятью, и управляет одной общей операционной системой;
- г) она содержит несколько процессоров, работающих с раздельной оперативной памятью, и управляет одной общей операционной системой.

#### Вопрос 2

**Вычислительная система, которая содержит несколько ЭВМ, каждая из которых имеет свою ОП и работает под управлением своей операционной системы:**

- а) многопроцессорной;
- б) однородной;
- в) специализированной;
- +г) многомашинной.

#### Вопрос 3

**Представителями какого класса являются классические фон-неймановские ВМ**

- +а) SISD;
- б) MISD;
- в) SIMD;

г) MIMD.

#### **Вопрос 4**

*Вычислительная машина какой архитектуры позволяют выполнять одну арифметическую операцию сразу над многими данными — элементами вектора.*

- а) SISD;
- б) MISD;
- +в) SIMD;
- г) MIMD.

#### **Вопрос 5**

*Какой уровень параллелизма осуществляется путем одновременного выполнения нескольких независимых задач на разных процессорах, которые, практически, не взаимодействуют друг с другом*

- а) уровень потоков;
- +б) уровень задачий;
- в) микроуровень;
- г) уровень команд

#### **Вопрос 6**

*Отношение времени, затрачиваемого на проведение вычислений на однопроцессорной ВС, ко времени решения той же задачи на параллельной n-процессорной системе:*

- +а) ускорение;
- б) индекс параллелизма;
- в) эффективность;
- г) утилизация.

#### **Вопрос 7**

*Может ли программа быть полностью распараллеленной?*

- а) программа всегда не может быть распараллелена;
- +б) программа может быть распараллелена, но не полностью;
- в) программа всегда может быть распараллелена.

#### **Вопрос 8**

*Укажите основные отличия архитектур CPU и GPU.*

- +а) CPU заточены под скорость выполнения, GPU – под параллельное выполнение;
- б) ядра CPU проще и дешевле, ядра GPU – дороже и сложнее;
- +в) ядра CPU сложнее и больше в размерах, ядра GPU – простые, маленькие, заточены под параллельное выполнение;
- г) только размеры ядер.

#### **Вопрос 9**

*Всегда ли распараллеливание программ даёт выигрыши в скорости вычислений?*

- а) да, всегда даёт выигрыш;
- б) даёт выигрыш только для небольших задач;
- +в) даёт выигрыш только для больших задач;
- г) не даёт выигрыш.

### **Вопрос 10**

*Какие основные особенности архитектуры современных GPU позволили им значительно увеличить вычислительную мощность?*

- +а) малые размеры ядер;
- +б) простота исполнения ядер;
- в) высокая тактовая частота;
- г) применение конвейеризации.

## **ПК-1.2. Применяет современные инструментальные средства при разработке программного обеспечения**

**Типовые вопросы открытого типа:**

1. \_\_\_\_\_ – технология организации параллельных вычислений в системах с общей памятью.

**Ответ: OpenMP**

2. Какое количество одновременно работающих потоков в OpenMP устанавливается по умолчанию?

**Ответ: равно количеству вычислительных ядер.**

3. Какие существуют группы процедур для обмена сообщениями между процессами в MPI?

**Ответ: индивидуальные операции (операции «точка-точка») и коллективные операции**

4. Индивидуальные операции позволяют организовать взаимодействие только \_\_\_\_\_ процессов.

**Ответ: двух**

5. Коллективные операции позволяют организовать взаимодействие \_\_\_\_\_ процессов, входящих в один коммуникатор.

**Ответ: всех**

6. Что такое ранг процесса в MPI?

**Ответ: номер процесса**

7. \_\_\_\_\_ в MPI - специально создаваемый служебный объект, объединяющий в своем составе группу процессов и ряд дополнительных параметров.

**Ответ: коммутатор**

8. \_\_\_\_\_ — программный интерфейс для передачи информации, который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу в ВС с распределенной памятью.

**Ответ: MPI**

9. \_\_\_\_\_ - это программно-аппаратная архитектура параллельных вычислений, которая позволяет существенно увеличить вычислительную производительность благодаря использованию графических процессоров фирмы Nvidia

**Ответ: CUDA**

10. Что является единицей выполнения программы в CUDA?

**Ответ: Тред (Thread), поток, нить**

11. Объединение потоков, которое выполняется целиком на одном SM (потоковом мультипроцессоре) называется \_\_\_\_\_.

**Ответ: Блок (Block)**

**Типовые тестовые вопросы:**

**Вопрос 1**

*Каким образом осуществляется программирование с помощью OpenMP?*

- а) с помощью новых встроенных функций;
- +б) с помощью новых встроенных функций и директив;
- в) с помощью новых встроенных функций, директив и типов.

**Вопрос 2**

*Что такое поток в OpenMP? В чём заключается разница между master и slave потоками?*

- а) поток – это экземпляр программы, все потоки равны;
- б) поток – это экземпляр программы, master поток управляет slave потоками;
- в) поток – это обособленная копия набора команд вместе со своим стеком, все потоки равны;
- +г) поток – это обособленная копия набора команд вместе со своим стеком, master поток управляет slave потоками.

**Вопрос 3**

*Директива parallel в технологии OpenMP:*

- а) запускает цикл на нескольких параллельных потоках;
- +б) запускает код в нескольких параллельных потоках;
- в) обозначает критическую секцию;
- г) обозначает барьер синхронизации параллельных потоков.

**Вопрос 4**

*Выберите все объекты OpenMP, которые позволяют управлять настройками количества потоков.*

- а) опция threadprivate;
- +б) опция num\_threads;
- +в) функция omp\_set\_num\_thread;
- г) директива single.

**Вопрос 5**

*Чем модификатор firstprivate отличается от модификатора lastprivate в OpenMP?*

- а) firstprivate задаёт значение переменной только в потоке-мастере, а lastprivate – только в подчинённом потоке;
- +б) firstprivate задаёт значение переменной при входе в поток, а lastprivate – на выходе из потока;
- в) firstprivate обнуляет значение переменной только в потоке-мастере, а lastprivate – только в подчинённом потоке;
- г) ничем не отличаются;

### **Вопрос 6**

*Выберите верные утверждения об опции reduction в OpenMP.*

- а) опция распараллеливает циклы;
- б) опция распараллеливает условные операторы;
- +в) опция выполняет указанную операцию на выходе из параллельной области;
- г) опция выполняет очистку переменных на выходе из параллельной области.

### **Вопрос 7**

*Какие функции выполняет критическая секция в OpenMP?*

- +а) впускает не более одного потока одновременно;
- б) впускает только мастер поток;
- в) включает режим высокоточных вычислений;
- г) является параллельным аналогом try-catch.

### **Вопрос 8**

*Какие существуют способы настройки OpenMP вне директив?*

- а) иных способов не существует;
- б) иных встроенных способов не существует, но можно реализовать самому;
- +в) существуют встроенные функции;
- г) существуют встроенные классы.

### **Вопрос 9**

*Укажите способы, которыми участок кода параллельной программы может быть выполнен только одним потоком.*

- +а) директива single;
- +б) директива if;
- +в) директива master;
- г) функция omp\_set\_num\_threads.

### **Вопрос 10**

*Какие существуют режимы планирования распараллеливания итераций цикла?*

- а) deferred – отложенный режим (зависит от пришедших данных);
- +б) static – статическое распределение по количеству итераций;
- в) lazy – ленивое распределение (равномерно-случайное);
- +г) guided – управляемое распределение (в зависимости от нагрузки).

### **Вопрос 11**

*Как работает барьер?*

- а) впускает только первый поток;
- б) впускает только мастер поток;
- в) впускает первый поток, ждёт завершения, впускает следующий;
- +г) ждёт, пока все потоки подойдут к нему, после впускает всех одновременно.

### **Вопрос 12**

***Какое поведение определяет опция copyin?***

- а) копирует значения всех указанных переменных в параллельную область;
- +б) копирует значения всех threadprivate переменных в параллельную область со значениями из нити мастера;
- в) создаёт внутри параллельных областей threadprivate переменные со значениями по умолчанию;
- г) возвращает из параллельной области значения указанных переменных.

### **Вопрос 13**

***Что такое устройство в терминах CUDA?***

- а) ядро GPU;
- +б) графический ускоритель;
- в) устройство, управляющее процессом выполнения и компиляции;

### **Вопрос 14**

***Что такое хост?***

- а) ядро GPU;
- б) графический ускоритель;
- +в) устройство, управляющее процессом выполнения и компиляции;

### **Вопрос 15**

***Что такое ядро (kernel)?***

- а) ядро GPU;
- б) графический ускоритель;
- в) CPU;
- +г) программа на CUDA.

### **Вопрос 16**

***Как осуществляется компиляция и выполнение программы, написанной с помощью технологии CUDA?***

- а) программа компилируется и выполняется полностью на GPU;
- +б) программа компилируется на CPU, выполняется на GPU;
- в) часть программы компилируется на CPU, часть – GPU, выполнение всегда на GPU.

### **Вопрос 17**

Какой из перечисленных способов распараллеливания наиболее оптимальный в большинстве случаев?

- а) по гридам
- б) по блокам;
- в) по тредам;
- +г) по блокам и тредам;
- д) всё вместе.

### **Вопрос 18**

*В чём преимущества shared памяти?*

- а) дешевизна;
- б) объём;
- +в) скорость.

### **Вопрос 19**

Каким образом можно спроектировать на плоскость текущее положение треда в блоке?

- а) int threadIdx = blockIdx.x + threadIdx.x;
- +б) int threadIdx = blockIdx.x \* blockDim.x + threadIdx.x;
- в) int threadIdx = blockIdx.x \* blockDim.y + threadIdx.y \* blockDim.x;
- г) int threadIdx = blockDim.x + threadIdx.y.

### **Вопрос 20**

*Какие параметры используются при запуске ядра?*

- а) grid, block – количество блоков в гриде и потоков в блоке;
- +б) grid, block, mem, stream - количество блоков в гриде, потоков в блоке, объём дополнительной shared-памяти, поток вызова;
- в) mem, stream - объём дополнительной shared-памяти и поток вызова;
- г) dev, mem, block – номер устройства, количество выделяемой памяти на блок, количество потоков в блоке.

### **Вопрос 24**

**Верно ли утверждение: При работе с блокирующими функциями MPI может возникнуть тупиковая ситуация**

- +а) Да;
- б) Нет

### **Вопрос 25**

**Верно ли утверждение: В MPI существуют как shared, так private переменные?**

- а) Да;
- +б) Нет

## **Типовые теоретические вопросы для зачета по дисциплине**

1. Тенденции развития современных процессоров: многопоточность и многоядерность.
2. Классификация архитектур ВС Флинна. Классификация Ванга и Бриггса.
3. Уровни параллелизма. Степень гранулярности. Метрики параллелизма.
4. Закономерности параллельных вычислений. Закон Амдала. Закон Густафсона. Закон Сан-Ная.
5. Технология параллельного программирования OpenMP. Основные понятия и определения. Модель вычислений и классы переменных.
6. OpenMP. Параллельные и последовательные области. Директива parallel.
7. OpenMP. Конструкции распределения работы.

8. Синхронизация в OpenMP.
9. Гетерогенные ВС. Архитектура GPU.
10. Обзор средств программирования для GPU.
11. CUDA. Понятие потока, блока, сети блоков. Функция-ядро. Спецификаторы функций и переменных.
12. CUDA. Иерархия памяти на GPU.
13. CUDA. Глобальная память. Шаблон работы с глобальной памятью. Использование pinned-памяти. CUDA-потоки.
14. CUDA. Разделяемая память. Шаблон работы с разделяемой памятью. Оптимизация работы с разделяемой памятью. Банк-конфликты.
15. MPI. Основные понятия и определения. Структура MPI-программы. Типы данных.
16. MPI. Коммуникационные операции типа «Точка-точка».
17. MPI. Тупиковые ситуации и способы борьбы с ними.
18. MPI. Коллективные коммуникационные операции.

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СОГЛАСОВАНО      ФГБОУ ВО "РГРТУ", РГРТУ, Костров Борис Васильевич,  
Заведующий кафедрой ЭВМ

27.11.25 12:54 (MSK)

Простая подпись