

**Федеральное государственное образовательное  
бюджетное учреждение высшего образования**  
**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ**  
**ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**  
**(Финансовый университет)**

**Департамент анализа данных и машинного обучения**  
**Факультета информационных технологий и анализа больших данных**

**УТВЕРЖДАЮ**

Проректор по учебной  
и методической работе

\_\_\_\_\_ Е.А. Каменева

25.04.2023 г.

**Маковейчук К. А.**

**Технологии параллельного программирования**

**Рабочая программа дисциплины**  
для студентов, обучающихся по направлению подготовки  
09.03.03 - Прикладная информатика,  
ОП «Инженерия данных»,  
ОП «Прикладная информатика»,  
ОП «Прикладные информационные системы в экономике и финансах»

*Рекомендовано Ученым советом  
Факультета информационных технологий и анализа больших данных  
(протокол №31 от 18.04.2023г.)*

*Одобрено Советом учебно-научного  
Департамента анализа данных и машинного обучения  
(протокол №2 от 29.03.2023г.)*

**Москва 2023**

## СОДЕРЖАНИЕ

1. Наименование дисциплины .....	2
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине .....	2
3. Место дисциплины в структуре образовательных программ.....	5
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся .....	6
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	7
5.1. Содержание дисциплины.....	7
5.2. Учебно–тематический план.....	9
5.3. Содержание семинаров, практических занятий .....	12
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	13
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы .....	13
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю .....	14
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	15
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины .....	28
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	29
10. Методические указания для обучающихся по освоению дисциплины .....	31
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем .....	34
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	35

## 1. Наименование дисциплины

«Технологии параллельного программирования».

## 2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
<b>ОП «Прикладная информатика»</b>			
ПКП-3	Способность применять методы разработки приложений в сфере экономики и финансов на платформе корпоративных информационных систем	1. Демонстрирует знание назначения и функционал типовых модулей корпоративных информационных систем, основные методы разработки приложений на их платформе	<b>Знать</b> принципы организации и функционирования корпоративных информационных систем с учетом параллельных вычислений, основные методы разработки приложений на их платформе с применением технологий параллельного программирования и распределенных вычислений; <b>Уметь</b> применять современные методы разработки приложений на основе типовых модулей корпоративных информационных систем с использованием технологий параллельного программирования и распределенных вычислений
		2. Владеет методологией разработки приложений в сфере экономики и финансов на платформе корпоративных информационных систем	<b>Знать</b> принципы построения архитектуры программного обеспечения с учетом параллельных вычислений в сфере экономики и финансов на платформе корпоративных информационных систем; стандарты распараллеливания программ в системах с общей памятью и передачи сообщений в системах с распределенной памятью, директивы, функции, переменные среды; средства разработки для GPU; методы и средства проектирования программного обеспечения для высокопроизводительных и гетерогенных систем <b>Уметь</b> применять современные подходы и методологии из области параллельных вычислений и программирования при разработке на платформе корпоративных информацион-

			ных систем приложений в сфере экономики и финансов; использовать технологии и существующие типовые решения для проектирования программного обеспечения с учетом параллельных вычислений, такие, как OpenMP для систем с общей памятью, MPI для систем с распределенной памятью, PyOpenCL, PyOpenCuda для распараллеливания процессов на GPU
ПКП-5	Способность применять технологии разработки настольных, мобильных и web-приложений в сфере экономики и финансов	1. Демонстрирует знание технологии разработки настольных, мобильных и web-приложений	<b><u>Знать</u></b> теоретические основы применения технологий параллельного программирования при разработке настольных, мобильных и web-приложений в сфере экономики и финансов <b><u>Уметь</u></b> применять теоретические основы технологий параллельного программирования при разработке настольных, мобильных и web-приложений в сфере экономики и финансов
		2. Владеет навыками разработки настольных приложений в сфере экономики и финансов	<b><u>Знать</u></b> теоретические основы применения технологий параллельного программирования при разработке настольных приложений в сфере экономики и финансов <b><u>Уметь</u></b> применять теоретические основы технологий параллельного программирования при разработке настольных приложений в сфере экономики и финансов
		3. Владеет навыками разработки мобильных приложений в сфере экономики и финансов	<b><u>Знать</u></b> теоретические основы применения технологий параллельного программирования при разработке мобильных приложений в сфере экономики и финансов <b><u>Уметь</u></b> применять теоретические основы технологий параллельного программирования при разработке мобильных приложений в сфере экономики и финансов

		4. Владеет навыками разработки web- приложений в сфере экономики и финансов	<p><b><u>Знать</u></b> теоретические основы применения технологий параллельного программирования при разработке web- приложений в сфере экономики и финансов</p> <p><b><u>Уметь</u></b> применять теоретические основы технологий параллельного программирования при разработке web-приложений в сфере экономики и финансов</p>
<b>ОП «Инженерия данных», ОП «Прикладные информационные системы в экономике и финансах»</b>			
ПKN-3	Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней	1. Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов	<p><b><u>Знать</u></b> современные подходы, технологии и средства разработки для реализации алгоритмов и структур данных методами параллельного программирования</p> <p><b><u>Уметь</u></b> применять современные подходы, технологии и средства разработки для реализации алгоритмов и структур данных методами параллельного программирования</p>
		2. Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе	<p><b><u>Знать</u></b> принципы организации и функциональные и нефункциональные требования к информационным системам с учетом параллельных вычислений, принципы выбора архитектурных решений на их основе с применением технологий параллельного программирования и распределенных вычислений;</p> <p><b><u>Уметь</u></b> собирать, формулировать, систематизировать и анализировать функциональные и нефункциональные требования к информационным системам с учетом параллельных вычислений, выбирать архитектурные решения на их основе с использованием технологий параллельного программирования и распределенных вычислений</p>

		<p>3. Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии</p>	<p><b><u>Знать</u></b> принципы создания объектно-ориентированного кода с учетом параллельных вычислений, декомпозиции задачи и проектирования системы в пределах одной платформы или технологии; стандарты распараллеливания программ в системах с общей памятью и передачи сообщений в системах с распределенной памятью, директивы, функции, переменные среды; средства разработки для GPU; методы и средства проектирования программного обеспечения для высокопроизводительных и гетерогенных систем</p> <p><b><u>Уметь</u></b> применять современные подходы и методологии из области параллельных вычислений и программирования при разработке объектно-ориентированного кода, декомпозиции задачи и проектирования системы в пределах одной платформы или технологии; использовать технологии и существующие типовые решения для проектирования программного обеспечения с учетом параллельных вычислений, такие, как OpenMP для систем с общей памятью, MPI для систем с распределенной памятью, PyOpenCL, PyOpenCuda для распараллеливания процессов на GPU</p>
--	--	--	---

### 3. Место дисциплины в структуре образовательных программ

Дисциплина «Технологии параллельного программирования» является дисциплиной Цикла профиля (элективный) по направлению подготовки 09.03.03 - Прикладная информатика, ОП «Инженерия данных», ОП «Прикладная информатика», ОП «Прикладные информационные системы в экономике и финансах».

**4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся**

**ОП «Прикладная информатика» / ОП «Инженерия данных», ОП «Прикладные информационные системы в экономике и финансах»**

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 7 / 6 (в часах)
<b>Общая трудоёмкость дисциплины</b>	<b>3/108</b>	<b>108</b>
<b>Контактная работа - Аудиторные занятия</b>	<b>50</b>	<b>50</b>
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>34</i>	<i>34</i>
<b>Самостоятельная работа</b>	<b>58</b>	<b>58</b>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	экзамен / зачет	экзамен / зачет

**ОП «Прикладная информатика» / ОП «Прикладные информационные системы в экономике и финансах»**

очно-заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 8 / 7 (в часах)
<b>Общая трудоёмкость дисциплины</b>	<b>3/108</b>	<b>108</b>
<b>Контактная работа - Аудиторные занятия</b>	<b>34</b>	<b>34</b>
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>18</i>	<i>18</i>
<b>Самостоятельная работа</b>	<b>74</b>	<b>74</b>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	экзамен / зачет	экзамен / зачет

## ОП «Прикладные информационные системы в экономике и финансах»

Институт онлайн-образования, заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 8 (в часах)
<b>Общая трудоёмкость дисциплины</b>	<b>3/108</b>	<b>108</b>
<i>Контактная работа - Аудиторные занятия</i>	<i>12</i>	<i>12</i>
<i>Лекции</i>	<i>4</i>	<i>4</i>
<i>Семинары, практические занятия</i>	<i>8</i>	<i>8</i>
<b>Самостоятельная работа</b>	<b>96</b>	<b>96</b>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	зачет	зачет

### 5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

#### 5.1. Содержание дисциплины

##### Раздел 1. Введение в «Технологии параллельного программирования».

История развития параллелизма в архитектуре ЭВМ. Конвейерная обработка данных и команд. Многофункциональная обработка периферийными процессорами. Векторная обработка. Суперкомпьютеры. Классификация вычислительных систем Флинна. Вычислительные системы с общей и с распределенной памятью. Обзор задач, требующих суперкомпьютеров. Технологии параллелизма. Иерархическая организация памяти в компьютере. Локальность по обращению. Анализ задачи умножения матриц.

##### Раздел 2. Основы OpenMP.

Особенности программирования для систем с общей памятью. Понятие процесса, потока и многопоточности. Технология OpenMP, ее особенности и компоненты. Основные сведения о спецификациях набора стандарта OpenMP. Директивы параллельного программирования. Модели памяти и классы переменных в OpenMP. Режимы выполнения многопоточных программ. Вложенный параллелизм.



### **Раздел 3. Директивы распределения и синхронизации работы OpenMP.**

Распараллеливание выполнения циклов. Распределение работы на основе независимых задач. Синхронизация выполнения различных потоков. Распределение операторов одного структурного блока между нитями. Копирование значений частных переменных.

### **Раздел 4. Основы векторизации исполняемого кода в OpenMP.**

Проблемы векторизации. Способы векторизации. Директивы для работы с параллелизмом по данным (векторизация) в OpenMP 4.0.

### **Раздел 5. Технологии параллельного программирования для графических процессоров (GPU).**

Возможные преимущества вычислений на графическом процессоре. Задача компьютерной визуализации трехмерных сцен. Архитектура графического процессора (GPU). Средства разработки для графического процессора. Python в работе с GPU. Основы неоднородных (гетерогенных) вычислений. Уровни параллельности в архитектуре GPU. Гетерогенное программирование на PyOpenCL. Построение приложений при помощи PyOpenCL и PyCUDA.

### **Раздел 6. Основы технологии MPI.**

Особенности программирования для систем с распределенной памятью. Технология MPI. Основы MPI на Python. Синхронные и асинхронные примитивы взаимодействия. Операции типа точка-точка. Операции коллективного взаимодействия. Параллельный алгоритм умножения матрицы на вектор. Операции сочетания параллельных результатов вычислений. Примитивы синхронизации между процессами. Закон Амдала. Оценки предельно возможного ускорения параллельных алгоритмов.

## 5.2. Учебно–тематический план

**ОП «Инженерия данных», ОП «Прикладная информатика»,  
ОП «Прикладные информационные системы в экономике и финансах»  
очная форма обучения**

№ п/ п	Наименование тем (разделов) дисциплины	Трудоёмкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самост оательн ая работа	
			Общая, в т.ч.:	Лекц ии	Семинары, практическ ие занятия		
1.	Введение в «Тех- нологии парал- лельного програм- мирования»	12	4	2	2	8	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
2.	Основы OpenMP	22	10	2	8	12	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
3.	Директивы рас- пределения и син- хронизации ра- боты OpenMP	14	6	2	4	8	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
4.	Основы векториза- ции исполняемого кода в OpenMP	14	6	2	4	8	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
5.	Технологии парал- лельного програм- мирования для графических про- цессоров (GPU)	24	12	4	8	12	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
6.	Основы техноло- гии MPI	22	12	4	8	10	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
	В целом по дисциплине	108	50	16	34	58	Согласно учеб- ному плану: кон- трольная работа
	Итого в %		46	32	68	54	

**ОП «Прикладная информатика»,**  
**ОП «Прикладные информационные системы в экономике и финансах»**  
очно-заочная форма обучения

№ п/ п	Наименование тем (разделов) дисциплины	Трудоёмкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самост оательн ая работа	
			Общая, в т.ч.:	Лекц ии	Семинары, практическ ие занятия		
1.	Введение в «Тех- нологии парал- лельного програм- мирования»	14	4	2	2	10	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
2.	Основы OpenMP	20	6	2	4	14	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
3.	Директивы рас- пределения и син- хронизации ра- боты OpenMP	16	4	2	2	12	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
4.	Основы векториза- ции исполняемого кода в OpenMP	16	4	2	2	12	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
5.	Технологии парал- лельного програм- мирования для графических про- цессоров (GPU)	22	8	4	4	14	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
6.	Основы техноло- гии MPI	20	8	4	4	12	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
	В целом по дисциплине	108	34	16	18	74	Согласно учеб- ному плану: кон- трольная работа
	Итого в %		31	47	53	69	

# ОП «Прикладные информационные системы в экономике и финансах»

Институт онлайн-образования, заочная форма обучения

№ п/ п	Наименование тем (разделов) дисциплины	Трудоёмкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа- Аудиторная работа			Самост оательн ая работа	
			Общая, в т.ч.:	Лекц ии	Семинары, практическ ие занятия		
1.	Введение в «Тех- нологии парал- лельного програм- мирования»	15	1	1	0	14	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
2.	Основы OpenMP	20	2	1	1	18	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
3.	Директивы рас- пределения и син- хронизации ра- боты OpenMP	19	3	1	2	16	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
4.	Основы векториза- ции исполняемого кода в OpenMP	15	1	0	1	14	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
5.	Технологии парал- лельного програм- мирования для графических про- цессоров (GPU)	21	3	1	2	18	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
6.	Основы техноло- гии MPI	18	2	0	2	16	Самостоятельные работы. Участие в решении задач на практических заня- тиях. Обсуждение решенных задач.
	В целом по дисциплине	108	12	4	8	96	Согласно учеб- ному плану: кон- трольная работа
	Итого в %		11	33	67	89	

### 5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
Введение в «Технологии параллельного программирования»	Суперкомпьютеры. Классификация вычислительных систем Флинна. Технологии параллелизма. Иерархическая организация памяти в компьютере. Локальность по обращению. Анализ задачи умножения матриц. <i>Рекомендуемые источники: 8. 1-4</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Дискуссия «Суперкомпьютеры – ТОП 100: анализ»
Основы OpenMP	Основные сведения о спецификациях набора стандарта OpenMP. Директивы параллельного программирования. Модели памяти и классы переменных в OpenMP. Режимы выполнения многопоточных программ. Вложенный параллелизм. <i>Рекомендуемые источники: 8. 1-4</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Выполнение тестовых заданий
Директивы распределения и синхронизации работы OpenMP	Распараллеливание выполнения циклов. Распределение работы на основе независимых задач. Синхронизация выполнения различных потоков. Распределение операторов одного структурного блока между нитями. Копирование значений приватных переменных. <i>Рекомендуемые источники: 8. 1-4</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Выполнение тестовых заданий
Основы векторизации исполняемого кода в OpenMP	Директивы для работы с параллелизмом по данным (векторизация) в OpenMP 4.0. <i>Рекомендуемые источники: 8. 1-4</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений
Технологии параллельного программирования для графических процессоров (GPU)	Python в работе с GPU. Основы неоднородных (гетерогенных) вычислений. Уровни параллельности в архитектуре GPU. Гетерогенное программирование на PyOpenCL. Построение приложений при помощи PyOpenCL и PyCUDA. <i>Рекомендуемые источники: 8. 1-4</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Выполнение тестовых заданий
Основы технологии MPI	Основы MPI на Python. Синхронные и асинхронные примитивы взаимодействия. Операции типа точка-точка. Операции коллективного взаимодействия. Параллельный алгоритм умножения матрицы на вектор.	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Выполнение тестовых и контрольных заданий

	Операции сочетания параллельных результатов вычислений. Примитивы синхронизации между процессами. <i>Рекомендуемые источники: 8. 1-4</i>	
--	--	--

## 6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

### 6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в «Технологии параллельного программирования»	История развития параллелизма в архитектуре ЭВМ. Конвейерная обработка данных и команд. Многофункциональная обработка периферийными процессорами. Векторная обработка. Вычислительные системы с общей и с распределенной памятью. Обзор задач, требующих суперкомпьютеров.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.
Основы OpenMP	Особенности программирования для систем с общей памятью. Понятие процесса, потока и многопоточности. Технология OpenMP, ее особенности и компоненты.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.
Директивы распределения и синхронизации работы OpenMP	Примеры задач, допускающих распараллеливание выполнения циклов. Сравнение эффективности распараллеливания циклов (директива For) и распределения работы на основе независимых задач (директива Task) на примере различных задач.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.
Основы векторизации исполняемого кода в OpenMP	Проблемы векторизации. Способы векторизации.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.
Технологии параллельного программирования для графических процессоров (GPU)	Возможные преимущества вычислений на графическом процессоре. Задача компьютерной визуализации трехмерных сцен. Архитектура графического процессора (GPU).	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.

	Средства разработки для графического процессора.	
Основы технологии MPI	Особенности программирования для систем с распределенной памятью. Технология MPI. Закон Амдала. Оценки предельно возможного ускорения параллельных алгоритмов.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.

## 6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

### Примерные задания контрольной работы

**Задание 1.** После выполнения выделенного параллельного участка в программе происходит неявная барьерная синхронизация параллельно работающих потоков. Их дальнейшее выполнение происходит только тогда, когда все они достигнут данной точки. Если в подобной задержке нет необходимости, опция `nowait` позволяет потокам, уже дошедшим до конца участка, продолжить выполнение без синхронизации с остальными. Реализуйте программу, иллюстрирующую описанную ситуацию.

Реализуйте программу на C++ с подключением технологии OpenMP, и проверьте получаемый результат, обоснуйте результат в выводах.

Можно использовать компилятор g++ совместно с установленной для него реализацией технологии OpenMP, либо IDE MS Visual Studio Community с настройками, разрешающими OpenMP, в свойствах проекта, либо использовать любой другой доступный компилятор C++ с OpenMP, в том числе облачный, например, <https://www.codingame.com/playgrounds/54443/openmp/playground>

Пример реализации:

```

#pragma omp parallel num_threads(4)
{
    printf("Перед директивой single без nowait \n");
#pragma omp single
    {
        printf("В директиве single \n");
    }
    printf("После директивы single без nowait. Это сообщение никогда не будет раньше предыдущих \n");
#pragma omp barrier
    printf("Перед директивой single с nowait \n");
#pragma omp single nowait
    {
        printf("В директиве single \n");
    }
    printf("После директивы single с no-wait. Это сообщение может быть раньше предыдущих \n");
}

```

**Задание 2.** Выполнить умножение матриц, используя любые изученные технологии распараллеливания и ускорения вычислений (OpenMP, Мpi4py, PyOpenCL). Сравнить производительность программы для матриц разного размера и при разном количестве потоков (для OpenMP); рабочих процессов (для MPI), для CPU и GPU (для PyOpenCL).

*Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Департамента анализа данных и машинного обучения.*

## **7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

Перечень компетенций, формируемых в процессе освоения дисциплины, содержится в разделе 2 «Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».



## Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
<b>ОП «Прикладная информатика»</b>			
ПКП-3. Способность применять методы разработки приложений в сфере экономики и финансов на платформе корпоративных информационных систем	1. Демонстрирует знание назначения и функционал типовых модулей корпоративных информационных систем, основные методы разработки приложений на их платформе	<p><b>Знать</b> принципы организации и функционирования корпоративных информационных систем с учетом параллельных вычислений, основные методы разработки приложений на их платформе с применением технологий параллельного программирования и распределенных вычислений;</p> <p><b>Уметь</b> применять современные методы разработки приложений на основе типовых модулей корпоративных информационных систем с использованием технологий параллельного программирования и распределенных вычислений</p>	Выполнить обследование задач финансового предприятия «Зета» в соответствии принципами организации информационных систем. Оценить задачи с точки зрения применимости алгоритмов параллельных вычислений.
	2. Владеет методологией разработки приложений в сфере экономики и финансов на платформе корпоративных информационных систем	<p><b>Знать</b> принципы построения архитектуры программного обеспечения с учетом параллельных вычислений в сфере экономики и финансов на платформе корпоративных информационных систем; стандарты распараллеливания программ в системах с общей памятью и передачи сообщений в системах с распределенной памятью, директивы, функции, переменные среды; средства разработки для GPU; методы и средства проектирования программного обеспечения для высокопроизводительных и гетерогенных систем</p> <p><b>Уметь</b> применять современные подходы и методологии из области параллельных вычислений и программирования при разработке на платформе корпоративных информационных систем приложений в сфере эко-</p>	Реализовать постановку задачи, алгоритм и программу решения задачи обработки условных финансовых данных с помощью итерационного метода решения системы линейных алгебраических уравнений (СЛАУ) как умножение матрицы на вектор с использованием технологий параллельного программирования и распределенных вычислений (возможные технологии для реализации программы: C++ & OpenMP   Python + PyOpenCL   Python + mpi4py)

		номики и финансов; использовать технологии и существующие типовые решения для проектирования программного обеспечения с учетом параллельных вычислений, такие, как OpenMP для систем с общей памятью, MPI для систем с распределенной памятью, PyOpenCL, PyOpenCuda для распараллеливания процессов на GPU	
ПКП-5. Способность применять технологии разработки настольных, мобильных и web-приложений в сфере экономики и финансов	1. Демонстрирует знание технологии разработки настольных, мобильных и web-приложений	<b>Знать</b> теоретические основы применения технологий параллельного программирования при разработке настольных, мобильных и web-приложений в сфере экономики и финансов <b>Уметь</b> применять теоретические основы технологий параллельного программирования при разработке настольных, мобильных и web-приложений в сфере экономики и финансов	Определить инструментальные средства информационных технологий для реализации программы построения фрактала Ньютона для уравнения $x^3-1=0$ Обосновать применение технологий параллельного программирования при разработке приложения.
	2. Владеет навыками разработки настольных приложений в сфере экономики и финансов	<b>Знать</b> теоретические основы применения технологий параллельного программирования при разработке настольных приложений в сфере экономики и финансов <b>Уметь</b> применять теоретические основы технологий параллельного программирования при разработке настольных приложений в сфере экономики и финансов	Реализовать постановку задачи, алгоритм и настольное приложение для решения задачи построения фрактала Ньютона для уравнения $x^3-1=0$ Применить в решении задачи технологии параллельного программирования.
	3. Владеет навыками разработки мобильных приложений в сфере экономики и финансов	<b>Знать</b> теоретические основы применения технологий параллельного программирования при разработке мобильных приложений в сфере экономики и финансов <b>Уметь</b> применять теоретические основы технологий параллельного программирования при разработке мобильных приложений в сфере экономики и финансов	Реализовать постановку задачи, алгоритм и мобильное приложение для решения задачи построения фрактала Ньютона для уравнения $x^3-1=0$ Применить в решении задачи технологии параллельного программирования.

	4. Владеет навыками разработки web- приложений в сфере экономики и финансов	<p><b><u>Знать</u></b> теоретические основы применения технологий параллельного программирования при разработке web- приложений в сфере экономики и финансов</p> <p><b><u>Уметь</u></b> применять теоретические основы технологий параллельного программирования при разработке web- приложений в сфере экономики и финансов</p>	<p>Реализовать постановку задачи, алгоритм и web-приложение для решения задачи построения фрактала Ньютона для уравнения <math>x^3-1=0</math></p> <p>Применить в решении задачи технологии параллельного программирования.</p>
<b>ОП «Инженерия данных», ОП «Прикладные информационные системы в экономике и финансах»</b>			
ПКН-3. Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней	1. Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов	<p><b><u>Знать</u></b> современные подходы, технологии и средства разработки для реализации алгоритмов и структур данных методами параллельного программирования</p> <p><b><u>Уметь</u></b> применять современные подходы, технологии и средства разработки для реализации алгоритмов и структур данных методами параллельного программирования</p>	<p>Реализовать структуру данных и алгоритм умножения векторов в Python, с использованием GPU (библиотека PyOpenCL или PyOpenCuda), либо с использованием библиотеки mpi4py. Сравнить два метода параллельного программирования.</p>
	2. Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе	<p><b><u>Знать</u></b> принципы организации и функциональные и нефункциональные требования к информационным системам с учетом параллельных вычислений, принципы выбора архитектурных решений на их основе с применением технологий параллельного программирования и распределенных вычислений;</p> <p><b><u>Уметь</u></b> собирать, формулировать, систематизировать и анализировать функциональные и нефункциональные требования к информационным системам с учетом параллельных вычислений, выбирать архитектурные решения на их основе с использованием технологий параллельного программирования и распределенных вычислений</p>	<p>Определить спецификацию функциональных и нефункциональных требований к приложению для расчета прогноза погоды на основании 5 матриц исходных данных для точек атмосферы на 4 высотах, с применением технологий параллельного программирования и распределенных вычислений.</p>

	<p>3. Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии</p>	<p><b><u>Знать</u></b> принципы создания объектно-ориентированного кода с учетом параллельных вычислений, декомпозиции задачи и проектирования системы в пределах одной платформы или технологии; стандарты распараллеливания программ в системах с общей памятью и передачи сообщений в системах с распределенной памятью, директивы, функции, переменные среды; средства разработки для GPU; методы и средства проектирования программного обеспечения для высокопроизводительных и гетерогенных систем</p> <p><b><u>Уметь</u></b> применять современные подходы и методологии из области параллельных вычислений и программирования при разработке объектно-ориентированного кода, декомпозиции задачи и проектирования системы в пределах одной платформы или технологии; использовать технологии и существующие типовые решения для проектирования программного обеспечения с учетом параллельных вычислений, такие, как OpenMP для систем с общей памятью, MPI для систем с распределенной памятью, PyOpenCL, PyOpenCuda для распараллеливания процессов на GPU</p>	<p>Разработать приложение для расчета прогноза погоды на основании 5 матриц исходных данных для точек атмосферы на 4 высотах, с применением технологий параллельного программирования и распределенных вычислений. Прогноз условно выполняется на основании решения уравнения <math>5x^5 + 3x^4 - 7x^3 + 2x^2 - 8x + 15 = y</math>, где <math>x</math> – данные матриц по каждой высоте.</p> <p>Выполнить разработку с применением параллельных вычислений и выбранной технологии, такой, как OpenMP, MPI, PyOpenCL или PyOpenCuda для распараллеливания процессов на GPU</p>
--	--	---	---

## **Примеры тестовых заданий**

**1. Какие подходы, применяемые в архитектурах компьютеров, позволяли повысить их производительность**

- Конвейерный способ обработки
- Независимые функциональные устройства
- Векторные операции верно
- Уменьшение энергопотребления

**2. Какой принцип организации памяти используется для обеспечения требуемой емкости и высокого быстродействия памяти компьютера за приемлемую цену**

- Интенсивный
- Экстенсивный
- Масштабируемости
- Иерархический

**3. Самый минимальный по объему тип памяти, но и самый быстрый это**

- Регистры центрального процессора
- Кэш первого уровня
- Оперативная память
- Кэш нулевого уровня

**4. Для эффективного использования иерархической структуры памяти при написании программ необходимо использовать**

- Принцип локальность по обращению
- Принципы объектного-ориентированного программирования
- Принцип случайного обращения
- Принцип маленьких данных

## **5. Классификация Флинна основывается на**

- на понятии потока
- на понятии декомпозиции данных
- на особенностях процессоров
- на особенностях организации памяти

## **6. SISD это**

- одиночный поток команд и одиночный поток данных
- множественный поток команд и одиночный поток данных
- одиночный поток команд и множественный поток данных
- множественный поток команд и множественный поток данных

## **7. Какие два крупных класса параллельных высокопроизводительных вычислительных систем можно выделить?**

- вычислительные системы с общей памятью
- вычислительные системы с распределенной памятью
- вычислительные системы с очень большим объемом оперативной памяти
- вычислительные системы с водяным охлаждением

## **8. Какие из этих технологий позволяют создавать параллельные программы**

- OpenMP
- MPI
- Pascal
- Parallel

## **9. Многопоточность это**

- свойство платформы или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков, выполняющихся «параллельно», то есть без предписанного порядка во времени.

- свойство центрального процессора автоматически выполнять программы сразу в несколько потоков
- свойство компилятора позволяющие создавать параллельные многопоточковые программы.

## **10. Технология параллельного программирования OpenMP включается в себя**

- директивы компилятора
- библиотечные функции
- переменные среды окружения
- параллельный компилятор

## **11. Для создания параллельных программ в OpenMP применяется модель программирования**

- Разветвление-слияние (Fork–join)
- Декомпозиция по данным
- Главный подчиненный (Master-slave)

## **12. Выберите код в котором создается параллельная область с 5 дополнительными потоками выполнения**

```
#pragma omp parallel shared(a,b,c) num_threads(4)
    #pragma omp for
    for (int i = 0; i < n; i++)
    {
        c[i] = a[i] + b[i];
    }
```

```
pragma omp parallel shared(a,b,c) num_threads(6)
    #pragma omp for
    for (int i = 0; i < n; i++)
    {
        c[i] = a[i] + b[i];
    }
```

```

#pragma omp parallel shared(a,b,c) num_threads(5)
    #pragma omp for
    for (int i = 0; i < n; i++)
    {
        c[i] = a[i] + b[i];
    }

```

**13. Следующий фрагмент кода запустили на компьютере с 8 ядерным процессором**

**Сколько раз будет выведено слово Print?**

```

#include "stdio.h"
#include "omp.h"

int main()
{
    printf("Print\n");
    #pragma omp parallel num_threads(10)
    {
        printf("Print\n");
    }
    #pragma omp parallel num_threads(2)
    {
        printf("Print\n");
    }
    printf("Print\n");
}

```

**14. Какое значение переменной n будет выведено на экран в следующей программе при запуске на компьютере с 4-х ядерным процессором?**

```

#include "stdio.h"
#include "omp.h"

int main()
{
    printf("Print\n");
    int n = 3;
    #pragma omp parallel num_threads(40)
    {
        n += 1;
    }
    printf("n=%d", n);
}

```

Значение не определено, от запуска к запуску может меняться

Программа не запустится, так как потоков больше чем ядер

Значение 43 будет выведено

Программа не скомпилируется из-за синтаксической ошибки



## 15. Выберите два основных класса переменных в OpenMP

- общие
- локальные
- глобальные
- независимые

16. В результате выполнения следующей программы на компьютере с 4 ядерным процессором на экран будут выведены числа. Впишите каждое из чисел в отдельное окно ввода

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

int main()
{
    int n=0;
    omp_set_nested(1);
    #pragma omp parallel num_threads(2) reduction(+:n)
    {
        n = 0;
        #pragma omp parallel num_threads(2) reduction(+:n)
        {
            n = 1;
        }
    }
    printf("%d ", n);
    n = 0;
    omp_set_nested(0);
    #pragma omp parallel num_threads(2) reduction(+:n)
    {
        n = 0;
        #pragma omp parallel num_threads(2) reduction(+:n)
        {
            n = 1;
        }
    }
    printf("%d ", n);
}
```

17. Если несколько потоков одновременно записывают значение общей переменной без выполнения синхронизации, то

- возникает ситуации «гонки данных» (data race)
- возникает ситуации не разделенного доступа
- в данном случаи нет никаких особенных ситуаций

**18. В результате выполнения данной программы на компьютере с восьмиядерным процессором слов «Print» будет выведено (ввести числом)**

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

int main()
{
    int n=1;
    omp_set_nested(1);
    omp_set_num_threads(2);
    #pragma omp parallel if (n>10)
    {
        printf("Print ");
    }
}
```

**19. Для выделения независимой задачи, которая может исполняться параллельно используется директива**

#pragma omp task

#pragma omp section

#pragma omp doit

**20. Какое максимальное ускорение (ускорение = время на 1 ядре / время на n ядрах ) возможно в следующей программе при запуске на компьютере с 8 ядерным процессором**

```
void sub_a(float **a, float **b, float **c, float **d, int n, int m)
{
    int i, j;
    #pragma omp parallel shared(a,b,c,d,n,m) private(i,j)
    {
        #pragma omp sections nowait
        {
            #pragma omp section
            for (i = 1; i < n; i++)
            {
                for (j = 0; j <= i; j++)
                    b[j][i] = (a[j][i] + a[j][i - 1]) / 2.;
            }
            #pragma omp section
            for (i = 1; i < m; i++)
            {
                for (j = 0; j <= i; j++)
                    d[j][i] = (c[j][i] + c[j][i - 1]) / 2.;
            }
        }
    }
}
```

Максимальное ускорение зависит от количества используемых ядер

Максимальное ускорение 8

Максимальное ускорение 2

### **Примерные вопросы для подготовки к зачету**

1. Что такое процесс и поток в современных операционных системах.
2. Для чего необходима синхронизация в многопоточных программах.
3. В каких состояниях может находиться поток.
4. В результате каких действий поток может переходить из одного состояния в другое.
5. Когда необходимо применять синхронизованные методы.
6. Какие примитивы позволяют осуществить обмен сообщениями между процессами.
7. Примитивы коллективного взаимодействия процессов.
8. Примитивы синхронизации между процессами.
9. Директивы распределения работы и синхронизации работы.
10. Распараллеливание выполнения циклов.
11. Распределение нескольких структурных блоков между потоками
12. Распределение работы на основе независимых задач
13. Синхронизация выполнения различных потоков. Простые директивы
14. Векторные вычисления с помощью OpenMP 4.0
15. Анализ и оптимизация программ с использованием современных компиляторов
16. Архитектура графического процессора (GPU).
17. Возможные преимущества вычислений на графическом процессоре.
18. Средства разработки для графического процессора.
19. Закон Амдала.
20. Классификация вычислительных систем Флинна.

## **Примерные вопросы для подготовки к экзамену**

1. История появления параллельной и высокопроизводительной обработки информации. Суперкомпьютеры.
2. Классификация вычислительных систем Флинна. Примеры.
3. Технологии параллелизма. Классификация, примеры.
4. Принцип иерархической организации памяти в компьютере. Примеры.
5. Локальность по обращению. Принципы локальности памяти.
6. Анализ задачи умножения матриц.
7. Основные сведения о спецификациях набора стандарта OpenMP.
8. Директивы параллельного программирования OpenMP.
9. Модели памяти и классы переменных в OpenMP.
10. Режимы выполнения многопоточных программ.
11. Вложенный параллелизм в OpenMP.
12. Директивы распределения и синхронизации работы OpenMP
13. Распараллеливание выполнения циклов в OpenMP: директива и принцип организации.
14. Распределение работы на основе независимых задач в OpenMP: директива и принцип организации.
15. Синхронизация выполнения различных потоков.
16. Распределение операторов одного структурного блока между нитями.
17. Копирование значений приватных переменных.
18. Основы векторизации исполняемого кода в OpenMP
19. Директивы для работы с параллелизмом по данным (векторизация) в OpenMP 4.0.
20. Технологии параллельного программирования для графических процессоров (GPU)
21. Python в работе с GPU: реализации библиотек, сравнение.
22. Основы неоднородных (гетерогенных) вычислений.
23. Уровни параллельности в архитектуре GPU.
24. Гетерогенное программирование на PyOpenCL и PyCUDA.
25. Построение приложений при помощи PyOpenCL и PyCUDA.

26. Основы технологии MPI
27. Основы MPI для Python.
28. Синхронные и асинхронные примитивы взаимодействия MPI.
29. Операции типа точка-точка в MPI.
30. Операции коллективного взаимодействия в MPI.
31. Параллельный алгоритм умножения матрицы на вектор: реализации на MPI.
32. Операции сочетания параллельных результатов вычислений в MPI.
33. Примитивы синхронизации между процессами в MPI.

### **Пример экзаменационного билета**

**Задание 1.** Теоретический вопрос. Директивы распределения и синхронизации работы OpenMP *(20 баллов)*

**Задание 2.** Практический вопрос. Реализовать умножение векторов в Python, написать две функции реализации: для CPU и GPU (библиотека PyOpenCL). Сравнить время выполнения CPU, GPU, kernel. *(20 баллов)*

**Задание 3. Практический вопрос.** Реализовать функцию умножения векторов в Python, с использованием библиотеки mpi4py. Сравнить время выполнения функции с временем выполнения функций из задания 2. *(20 баллов)*

### **8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

#### **а) основная:**

1. Малявко, А. А. Параллельное программирование на основе технологий openmp, mpi, cuda : учебное пособие для академического бакалавриата / А. А. Малявко. — 2-е изд., испр. и доп. — Москва : Юрайт, 2023. — 129 с. — ЭБС Юрайт. — URL: <https://biblio-online.ru/bcode/514199> (дата обращения: 19.05.2023). — Текст : электронный.

2. Арыков, С. Б. Параллельное программирование над общей памятью. OpenMP : учебное пособие / С. Б. Арыков, М. А. Городничев, Г. А. Щукин. — Новосибирск : НГТУ, 2019. — 95 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/152252> (дата обращения: 19.05.2023). — Текст : электронный.

**б) дополнительная:**

3. Карепова, Е. Д. Основы многопоточного и параллельного программирования: учебное пособие / Е. Д. Карепова. – Красноярск : Сиб. Федер. ун-т, 2016. — 356 с. — ЭБС ZNANIUM.com. — URL: <http://znanium.com/catalog/product/966962> (дата обращения: 19.05.2023). — Текст : электронный.

4. Филатов, А. С. Параллельное программирование : учебное пособие / А. С. Филатов. — Москва : РТУ МИРЭА, 2021. — 46 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/218429> (дата обращения: 19.05.2023). — Текст : электронный.

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

1. <http://www.fa.ru> – официальный сайт Финансового университета при Правительстве Российской Федерации

2. <http://www.library.fa.ru/> - Библиотечно-информационный комплекс Финуниверситета

3. Электронно-библиотечная система BOOK.RU <http://www.book.ru>

4. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>

5. Электронно-библиотечная система Znanium <http://www.znanium.com>

6. Электронно-библиотечная система издательства «ЮРАЙТ» <https://urait.ru/>

7. Электронно-библиотечная система издательства Проспект <http://ebs.prospekt.org/books>

8. Электронно-библиотечная система издательства Лань <https://e.lanbook.com/>

9. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
10. Электронная библиотека Издательского дома «Гребенников»  
<https://grebennikon.ru/>
11. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
12. Национальная электронная библиотека <http://нэб.рф/>
13. <https://org.fa.ru> - личный кабинет обучающегося
14. <https://org.fa.ru/app/ebs/list> - реестр ЭБС Финансового университета
15. <https://openedu.ru/course/msu/PARPROG/> - онлайн-курс «Суперкомпьютеры и параллельная обработка данных» на национальной платформе «Открытое образование» (курс МГУ им. М. В. Ломоносова)
16. <https://mooc.tsu.ru/mooc-openedu/mpi/> - онлайн-курс «Параллельное программирование с использованием OpenMP и MPI» на платформе MOOK Томского государственного университета
17. <https://stepik.org/course/115024/promo?search=2147452613> - онлайн-курс «Введение в параллельное программирование (OpenMP и MPI)» на платформе Stepik (Томский государственный университет, альтернативная платформа)
18. <https://teach-in.ru/course/parallel-computing-lukyanenko/lecture> - онлайн-курс «Параллельные вычисления» на платформе «Лекторий Teach-in» (лекции ученых МГУ им. М. В. Ломоносова)
19. <https://campus.fa.ru/course/view.php?id=15769> – базовый ЭУК по дисциплине «Технологии параллельного программирования» Финансового университета
20. <http://onreader.mdl.ru/PythonParallelProgrammingCookbook.2nd/content/index.html> - Законе Джанкарло. Книга рецептов параллельного программирования Python. 2е изд. Copyright © 2019 Packt Publishing. Документ предоставляется по лицензии Creative Commons Attribution 3.0 License.
21. [https://registry.khronos.org/OpenCL/specs/3.0-unified/html/OpenCL\\_API.html](https://registry.khronos.org/OpenCL/specs/3.0-unified/html/OpenCL_API.html) - The OpenCL™ Specification Khronos® OpenCL Working Group Version V3.0.14, Mon, 17 Apr 2023
22. <https://github.com/inducer> - Andreas Klöckner's web page PyOpenCL
23. <https://documen.tician.de/pyopencl/> - PyOpenCL's documentation

24. <https://pypi.org/project/mmpi4py/> - Python bindings for MPI
25. <https://mpi4py.readthedocs.io/en/stable/index.html> - MPI for Python

## **10. Методические указания для обучающихся по освоению дисциплины**

Цель методических рекомендаций – обеспечить студенту бакалавриата (далее – студенту) оптимальную организацию процесса изучения дисциплины, а также выполнения различных форм самостоятельной работы.

### ***Методические рекомендации по изучению дисциплины***

Студентам необходимо ознакомиться содержанием рабочей программы дисциплины (далее – РПД), с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, списком рекомендованной литературы и перечнем ресурсов сети «Интернет», необходимыми для освоения дисциплины. Часть разделов дисциплины студенты могут освоить с помощью бесплатных онлайн-курсов (с получением сертификата или иного документа об окончании), указанных в перечне ресурсов. Лекции, задания для семинарских и самостоятельных занятий, презентации и другие материалы по дисциплине, представленные в базовом ЭУК по дисциплине на образовательном портале Финансового университета, студенты изучают в процессе освоения курса в течение семестра. На сайте департамента студенты могут ознакомиться в течение семестра с графиком консультаций преподавателей.

### ***Рекомендации по подготовке к лекционным занятиям (теоретический курс)***

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет. Именно поэтому контроль над систематической работой студентов всегда находится в центре внимания департамента.

Лекционные занятия проводятся в соответствии с тематическим планом, при изложении материала рекомендуется использовать презентации в среде PowerPoint и фрагменты печатных материалов по теме лекции.

Студентам рекомендуется:



- перед каждой лекцией просматривать рабочую программу дисциплины, что позволит сэкономить время на записывание темы лекции, ее основных вопросов, рекомендуемой литературы;

- на отдельные лекции приносить соответствующий материал на бумажных или электронных носителях, представленный лектором на портале. Данный материал будет охарактеризован, прокомментирован, дополнен непосредственно на лекции;

- перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам, если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на практических занятиях. Не оставляйте «белых пятен» в освоении материала.

### ***Рекомендации по подготовке к практическим (семинарским) занятиям***

Проведение практических занятий осуществляется в компьютерных классах.

Студентам следует:

- приносить с собой на бумажных или электронных носителях рекомендованную преподавателем литературу к конкретному занятию;

- до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;

- при подготовке к практическим занятиям следует обязательно использовать не только лекции, но и другую учебную литературу;

- в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении, при решении задач, заданных для самостоятельного решения;

- в ходе семинара давать конкретные, четкие ответы по существу вопросов;

- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

## ***Методические рекомендации по выполнению различных форм самостоятельных домашних заданий***

Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны выполняться самостоятельно и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению.

Студентам следует:

- руководствоваться графиком самостоятельной работы, определенным РПД;
- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, и разбирать на семинарах и консультациях неясные вопросы;
- использовать при подготовке нормативные документы Финансового университета;
- при подготовке к экзамену параллельно прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные моменты для их обсуждения на плановой консультации.

### ***Методические рекомендации по работе с литературой***

Любая форма самостоятельной работы студента (подготовка к семинарскому занятию, выполнение расчетно-аналитической работы, начинается с изучения соответствующей литературы как в библиотеке, так и дома.

К каждой теме учебной дисциплины подобрана основная и дополнительная литература.

При работе с литературой рекомендуется делать записи. Записи в той или иной форме не только способствуют пониманию и усвоению изучаемого материала, но и помогают вырабатывать навыки явного изложения в письменной форме тех или иных теоретических вопросов.

### ***Методические указания по проведению практических занятий***

По структуре практические занятия следует разделить на учебные и контрольные.

*Учебные практические занятия* структурно состоят из следующих компонент:

- проверка наличия выполненного задания самостоятельной работы каждого студента;
- выборочная проверка корректности выполнения домашнего задания;
- разбор типичных ошибок, возникших в самостоятельной работе;
- рассмотрение теоретических вопросов, связанных с текущим практическим занятием;
- разбор методов выполнения практических заданий и решения задач;
- корректировка заданий для самостоятельной работы студентов.

*Контрольные практические занятия* структурно состоят из следующих компонент:

- проведение аудиторных самостоятельных работ;
- подведение итогов и разбор типичных ошибок, возникших при выполнении самостоятельных работ.

Студенты должны обратить внимание на перечень основных контрольных мероприятий, которые проводятся в соответствии с рабочей программой дисциплины.

Конкретные сроки проведения этих мероприятий своевременно доводятся до сведения студентов.

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

### **11.1. Комплект лицензионного программного обеспечения:**

Пакет офисных программ (Astra Linux, Library Office и пр.);

Антивирус Kaspersky;

### **11.2. Современные профессиональные базы данных и информационные справочные системы:**

Информационно-правовая система «Консультант Плюс»;

Информационно-правовая система «Гарант»;

Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>

Система комплексного раскрытия информации «СКРИН»: <https://skrin.ru>

11.3. Сертифицированные программные и аппаратные средства защиты информации - не предусмотрены.

11.4. Язык программирования Python 3.x в среде Windows, библиотеки PyOpenCl, mpi4py.

11.5. Платформа для научных исследований, основанная на языке программирования Python, Anaconda.

11.6. Интерфейс передачи сообщений MS MPI <https://www.microsoft.com/en-us/download/details.aspx?id=100593>

11.7. Компилятор g++ <https://www.msys2.org>

11.8. Открытый стандарт для распараллеливания программ на языках Си, Си++ и Фортран OpenMP, реализация для компилятора g++ [https://packages.msys2.org/package/mingw-w64-x86\\_64-openmp?repo=mingw64](https://packages.msys2.org/package/mingw-w64-x86_64-openmp?repo=mingw64)

## **12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Материально-техническая база Финансового университета, необходимая для осуществления образовательного процесса по данной дисциплине, в соответствии с требованиями ФОС ВО включает в себя специальные помещения для проведения лекций, семинарских занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации; помещения укомплектованы специализированной мебелью и техническими средствами обучения, необходимыми для представления информации большой аудитории.

Помещения для самостоятельной работы студентов включают в себя библиотеку с читальным залом, укомплектованную в соответствии с существующими нормами необходимой учебной и учебно-методической литературой и иными материалами; медиатеку с выходом в Интернет, компьютерные классы с возможностью работы в Интернет; аудитории для консультационной деятельности.