

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Александрович
Должность: И.о. директора института перспективной инженерии
Дата подписания: 18.06.2026 12:13:40
Уникальный программный ключ:
990bea3aeec48c23bd8699e48288f8d1960c600

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное учреждение
высшего образования**
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Директор института
перспективной инженерии,
кандидат технических наук,
доцент

Порохня А. А.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

Параллельные вычисления

название дисциплины (модуля)

Направленность (профиль)	09.04.02 Информационные системы и технологии
Профиль	Управление данными
Форма обучения	очная
Год начала обучения	2026
Реализуется в 1 семестре	

Введение

1. Назначение: Оценочные материалы (оценочные средства) предназначены для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, и представляют собой совокупность контрольно-измерительных материалов и методов их использования для измерения уровня достижения обучающимися установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «Параллельные вычисления»

3. Разработчик Николаев Е.И., доцент департамента цифровых, робототехнических систем и электроники.

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Дроздова В. И., профессор департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Члены комиссии:

Шагрова Г. В., профессор департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Винокурский Д. Л., доцент департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Представитель организации-работодателя: Ю. В. Рокотов, к.т.н., директор ООО «КиберСофт»

(ФИО, должность)

Экспертное заключение: Оценочные материалы (оценочные средства) по дисциплине «Параллельные вычисления» составлены в соответствии с требованиями самостоятельно устанавливаемого Федерального государственного образовательного стандарта высшего образования по направлению 09.04.02 «Информационные системы и технологии» и позволяют оценить уровень сформированности компетенции ОПК-2.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенци(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
Компетенция: ОПК-2				
Результаты обучения по дисциплине (модулю): Разрабатывает оригинальные алгоритмы для решения профессиональных задач <i>Индикатор:</i> ИД-1 _{ОПК-2}	Не отвечает (или отвечает некорректно) на вопросы собеседования. Не способен ответить по вопросам коллоквиума. Не предоставил ответы на контрольные вопросы лабораторных работ	При собеседовании некорректно отвечает на вопрос. Демонстрирует незнание отдельных вопросов теоретического материала в рамках коллоквиума. Не ответил на большинство контрольных вопросов лабораторных работ.	В рамках собеседования в полной степени раскрыто содержание вопроса для собеседования. При ответе на вопросы коллоквиума демонстрирует пробелы в знании отдельных элементов теоретического материала.	В рамках собеседования в полной степени раскрыто содержание вопроса для собеседования. В рамках коллоквиума даны ответы на все вопросы преподавателя. В рамках защиты
Результаты обучения по дисциплине (модулю): Использует современные интеллектуальные технологии для решения профессиональных задач <i>Индикатор:</i> ИД-2 _{ОПК-2}	Не выполняет учебное задание лабораторных работ. Не способен решать задачи разрабатывать приложения в рамках индивидуального задания к лабораторным работам. Не способен проектировать архитектуру приложений для решения задач.	Выполняет корректно учебное задание лабораторных работ. Не способен решать задачи разрабатывать приложения в рамках индивидуального задания к лабораторным работам. Проектирование архитектуры приложения реализует при помощи преподавателя.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает приложения в рамках индивидуального задания к лабораторным работам.	Выполняет корректно учебное задание лабораторных работ. Разрабатывает и документирует приложения в рамках индивидуального задания к лабораторным работам.

			Проектирование архитектуры приложения реализует при помощи преподавателя.	
Результаты обучения по дисциплине (модулю): Разрабатывает оригинальные программные средства для решения профессиональных задач <i>Индикатор:</i> ИД-З _{ОПК-2}	Не владеет навыками по документированию приложений. Не обладает навыками использования интегрированных сред разработки. Не владеет навыками тестирования, отладки и документирования информационных систем.	Не владеет навыками по документированию приложений. Не обладает навыками использования интегрированных сред разработки. Владеет навыками тестирования, отладки и документирования информационных систем.	Владеет навыками по документированию приложений. Демонстрирует навыки использования интегрированных сред разработки.	Владеет навыками по документированию приложений. Демонстрирует навыки уверенного использования интегрированных сред разработки. Владеет навыками тестирования, отладки и документирования информационных систем.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		Вопросы к экзамену:	
1.		1. Паттерн Tiling/grids 2. Паттерн «разделяй и властвуй» (divide and conquer) 3. Декомпозиция параллельных программ 4. Потoki 5. Модель исполнения и ядро CUDA 6. Блоки и сетки 7. Модель памяти CUDA 8. Регистры 9. Разделяемая память 10. Константная память 11. Глобальная память 12. Локальная память 13. Конфигурации, основанные на ГП и ЦП 14. Аппаратная архитектура графических процессоров 15. Аппаратные средства компьютерной графики 16. Вычислительные возможности графического процессора 17. Архитектура графического процессора 18. Архитектура программного обеспечения для GPU 19. Уровни CUDA API 20. Структура CUDA-программы 21. Альтернативные решения для гетерогенных вычислений 22. OpenCL 23. Технология DirectCompute 24. Параллельное программирование на CPU 25. Шаблоны параллельного программирования 26. Паттерн, основанный на циклах 27. Паттерн Fork/join 28. Трассировка лучей с использованием константной памяти	ОПК-2

		29. Производительность приложения с использованием константной памяти 30. Измерение производительности 31. Текстурная память 32. Моделирование теплообмена 33. Модель теплообмена 34. Визуализация теплообменных процессов 35. Применение текстурной памяти 36. Двумерная текстурная память 37. Разделяемая память и синхронизация 38. Константная память 39. Основы работы с константной памятью 40. Основы метода трассировки лучей 41. Трассировка лучей на GPU	
		Задания закрытого типа	
2.		Какие типы относятся к типам-делегатам? 1) Action<T> 2) Func<T> 3) List<T> 4) IEnumerable<T>	ОПК-2
3.		Какой метод позволяет реализовать остановку выполнения программы и ожидание продолжения выполнения в течении заданного количества миллисекунд? 1) Thread.Sleep() 2) System.Console() 3) Math.Wait() 4) Waiting.Stop()	ОПК-2
4.		Какой класс используется для представления потока в многопоточном приложении? 1) Thread 2) Task	ОПК-2

		3) MultiTask 4) Waiting 5) Action<T>	
5.		Какое пространство имен является корневым в библиотеке .NET 1) Data 2) System 3) Object 4) Input 5) IO	ОПК-2
6.		В каких фрагментах кода объявляются переменные? 1) <code>int a;</code> 2) <code>int b = 7;</code> 3) <code>a = 123;</code> 4) <code>string str = "Hello, World!!!";</code>	ОПК-2
7.		С помощью какого зарезервированного слова происходит объявление пространства имен? 1) using 2) break 3) namespace 4) foreach 5) name	ОПК-2
8.		Даны определения классов на языке C#.	ОПК-2

		<pre> class Child : Parent { int b; void E() { b = a; } } public class Parent { protected int a; } class Child1 : Parent { int c; public Child1() { } } </pre> Существуют ли ошибки в данном фрагменте? 1) ошибок нет 2) переменная a является недоступной в производном классе, следовательно, присвоение b = a некорректно 3) в двух классах отсутствуют конструкторы 4) все поля классов являются закрытыми, поэтому их использование не представляется возможным	
9.		Что такое конструктор по умолчанию в языке C#? 1) Конструктор по умолчанию представляет собой метод, который вызывается перед созданием объекта и устанавливает значения всех полей класса в предопределенные значения 2) Программист, определяя класс, может явно не реализовывать конструктор. В этом случае .NET Framework использует пустой метод в качестве конструктора по умолчанию 3) Программист определяет сколько угодно своих конструкторов, но .NET Framework будет игнорировать их, если не отключен конструктор по умолчанию	ОПК-2
10.		Даны определения классов на языке C#.	ОПК-2

		<pre> public class Parent { protected int a; Parent() { a = 0; } } class Child1 : Parent { int c; public Child1() { :base() } } </pre> Существуют ли ошибки в данном фрагменте? 1) ошибок нет 2) конструктор базового класса private, а производного – public 3) все поля классов являются закрытыми, поэтому их использование не представляется возможным 4) конструктор базового класса не виден из производного класса	
11.		Даны определения классов на языке C#. <pre> public class Parent { protected int a; protected Parent() { a = 0; } } class Child1 : Parent { int c; public Child1() { :base() } } </pre> Существуют ли ошибки в данных фрагментах? 1) конструктор базового класса не виден из производного класса 2) конструктор базового класса private, а производного – public 3) все поля классов являются закрытыми, поэтому их использование не представляется возможным 4) ошибок нет	ОПК-2

12.	Даны определения классов на языке C#. <pre> public class Parent: Child1 { protected int a; protected Parent() { a = 0; } } class Child1 : Child { int c; } class Child : Parent { int b; void E() { b = a; } } </pre> Существуют ли ошибки в данных фрагментах? 1) классы наследуются по кругу 2) неверное использование модификатора доступа protected в классе Parent 3) в двух классах отсутствуют конструкторы 4) все поля классов являются закрытыми, поэтому их использование не представляется возможным 5) ошибок нет	ОПК-2
13.	Сколько наследников может быть у представленного класса? <pre> public class Child : Parent { int b; void E() { b = a; } } </pre>	ОПК-2

		1) столько, сколько реализует программист (принципиальных ограничений нет) 2) количество наследников ограничивается в базовом классе Parent 3) не более 128	
14.		В языке C# программист определил класс следующим образом) <pre> public class Parent { protected int a; protected Parent() { a = 0; } } </pre> Какой класс является базовым для данного класса? 1) System.Data 2) System.Object 3) System.IO 4) нет базового класса	ОПК-2
15.		В языке C# программист определил класс следующим образом) <pre> public class Parent { protected int a; protected Parent() { a = 0; } } </pre> Как характеризуется переменная a по отношению ко всем производным классам? 1) в производных классах переменная a является недоступной как внутри класса, так и из объектной переменной	ОПК-2

		2) в производных классах переменная a является недоступной внутри класса, но доступной из объектной переменной 3) в производных классах переменная a является видимой внутри класса, но недоступной из объектной переменной 4) в производных классах переменная a является видимой как внутри класса, так и из объектной переменной	
16.		В языке C# программист определил классы и создал их объекты следующим образом) <pre> public class Parent { protected int a; protected Parent(int pA) { a = pA; } } public class Child : Parent { int b; public Child() :base(0) { b = a; } } static void Main(string[] args) { Parent ob1 = new Parent(); Child ob2 = new Child(); } </pre> Какая ошибка была допущена? 1) некорректное использование зарезервированного слова base 2) в конструкторе класса Child нет доступа к переменной a базового класса 3) нельзя создавать объект ob1, так как класс Parent не имеет открытого конструктора 4) нельзя создать объект базового класса, так как объявлен класс-наследник 5) ошибок нет	ОПК-2
17.		В языке C# программист определил классы и создал их объекты следующим образом)	ОПК-2

		<pre> public class Parent { protected int a; protected Parent(int pA) { a = pA; } public Parent() : this(0) { } } public class Child : Parent { int b; public Child() : base() { b = a; } } static void Main(string[] args) { Parent ob1 = new Parent(); Child ob2 = new Child(); } </pre> Какая ошибка была допущена? 1) ошибок нет 2) некорректное использование зарезервированного слова base 3) в конструкторе класса Child нет доступа к переменной a базового класса 4) некорректное использование зарезервированного слова this 5) нельзя создавать объект ob1, так как класс Parent не имеет открытого конструктора	
18.		В языке C# программист определил классы и интерфейсы следующим образом) <pre> public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } public void Draw() { Console.WriteLine(data.ToString()); } } public interface IDraw { void Draw(); } public interface IString { string GetStr(); } </pre>	ОПК-2

		Какая ошибка была допущена? 1) нет ошибок 2) класс MainObject не может иметь двух базовых классов 3) некорректное использование зарезервированного слова interface 4) в классе IString метод GetStr не определен	
19.		На языке C# программист определил класс, интерфейсы и создал объекты следующим образом) <pre> public interface IDraw { void Draw(); } public interface IString { string GetStr(); } public class Parent { protected int a; } public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } public void Draw() { Console.WriteLine(data.ToString()); } } static void Main(string[] args) { MainObject ob1 = new MainObject(); ob1.Draw(); Parent ob2 = new Parent(); ob2.Draw(); } </pre> Какая ошибка была допущена? 1) в классе Parent нет конструктора, поэтому объект ob2 не может быть создан 2) в классе Parent не определен метод Draw, который вызывается через объектную переменную ob2	ОПК-2

		3) класс MainObject не может иметь двух базовых классов 4) ошибок нет	
20.		На языке C# программист определил класс, интерфейсы и создал объекты следующим образом) <pre> public interface IDraw { void Draw(); } public interface IString { string GetStr(); } public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } </pre> Какая ошибка была допущена? 1) класс MainObject не может иметь двух базовых классов 2) класс MainObject не реализует метод Draw, наследуемого интерфейса IDraw 3) класс MainObject содержит метод GetStr, который дублирует метод базового класса IString 4) ошибок нет	ОПК-2
21.		На языке C# программист определил класс, интерфейсы)	ОПК-2

		<pre> public interface IString { string GetStr(); } public class Data1 : IString { int data; public string GetStr() { return "Данные: " + data.ToString(); } } </pre> <pre> public class Data2 : IString { string data; public string GetStr() { return data; } } </pre> Затем программист реализовал функцию Main) <pre> static void Main(string[] args) { IString[] mas = new IString[] { new Data1(), new Data2() }; } </pre> Какая ошибка была допущена? 1) нет ошибок 2) в массиве присутствуют элементы различного типа 3) не указан размер массива при его создании 4) тип массива, заявленный при объявлении, и тип добавляемых элементов не совпадают	
22.		На языке C# программист определил класс, интерфейсы)	ОПК-2

		<pre> public interface IString { string GetStr(); } public class Data1 : IString { int data; public string GetStr() { return "Данные: " + data.ToString(); } } </pre> <pre> public class Data2 : IString { string data; public string GetStr() { return data; } } </pre> Затем программист реализовал функцию Main) <pre> static void Main(string[] args) { IString[] mas = new IString[] { new Data1(), new Data2() }; IString ob = new IString(); } </pre> Какая ошибка была допущена? 1) нет ошибок 2) Невозможно создать объект интерфейса ob 3) в массиве присутствуют элементы различного типа 4) тип массива, заявленный при объявлении, и тип добавляемых элементов не совпадают	
23.		Для чего служит оператор break? 1) прерывает выполнение текущей итерации цикла и цикла в целом 2) прерывает выполнение текущей итерации цикла и переходит к следующей итерации 3) прерывает выполнение текущей итерации цикла и переходит к выполнению всего цикла с начала	ОПК-2

24.		Для чего служит оператор continue? 1) прерывает выполнение текущей итерации цикла и цикла в целом 2) прерывает выполнение текущей итерации цикла и переходит к следующей итерации цикла 3) прерывает выполнение текущей итерации цикла и переходит к выполнению всего цикла с начала	ОПК-2
25.	3	В переменной Sum по завершении выполнения фрагмента кода будет содержаться... <pre> long Sum = 0; int i; for (i = 1; i <= 1000000; i++) if (i % 3 == 0) Sum += i; </pre> 1) сумма кубов всех чисел от 1 до 1000000 2) 0 3) сумма всех чисел кратных 3 и находящихся в диапазоне от 1 до 1000000	ОПК-2
26.		Задания открытого типа	
27.		Библиотека от компании NVidia для реализации параллельных вычислений на базе графических процессов – это ...	ОПК-2
28.		С помощью какого зарезервированного слова происходит подключение пространства имен для использования его типов?	ОПК-2
29.		Самый распространенный интерфейс для подключения графических карт на материнской плате ЭВМ – это...	ОПК-2
30.		Семейство видеокарт производителя NVidia, объединяющее устройства для настольных компьютеров и ноутбуков	ОПК-2
31.		Как называется отдельный экземпляр класса?	ОПК-2

32.		Семейство видеокарт производителя NVidia, объединяющее устройства для профессиональной работы с компьютерной графикой	ОПК-2
33.		Семейство видеокарт производителя NVidia, объединяющее устройства для мобильных и встраиваемых устройств	ОПК-2
34.		Семейство видеокарт производителя NVidia, объединяющее устройства для высокопроизводительной обработки данных	ОПК-2
35.		Популярная технология в области обработки больших объемов данных (технологии Big Data), свободно распространяемая версия фреймворка Google MapReduce	ОПК-2
36.		Тип памяти, используемый в программировании графических процессоров, предполагающий как доступ как со стороны host, так и со стороны device – это _____ память	ОПК-2
37.		Глобальная память может быть выделена из host-кода с использованием функции _____ ()	ОПК-2
38.		Глобальная память может быть освобождена из host-кода с использованием функции _____ ()	ОПК-2
39.		_____ память содержит стек для каждого потока в ядре CUDA	ОПК-2
40.		_____ память – это часть глобальной памяти, которая кэшируется в SM и характеризуется режимом доступа read only	ОПК-2
41.		При разработке приложения CUDA требуется объявить функцию со следующими характеристиками: – выполняется на device; – может быть вызвана только с device. С каким модификатором должна быть объявлена данная функция?	ОПК-2
42.		При разработке приложения CUDA требуется объявить функцию со следующими характеристиками: – исполняется на device; – вызывается с host;	ОПК-2

		– может быть вызвана с device для GPU с compute capability выше 3.x. С каким модификатором должна быть объявлена данная функция?	
43.		При разработке приложения CUDA требуется объявить функцию со следующими характеристиками: – выполняется на host; – может быть вызвана только с host. С каким модификатором должна быть объявлена данная функция?	ОПК-2
44.		Встроенная переменная CUDA для описания размерности сетки	
45.		Встроенная переменная CUDA для описания индекса блока в сетке	
46.		Встроенная переменная CUDA для описания размерности блока	
47.		Встроенная переменная CUDA для индекса потока в блоке	
48.		Встроенная переменная CUDA, содержит размер варпа	
49.		Функция для реализации атомарной операции CUDA, предназначенная для сложения	
50.		_____ память – это is read-only память, доступная через различные команды, инициирующие запросы на чтение, которые обслуживаются иерархической структурой кэшей, оптимизированной для взаимодействия с множеством потоков.	

2. Описание шкалы оценивания

Рейтинговая система оценки не предусмотрена для студентов, обучающихся на образовательных программах уровня высшего образования магистратуры.

3. Критерии оценивания компетенций заданий открытого типа

Оценка **«отлично»** выставляется студенту, если являются верными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; представленный документ не содержит ошибок и решены все задачи. Компетенция ОПК-2 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если являются верными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; отчетный документ содержит незначительные ошибки, не влияющие на общий результат и не представлены отдельные задачи, не влияющие на общий результат. Компетенция ОПК-2 освоена на среднем уровне.

Оценка **«удовлетворительно»** выставляется студенту, если в основном верными являются результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; отчетный документ содержит ошибки, не позволяющие оценить качество разработки; представлены не все данные и задачи решены не в полном объеме. Компетенция ОПК-2 освоена на минимальном уровне.

Оценка **«неудовлетворительно»** выставляется студенту, если являются неверными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; представленный документ содержит много грубых ошибок, не представлены ключевые данные и задачи решены недостаточно полно. Компетенция ОПК-2 не сформирована.

4. Критерии оценивания компетенций заданий закрытого типа

Оценка **«отлично»** выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция ОПК-2 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ОПК-2 освоена на среднем уровне.

Оценка **«удовлетворительно»** выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция ОПК-2 освоена на минимальном уровне.

Оценка **«неудовлетворительно»** выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ОПК-2 не сформирована.