

Министерство науки и высшего образования Российской Федерации  
**Муромский институт (филиал)**  
федерального государственного бюджетного образовательного учреждения высшего образования  
**«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»**  
(МИ ВлГУ)

**Кафедра ЭиВТ**

«УТВЕРЖДАЮ»  
Заместитель директора по УР  
\_\_\_\_\_ Д.Е. Андрианов  
\_\_\_\_\_ 16.06.2020

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

*Архитектура и программирование гетерогенных вычислительных систем*

**Направление подготовки**

*09.03.01 Информатика и вычислительная техника*

**Профиль подготовки**

*Вычислительные машины, комплексы, системы и сети*

Семестр	Трудоем- кость, час./зач. ед.	Лек- ции, час.	Практи- ческие занятия, час.	Лабора- торные работы, час.	Консультация, час.	Конт- роль, час.	Всего (контакт- ная работа), час.	СРС, час.	Форма промежу- точного контроля (экз., зач., зач. с оц.)
7	144 / 4	16		16	3,8	0,35	36,15	81,2	Экз.(26,65)
8	72 / 2	12	12	12		2,25	38,25	33,75	Зач.
Итого	216 / 6	28	12	28	3,8	2,6	74,4	114,95	26,65

Муром, 2020 г.

## 1. Цель освоения дисциплины

Цель дисциплины: Дать слушателям достаточно полное представление об основах технологии параллельного программирования CUDA и OpenCL для современных графических ускорителей.

Результатами учебных занятий являются: освоение основных принципов параллельного программирования для графических ускорителей. Использование новых идей и новой технологии позволит использовать все возможности современных процессоров и графических ускорителей для получения решения сложных многомерных задач с помощью современных языков программирования.

## 2. Место дисциплины в структуре ОПОП ВО

Данная дисциплина связана и базируется на таких дисциплинах как: «Вычислительная математика», «Программирование», «ЭВМ и периферийные устройства», «Сети и телекоммуникации», «Операционные системы», «Параллельные вычислительные системы». Дисциплина является основой для проектирования выпускной квалификационной работы.

## 3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ПК-5 Способен разрабатывать требования, проектировать программное обеспечение, разрабатывать стратегии тестирования, проектировать пользовательские интерфейсы	ПК-5.1 Осуществляет проектирование программного обеспечения с применением современных средств и технологий.	Знать области применения технологий распределенной обработки данных (ПК-5.1) Знать технологию разработки алгоритмов и программ, методы отладки и решения задач на современных многоядерных и распределенных вычислительных системах: суперЭВМ, вычислительных кластерах (ПК-5.1) Знать современные технологии распределенной обработки данных; принципы построения и методы работы в распределенных вычислительных системах обработки информации (ПК-5.1) Уметь строить модели вычислительных процессов с учетом особенностей архитектуры вычислительных систем, на которых планируется использовать разрабатываемое ПО (ПК-5.1) Уметь проектировать и разрабатывать распределенные	вопросы к устному опросу

		<p>приложения на базе многоуровневой архитектуры и распределенных вычислений (ПК-5.1)</p> <p>Уметь применять технологии построения распределенных вычислительных систем обработки информации (ПК-5.1)</p> <p>Владеть технологиями виртуализации, параллельных вычислений, облачных вычислений и консолидации распределенных ресурсов (ПК-5.1)</p> <p>Владеть практическими навыками проектирования распределенных вычислительных систем обработки информации (ПК-5.1)</p>	
--	--	---	--

## 4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

### 4.1. Форма обучения: очная

Уровень базового образования: среднее общее.

Срок обучения 4г.

#### 4.1.1. Структура дисциплины

№ п/п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Архитектура CPU и GPU. Основные понятия	7	6								устный опрос
2	Программная модель CUDA и OpenCL	7	10		16					81,2	устный опрос
Всего за семестр		144	16		16			3,8	0,35	81,2	Экз.(26,65)
3	Программная модель CUDA и OpenCL	8	2							33,75	устный опрос
4	Модель памяти GPU	8	4							-9,75	устный опрос
5	Практическое применение GPGPU	8	6	12	12					9,75	устный опрос
Всего за семестр		72	12	12	12		+	0	2,25	33,75	Зач.
Итого		216	28	12	28			3,8	2,6	114,95	26,65

#### 4.1.2. Содержание дисциплины

##### 4.1.2.1. Перечень лекций

##### Семестр 7

Раздел 1. Архитектура CPU и GPU. Основные понятия

##### Лекция 1.

История развития вычислительных систем. Типы параллелизма (2 часа).

##### Лекция 2.

Критерии применимости параллельных вычислений. Сравнение классической архитектуры Intel и AMD (2 часа).

### **Лекция 3.**

Принципиальное отличие классической и GPU архитектуры. Объединенная архитектура графических процессоров. Преимущества унифицированной архитектуры (2 часа).

*Раздел 2. Программная модель CUDA и OpenCL*

### **Лекция 4.**

Программный стек CUDA (2 часа).

### **Лекция 5.**

Команды работы с памятью. Пример вызова CUDA (2 часа).

### **Лекция 6.**

Понятия Work Item, Work Group и Warp. Команды работы с памятью. Компиляция и запуск CUDA программ (2 часа).

### **Лекция 7.**

Понятия Host и Device. Контекст и очередь исполнения. Сборка и запуск ядер на устройствах (2 часа).

### **Лекция 8.**

Язык OpenCL (2 часа).

## **Семестр 8**

*Раздел 3. Программная модель CUDA и OpenCL*

### **Лекция 9.**

Понятия Work Item, Work Group и Warp. Команды работы с памятью. Компиляция и запуск OpenCL программ (2 часа).

*Раздел 4. Модель памяти GPU*

### **Лекция 10.**

Глобальная, константная, текстурная, локальная, разделяемая и регистровая память. Размещение различных данных в различной памяти (2 часа).

### **Лекция 11.**

Сравнения производительности глобальной и текстурной памяти на задачах произвольного чтения (2 часа).

*Раздел 5. Практическое применение GPGPU*

### **Лекция 12.**

Использование Scan, Reduce, Histogram, Bitonic sort (2 часа).

### **Лекция 13.**

Использование текстурной памяти. Использование аппаратной интерполяции (2 часа).

### **Лекция 14.**

Алгоритмы обработки данных на гетерогенных вычислительных системах (2 часа).

## **4.1.2.2. Перечень практических занятий**

## **Семестр 8**

*Раздел 5. Практическое применение GPGPU*

### **Практическое занятие 1**

Программный стек CUDA (2 часа).

### **Практическое занятие 2**

Программный стек CUDA (2 часа).

### **Практическое занятие 3**

Программная модель OpenCL (2 часа).

### **Практическое занятие 4**

Программная модель OpenCL (2 часа).

### **Практическое занятие 5**

Отличия модели исполнения, работы с текстурами, сборки и компиляции программ OpenCL от CUDA (2 часа).

### **Практическое занятие 6**

Отличия модели исполнения, работы с текстурами, сборки и компиляции программ OpenCL от CUDA (2 часа).

### 4.1.2.3. Перечень лабораторных работ

#### Семестр 7

*Раздел 2. Программная модель CUDA и OpenCL*

##### Лабораторная 1.

Обработка одномерных массивов и матриц при помощи CUDA (4 часа).

##### Лабораторная 2.

Обработка одномерных массивов и матриц при помощи CUDA (4 часа).

##### Лабораторная 3.

Визуализация анимированных графиков функций при помощи CUDA (4 часа).

##### Лабораторная 4.

Визуализация анимированных графиков функций при помощи CUDA (4 часа).

#### Семестр 8

*Раздел 5. Практическое применение GPGPU*

##### Лабораторная 5.

Работа с текстурной памятью (4 часа).

##### Лабораторная 6.

Работа с текстурной памятью (4 часа).

##### Лабораторная 7.

Работа с текстурной памятью (4 часа).

### 4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Программирование на CUDA Фортране.
2. Операторы CUDA Фортрана.
3. Поточковые блоки. Подпрограммы и функции. Обработка ошибок.
4. Переменные спецификаторы.
5. Предопределенные переменные в подпрограммах ГП.
6. Конфигурация выполнения.
7. Параллельное (concurrent) выполнение потоков.
8. Оптимизация фортрановских программ для CUDA.
9. Управление потоками и событиями.
10. Функция обработки ошибок. Калькулятор загруженности CUDA.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

### 4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

### 4.1.2.6. Примерный перечень тем курсовых работ (проектов)

1. Разработка распределенной вычислительной системы с использованием графических процессоров и технологии NVIDIA CUDA.
2. Разработка распределенной вычислительной системы с использованием графических процессоров и технологии OpenCL.

## 5. Образовательные технологии

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении практических работ применяется имитационный или симуляционный подход. Шаги решения задач студентам демонстрируются при помощи мультимедийной техники. В дальнейшем студенты самостоятельно решают аналогичные задания.

## **6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.**

Фонды оценочных материалов (средств) приведены в приложении.

## **7. Учебно-методическое и информационное обеспечение дисциплины.**

### **7.1. Основная учебно-методическая литература по дисциплине**

1. Параллельные вычисления на GPU. Архитектура и программная модель CUDA : учебное пособие / А. В. Боресков, А. А. Харламов, Н. Д. Марковский [и др.]. — Москва : Московский государственный университет имени М.В. Ломоносова, 2015. — 336 с. — ISBN 978-5-19-011058-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/54647.html> - <https://www.iprbookshop.ru/54647.html>

### **7.2. Дополнительная учебно-методическая литература по дисциплине**

1. Биллиг, В. А. Параллельные вычисления и многопоточное программирование : учебник / В. А. Биллиг. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 310 с. — ISBN 978-5-4497-0936-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102044.html> - <https://www.iprbookshop.ru/102044.html>

### **7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем**

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института ([www.mivlgu.ru/iop](http://www.mivlgu.ru/iop)), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

Информационно-образовательный портал МИ ВлГУ <https://www.mivlgu.ru/iop/>

Электронная библиотека ВлГУ <http://e.lib.vlsu.ru/>

Электронная библиотека «ЭВРИКА» <http://elib.mivlgu.local/>

Научная электронная библиотека "eLibrary" <http://elibrary.ru>

Программное обеспечение:

LibreOffice (Mozilla Public License v2.0)

Microsoft Visual Studio (Программа Microsoft Azure Dev Tools for Teaching (Order Number: IM126433))

Notepad++ (GNU GPL 3)

Wireshark (GNU GPL 2+)

StarUML (Proprietary commercial software (formerly GNU GPL))

### **7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

[iprbookshop.ru](http://iprbookshop.ru)

[mivlgu.ru](http://mivlgu.ru)

[e.lib.vlsu.ru](http://e.lib.vlsu.ru)

[elib.mivlgu.local](http://elib.mivlgu.local)

[elibrary.ru](http://elibrary.ru)

[mivlgu.ru/iop](http://mivlgu.ru/iop)

## **8. Материально-техническое обеспечение дисциплины**

Лекционная аудитория

Проектор ACER P1100 DLP Projector EMEA; Компьютер Celeron 1.8 GHz; Экран настенный; Акустическая система

Лаборатория программирования и лицензионного программного обеспечения

Компьютер Kraftway Credo KC 36 - 12 шт.; проектор NEC Projector VT595G; экран настенный; акустическая система.

## **9. Методические указания по освоению дисциплины**

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

На практических занятиях пройденный теоретический материал подкрепляется решением задач по основным темам дисциплины. Занятия проводятся в компьютерном классе, используя специальное программное обеспечение. Каждой подгруппе обучающихся преподаватель выдает задачу, связанную с разработкой и программной реализацией алгоритмов обработки информации. В конце занятия обучающие демонстрируют полученные результаты преподавателю и при необходимости делают работу над ошибками.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводится в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Курсовая работа выполняется в соответствии с методическими указаниями на курсовую работу. Обучающийся выбирает одну из указанных в перечне тем курсовых работ, исходя из своих интересов, наличия соответствующих литературных и иных источников. В ходе выполнения курсовой работы преподаватель проводит консультации обучающегося. На заключительном этапе обучающийся оформляет пояснительную записку к курсовой работе и выполняет ее защиту в присутствии комиссии из преподавателей кафедры.

Форма заключительного контроля при промежуточной аттестации – зачет. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.03.01 Информатика и вычислительная техника* и профилю подготовки *Вычислительные машины, комплексы, системы и сети*

Рабочую программу составил к.т.н., доцент, Колпаков А.А. \_\_\_\_\_

Программа рассмотрена и одобрена на заседании кафедры *ЭиВТ* протокол № 24 от 27.05.2020 года.

Заведующий кафедрой *ЭиВТ* \_\_\_\_\_ *Кропотов Ю.А.*

(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета ФРЭКС

протокол № 9 от 11.06.2020 года.

Председатель комиссии ФРЭКС \_\_\_\_\_

*Белов А.А.*

(Подпись)

**Лист актуализации рабочей программы дисциплины**

Программа одобрена на 2021/2022 учебный год.

Протокол заседания кафедры № 32 от 19.05.2021 года.

Заведующий кафедрой ЭиВТ \_\_\_\_\_ Белов А.А.  
(Подпись)

Программа одобрена на 2022/2023 учебный год.

Протокол заседания кафедры № 34 от 11.05.2022 года.

Заведующий кафедрой ЭиВТ \_\_\_\_\_ Белов А.А.  
(Подпись)

Программа одобрена на \_\_\_\_\_ учебный год.

Протокол заседания кафедры № \_\_\_\_\_ от \_\_\_\_\_ 20\_\_ года.

Заведующий кафедрой \_\_\_\_\_  
(Подпись) (Ф.И.О.)

**Фонд оценочных материалов (средств) по дисциплине**  
**Архитектура и программирование гетерогенных вычислительных систем**

## **1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине**

1. Выделить на GPU массив `arr` из 109 элементов типа `float` и инициализировать его с помощью ядра следующим образом: `arr[i] = sin((i%360)*Pi/180.0)`. Скопировать массив в память центрального процессора и посчитать ошибку `err = sum_i(abs(sin((i%360)*Pi/180.0) - arr[i]))/109`. Провести исследование зависимости результата от использования функций: `sin`, `sinf`, `__sinf` и ключа компиляции `-use-fast-math`. Объяснить результат. Проверить результат при использовании массива типа `double`.

2. Реализовать программу из задания 1 без использования CUDA API. Расчетные ядра должны быть собраны в отдельный модуль.

3. Реализовать программу для накладывания фильтров на изображения. Возможные фильтры: размытие, выделение границ, избавление от шума. Реализовать три варианта программы, а именно: на глобальной памяти, с применением разделяемой памяти и текстур. Сравнить время выполнения каждого из фильтров.

4. Модифицировать предыдущую программу так, чтобы использовались все имеющиеся в распоряжение программы GPU. Программа должна определять количество доступных графических процессоров и распределять работу по ним. Выполнить две реализации: каждый доступный GPU обрабатывает свою картинку, каждый GPU обрабатывает свою часть картинки. Программа в качестве входных

параметров принимает входную и выходную директорию с картинками.

5. Провести профилирование и оптимизацию одной из реализаций программы из п.4. Предоставить отчет о профилировании ядра, произвести оптимизацию программы, предоставить описание выполненных изменений кода и объяснение полученного результата.

6. При помощи метода наименьших квадратов найти окружность в изображении. Для каждой случайной выборки точек организовать их обработку на GPU. Случайные выборки организовать при помощи библиотеки CURAND.

7. Для заданной программы на Си или Фортран выполнить перенос части кода на графический процессор используя возможности компилятора PGI (директивы `OpenACC`). Студент может взять в качестве отправной точки пример из магистерской диссертации, предыдущих курсов или реализовать сам, например, решение уравнения теплопроводности в бруске с заданными параметрами.

8. Для заранее заданного кода провести его анализ, модификацию (в случае необходимости) и перенос расчетов (части расчетов в том случае, если это является законченным логическим блоком) на графический процессор любым из доступных студенту способом.

### **Общее распределение баллов текущего контроля по видам учебных работ для студентов**

Рейтинг-контроль 1	Устный опрос, 1 лабораторная работы	20
Рейтинг-контроль 2	Устный опрос, 1 лабораторная работы	20
Рейтинг-контроль 3	Устный опрос, 1 лабораторная работы, курсовая работа	40
Посещение занятий студентом		5

Дополнительные баллы (бонусы)		15
Выполнение семестрового плана самостоятельной работы		0

## **2. Промежуточная аттестация по дисциплине**

**Перечень вопросов к экзамену / зачету / зачету с оценкой.**

**Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)**

ПК-5:

Блок 1 (знать).

1. Синхронизация потоков и балансировка нагрузки.
2. Методы синхронизации.
3. Задачи, где применение распределенных вычислений наиболее востребовано
4. Целесообразность применения распределенных вычислений.
5. Закон Амдала.
6. Архитектурно-программная технология CUDA. Особенности, плюсы и минусы.

Блок 2 (уметь).

1. Основные отличия между GPU и CPU.
2. Архитектура GPU.
3. CUDA. Нити, блоки. Варп.
4. CUDA. Встроенные типы и переменные.
5. CUDA. Обработка ошибок.
6. CUDA. Атомарные операции.
7. CUDA. Типы памяти.
8. CUDA. Оптимизация CUDA программ.
9. CUDA. Библиотека CUBLAS.
10. CUDA. Библиотека CURAND.
11. Фреймворк OpenCL. Особенности, плюсы и минусы.
12. GRID-системы. Особенности, плюсы и минусы.

Блок 3 (владеть).

1. Возможности языка C# при работе с несколькими потоками и использовать несколько ядер процессора.
2. Возможности языка C# в распределённых вычислениях.
3. Облачные вычисления и распределённые БД.
4. Пиринговые сети.
5. Добровольные вычисления.

### **Методические материалы, характеризующие процедуры оценивания**

На основе типовых заданий программным комплексом информационно-образовательного портала МИ ВлГУ формируются в автоматическом режиме тестовые задания для студентов: три вопроса из блока 1, три вопроса из блока 2 и один вопрос из блока 3. Программный комплекс формирует индивидуальные задания для каждого зарегистрированного в системе студента и устанавливает время прохождения тестирования. Результатом тестирования является процент правильных ответов, с учетом индивидуального семестрового рейтинга студента формируется итоговый балл и выставление зачета по дисциплине.

При этом для подготовки к опросам, контрольным работам, а также при выполнении практических, лабораторных работ, курсового проектирования студентам рекомендуется пользоваться следующими методическими указаниями и курсом лекций:

- 1) Курс лекций доступен по ссылке: <https://www.mivlgu.ru/iop/course/view.php?id=454>

2) Методические указания для практических занятий доступны по ссылке:  
<https://www.mivlgu.ru/iop/course/view.php?id=454>

3) Методические указания для лабораторных занятий доступны по ссылке:  
<https://www.mivlgu.ru/iop/course/view.php?id=454>

4) Методические указания по курсовому проектированию доступны по ссылке:  
<https://www.mivlgu.ru/iop/mod/resource/view.php?id=4250>

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	<i><b>Уровень сформированности компетенций</b></i>
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	<i><b>Высокий уровень</b></i>
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	<i><b>Продвинутый уровень</b></i>
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	<i><b>Пороговый уровень</b></i>
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	<i><b>Компетенции не сформированы</b></i>

### 3. Задания в тестовой форме по дисциплине

Примеры заданий:

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.

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.

Выберите верные утверждения о программной архитектуре CUDA.

- +а) CUDA Host API – это функции, выполняющиеся только на CPU;
- б) программа компилируется и выполняется полностью на GPU;
- в) допускается одновременное использование Driver API и Runtime API;
- +г) Runtime API – это высокоуровневая реализация с помощью Driver API;

Что такое устройство в терминах CUDA?

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

Что такое хост?

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

Что такое ядро (kernel)?

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

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.

\_\_\_\_\_ это – единица выполнения программы в CUDA

\_\_\_\_\_ это – группа нитей с общей памятью в CUDA

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке <https://www.mivlgu.ru/iop/question/edit.php?courseid=454>

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.