

**Негосударственное аккредитованное некоммерческое  
Частное образовательное учреждение высшего образования  
«АКАДЕМИЯ МАРКЕТИНГА И СОЦИАЛЬНО-  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ – ИМСИТ»  
(г. Краснодар)**

**Факультет информатики и вычислительной техники  
Кафедра математики и вычислительной техники**



УТВЕРЖДАЮ  
Председатель НМС,  
проректор по учебной работе,  
профессор

 Н.Н. Павелко

16 апреля 2018г.

**Б1.В.ДВ.07.01**  
**ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ**  
рабочая программа учебной дисциплины  
для студентов направления подготовки  
09.03.04 Программная инженерия  
Направленность (профиль) программы: «Информационно-  
вычислительные системы»  
Квалификация (степень выпускника) бакалавр

**г. Краснодар  
2018**

Рабочая программа составлена с учётом Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 Программная инженерия (уровень бакалавриата), утверждённого приказом Министерства образования и науки Российской Федерации от 12.03.2015 г. № 229

Составитель  В.В. Бужан

Согласовано:

Проректор по качеству, доцент



К.В. Писаренко

Рецензенты:

Левченко В.И., к.т.н., доцент, доцент кафедры автоматизации производственных процессов КубГТУ

Суриков А.И., директор ООО «1С-КОНСОЛЬ»

Рабочая программа рассмотрена на заседании кафедры математики и вычислительной техники от 19.03.2018 г., протокол №8

Зав. кафедрой математики и вычислительной техники, к.т.н., доцент  Н.С.Нестерова

Рабочая программа утверждена на заседании Научно-методического совета Академии от 16.04.2018 г., протокол №8.

## Содержание

1 Цели и задачи дисциплины .....	4
2 Место дисциплины в структуре ООП ВПО .....	6
3 Требования к результатам освоения содержания дисциплины .....	6
4 Объем учебной дисциплины и виды учебной работы .....	7
5 Содержание дисциплины .....	8
5.1 Содержание разделов (модулей) дисциплины .....	8
5.2 Разделы (модули) дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами .....	8
5.3 Разделы (модули) дисциплины и виды занятий .....	9
6 Лабораторные работы .....	10
7 Практические занятия .....	11
8 Самостоятельная работа .....	11
9 Примерная тематика курсовых работ (проектов) .....	12
10 Оценочные средства .....	12
11 Информационно-коммуникационные образовательные технологии .....	12
12 Учебно-методическое и информационное обеспечение дисциплины .....	13
13 Материально-техническое обеспечение дисциплины .....	14
14 Методические рекомендации по организации изучения дисциплины .....	14
15 Изучение дисциплины лицами с ограниченными возможностями здоровья .....	14
Регламент дисциплины .....	5

## 1 Цели и задачи дисциплины

*Цель* дисциплины «Технологии параллельного программирования» состоит в изучении математических моделей, методов параллельного программирования в объёме, достаточном для успешного начала работ в области параллельного программирования. Излагаемый набор знаний и умений составляет теоретическую основу для методов разработки сложных программ и включают такие темы, как цели и задачи параллельной обработки данных, принципы построения параллельных вычислительных систем, моделирование и анализ параллельных вычислений, принципы разработки параллельных алгоритмов и программ, технологии и системы разработки параллельных программ, параллельные численные алгоритмы для решения типовых задач вычислительной математики.

*Задачами дисциплины* является формирование у студентов теоретических знаний и практических навыков разработки алгоритмов и программ и их реализации на многоядерных и многопроцессорных вычислительных системах.

*Предметом изучения* являются алгоритмы параллельных вычислений и инструментарий разработчика программного обеспечения для их реализации на суперкомпьютерах.

### **Основные научные понятия, термины (дефиниции):**

*Алгоритм* – последовательность действий, которую необходимо выполнить для получения нужного результата.

*Данные-члены* – члены класса, которые содержат данные класса – поля, константы, события.

*Динамическая библиотека* – совокупность подпрограмм, обычно редко используемых, которые хранятся в одном DLL-файле на внешнем носителе и загружаемые в память компьютера во время выполнения программы по мере надобности и выгружаемые из памяти по завершении своей работы.

*Виртуальная машина* – интерпретатор специального промежуточного кода (байт-кода), реализующий завершающую ступень в мультиплатформенном программировании (предшествующая ступень – компиляция байт кода).

*Задача* – выполняемая процессором программа.

*Загрузочный модуль* – готовая к выполнению, прошедшая компиляцию и компоновку, программа.

*Класс* – абстрактный тип данных, в котором инкапсулируются (объединяются) данные и функции для их обработки.

*Компилятор* – транслятор с языка программирования высокого уровня. Обработывая исходную программу, компилятор создаёт эквивалентную программу на машинном языке, которая называется также объектной программой или объектным кодом.

*Компоновщик* – специальная программа, объединяющая в исполняемый модуль объектные коды.

*Константа* – неизменная величина.

*Конструкторы (constructor)* – специальные функции, вызываемые автоматически при инициализации объекта.

*Идентификатор* – имя программного объекта.

*Исполняемый модуль* – см. *загрузочный модуль*.

*Исходный модуль* – текст программы на языке программирования.

*Интерпретатор* – транслятор, который, распознавая исходную программу, не формирует машинный код в явном виде, а для каждой операции исполняет заранее заготовленную машинную команду или команды.

*Методы (method)* – функции, ассоциированные с некоторым классом.

*Модуль* – динамическая библиотека без атрибутов сборки.

*Модульное программирование* – технология, предполагающая выделение групп подпрограмм, использующих одни и те же глобальные данные в отдельно компилируемые модули.

*Объект* – тип (экземпляр) класса.

*Объектная программа* – программа на машинном языке, созданная компилятором из исходной программы на языке программирования высокого уровня.

*Объектный код* – см. *объектная программа*.

*Объектный модуль* – прошедшая компиляцию программа.

*Технологии параллельного программирования* – технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого типа (класса), а классы образуют иерархию с наследованием свойств.

*Переменная* – именованная область памяти, хранящая данные определённого типа.

*Перегрузка функций* – использование нескольких функций с одним и тем же именем, но с различными типами параметров.

*Перегрузка операций* – переопределение действий операций так, чтобы при использовании с объектами конкретного класса они выполняли заданные функции.

*Программа* – совокупность описаний (объявлений) и операторов, объединённая единым алгоритмом.

*Простой тип* – тип данных, не инкапсулированный в какой-либо класс.

*Процесс* – см. *Задача*.

*Рекурсия* – процесс вычислений, инициирующий сам себя (прямая рекурсия) или группа процессов, вызывающих друг друга (косвенная рекурсия). На практике рекурсия реализуется в виде функций.

*Сборка (assembly)* – логическая группировка одного или нескольких управляемых модулей и/или файлов ресурсов.

*Свойства (property)* – наборы функций, которые могут быть доступны таким же способом, как общедоступные поля класса.

*Структурное программирование* – совокупность рекомендуемых технологических приёмов, охватывающих выполнение всех этапов разработки программного обеспечения. В его основе лежит декомпозиция сложных систем с целью последующей реализации в виде отдельных небольших (до 40-50 операторов) подпрограмм.

*Ссылочный тип (referencetype)* – класс, все экземпляры которого хранят адреса ячеек в специально отведённой для этих целей области памяти, называемой «кучей».

*Скриншот* – копия фрагмента экранного изображения, сохранённая в одном из графических форматов файлов.

*Транслятор* – программа, получающая на входе исходный модуль программы и получающая на выходе функционально-эквивалентный исходному модулю объектный модуль.

*Тип данных* – концепция языков программирования высокого уровня, которая определяет внутреннее представление данных в памяти компьютера, множество значений, которые могут принимать величины этого типа и применимые к ним операции и функции.

*Тип значений (valuetype)* – класс, экземпляры которого хранят данные непосредственно в памяти.

*Указатель* – переменная, предназначенная для хранения адреса ячейки памяти.

*Файл* – логически связанная совокупность данных определённой длины, имеющая имя.

*Форма* – объект программы, в котором разработчик размещает элементы управления, служащие для ввода, отображения и изменения данных в полях.

*Функции-члены* – члены класса, которые обеспечивают некоторую функциональность для манипулирования данными класса.

*Член класса* – переменная, константа или функция, инкапсулируемая в классе.

*Экземпляр класса* – см. объект.

*MPI* – программный интерфейс ([API](#)) для передачи [информации](#), который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу. Разработан [Уильямом Гроуппом](#), [Эвином Ласком \(англ.\)](#) и другими.

## 2 Место дисциплины в структуре ООП ВПО

Дисциплина входит в вариативную часть образовательной программы бакалавра. Изучение данной дисциплины базируется на следующих курсах: «Программирование», «Информатика», «Технологии программирования».

## 3 Требования к результатам освоения содержания дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ОПК -3 готовностью применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

ПК -1 готовностью применять основные методы и инструменты разработки программного обеспечения

ПК – 3 владением навыками использования различных технологий разработки программного обеспечения

В результате изучения дисциплины студент должен:

### **ЗНАТЬ:**

□ Методы параллельных вычислений для задач вычислительной математики (матричные вычисления, решение систем линейных уравнений, сортировка, обработка графов, уравнения в частных производных, многоэкстремальная оптимизация).

□ Основные подходы к разработке параллельных программ.

**УМЕТЬ:**

- Строить модель выполнения параллельных программ.
- Оценивать эффективности параллельных вычислений.
- Анализировать сложность вычислений и возможность распараллеливания разрабатываемых алгоритмов.
- Применять общие схемы разработки параллельных программ для реализаций собственных алгоритмов.
- Оценивать основные параметры получаемых параллельных программ, таких как ускорение, эффективность и масштабируемость

**ВЛАДЕТЬ:**

- основами разработки параллельных программ для многоядерных/многопроцессорных вычислительных систем.

**4 Объем учебной дисциплины и виды учебной работы**

## ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего часов /зачетн. ед.	Семестр 8
<b>Аудиторные занятия, всего</b>	36/1	36/1
в том числе:		
лекции	12/0,33	12/0,33
практические занятия (ПЗ)	24/0,67	24/0,67
<b>Самостоятельная работа, всего</b>	36/1	36/1
в том числе:		
Самостоятельное решение задач.	18/0,5	18/0,5
Подготовка к контрольным работам	18/0,5	18/0,5
Вид промежуточной аттестации	зачет	зачет
<b>Общая трудоемкость по дисциплине</b>		
часы	72	72
зачетные единицы	2	2

**5 Содержание дисциплины****5.1 Содержание разделов (модулей) дисциплины**

№ п.п.	Наименование модуля	Содержание раздела	Форма текущего контроля
1	2	3	4
1	<b>Модуль 1</b> Алгоритмы параллельных вычислений	Многоядерные вычислительные системы. Моделирование и анализ параллельных алгоритмов. Этапы разработки параллельных алгоритмов. Типовые параллельные алгоритмы.	ЗПР (защита практической работы)
2	<b>Модуль 2</b> Технологии параллельного программирования	Средства разработки параллельных программ. Интерфейс передачи сообщений – MPI. Технология	ЗПР (защита практической работы)

	программирования OpenMP. Программирование МВС с графическими процессорами	
--	---	--

## 5.2 Разделы (модули) дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами

Указанная дисциплина обеспечиваемых дисциплин не имеет.

## 5.3 Разделы (модули) дисциплины и виды занятий

### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

8 семестр

№ раз-дела	Наименование разделов	Количество часов			
		Всего	Аудиторная работа		Внеауд. работа СР
			Л	ПЗ	
1	2	3	4	5	6
1	<b>Модуль 1</b> Тема 1.1. Современное состояние суперкомпьютерной вычислительной техники. Использование мировых сетевых ресурсов для параллельных вычислений	9	2	2	5
2	<b>Модуль 1</b> Тема 1.2. Концепция неограниченного параллелизма	9	2	2	5
3	<b>Модуль 1</b> Тема 1.3. Этапы разработки параллельных алгоритмов	9	2	2	5
4	<b>Модуль 1</b> Тема 1.4. Типовые параллельные алгоритмы: умножение матриц	9	2	2	5
5	<b>Модуль 1</b> Тема 1.5. Алгоритмы Фокса и Кеннона. Решение систем линейных уравнений. Алгоритмы параллельной сортировки	9	2	2	5
6	<b>Модуль 1</b> Тема 2.1. Распространённые языки программирования и коммуникационные библиотеки и интерфейсы	9	2	2	5
7	<b>Модуль 2</b> Тема 2.2. Средства автоматического распараллеливания, параллельные компиляторы	9	2	2	5
8	<b>Модуль 2</b> Тема 2.3. Параллельные предметные библиотеки	9	2	2	5
9	<b>Модуль 2</b> Тема 2.4. Инструментальные системы для проектирования параллельных программ	9	2	2	5
10	<b>Модуль 2</b> Тема 2.5. Общие принципы построения и реализации MPI	9	2	2	5



11	<b>Модуль 2</b> Тема 2.6. Технология программирования OpenMP	9	2	2	5
12	<b>Модуль 2</b> Тема 2.7. Программирование MVC с графическими процессорами	9	2	2	5
	<i>Итого:</i>	108	24	24	60

### 6 Лабораторные работы

Лабораторный практикум для данной дисциплины не предусмотрен

### 7 Практические занятия

#### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

8 семестр

№ занятия	Тема практического занятия	Количество часов
1	Параллельные алгоритмы	4
2	Численное интегрирование с использованием функций MPI	4
3	Параллельное программирование на платформе .NET	4
4	Параллельные алгоритмы на векторах	4
5	Параллельные алгоритмы на матрицах	4
6	Параллельное программирование с помощью асинхронных вызовов	4

### 8 Самостоятельная работа

#### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ работы	Вид работы	Вид контроля	ч. / з. е.
1	Работа с конспектом лекций. Работа с дополнительной литературой. Изучение тем занятий по электронным учебным пособиям	Контрольный опрос (устный, письменный). Индивидуальное собеседование	18/0,5
2	Решение задач	Контрольная аудиторная (домашняя) работа. Зачёт	42/1,17
			60/1,67

## 9 Примерная тематика курсовых работ (проектов)

Курсовые работы не предусмотрены учебным планом

## 10 Оценочные средства

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, осваивающих дисциплину «Технологии параллельного программирования».

Фонд оценочных средств включает контрольные материалы для проведения текущего контроля по проблемным вопросам и промежуточной аттестации в форме зачёта.

Фонд оценочных средств прилагается к рабочей программе дисциплины.

## 11 Информационно-коммуникационные образовательные технологии

### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

№ занятия	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
1	ПР	Круглый стол на тему: «Современное состояние суперкомпьютерной вычислительной техники»	2
2	ПР	Обсуждение методом мозгового штурма на тему «Параллельные алгоритмы на векторах»	2
3	ПР	Обсуждение методом мозгового штурма на тему «Параллельные алгоритмы на матрицах»	2
4	ПР	Круглый стол на тему: «Перспективы развития суперкомпьютерной техники и параллельных вычислений»	2
5	ПР	Обсуждение методом мозгового штурма на тему: «Численное интегрирование»	2
			10

## 12 Учебно-методическое и информационное обеспечение дисциплины

### а) основная литература

Дисциплина «Технологии параллельного программирования» обеспечивается необходимой учебной, учебно-методической и специализированной литературой.

1. Воеводин В., Воеводин Вл. Параллельные вычисления. Санкт-Петербург: БХВ-Петербург, 2015 г. , 608 с <http://www.twirpx.com/file/159421/>
2. Бужан В.В. Объектно-ориентированное программирование: учебно-методическое пособие для студентов очной и заочной форм обучения направления подготовки 230100.62 – Информатика и вычислительная техника. – Краснодар: ИМСИТ, 2013. – 52 с.
3. Бужан В.В. Программирование: учебно-методическое пособие для студентов очной и заочной форм обучения направления подготовки 230100.62 – Информатика и вычислительная техника. – Краснодар: ИМСИТ, 2013. – 84 с.

### б) дополнительная литература

1. Информатика и программирование. Компьютерный практикум: учеб. пособие для вузов / Гуда, А.Н., Бутаков, М.А., Нечитайло, Н.М., Чернов, А.В. – М.: Дашков и К, 2010. – 240 с.
2. Павловская Т.А. С#. Программирование на языке высокого уровня: Учебник для вузов. СПб.: Питер, 2012. — 432 с.
3. Хорев, П.Б. Технологии объектно-ориентированного программирования. – М.: Академия, 2008. – 448 с.
4. Истомин Е.П., Новиков В.В., Новикова М.В. Высокоуровневые методы информатики и программирования. СПб.: ООО «Андреевский издательский дом», 2008. — 228 с.
5. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2013. — 464 с.
6. Нейгел К., Ивьен Б., Глинн Дж., Уотсон К., Скиннер М. С# 4 и платформа .NET 4 для профессионалов. М.: И.Д. «Вильямс», 2011. – 1440 с
7. Ватсон К., Беллиназо М., Корне О., Гринфосс Д.Э.З., Найджел К., Рейд Д.Х.П.Д, Рейнольде М., Скиннер М., Уайт Э. С#. — М.: «Лори», 2008. – 852 с.
8. Подбельский В.В. Язык С++. Москва. «Финансы и статистика». 2010 — 560 с.
9. Бужан В.В. Основы программирования в среде Borland С++ Builder. Учебное пособие. — Краснодар.: ЮИМ, 2005. — 109 с.
10. Страуструп Б. Язык программирования Си++: Пер. с англ.— М.: Радио и связь, 2001. — 352 с.
11. Алексеев Е.Р. Программирование на VisualC++ и TurboC++ Explorer. М.:ИТ Пресс, 2007. – 352 с.
12. Пахомов Б.Н. С/С++ и MSVisualC++ 2005 для начинающих. – СПб.: БХВ-Петербург, 2007. – 464 с.
13. Сергеев А.П., Терен А.Н. Программирование в MicrosoftVisualC++ 2005. – М.:Издательский дом «Вильямс», 2006. – 352 с.
14. Хортон А. VisualC++ 2005: Базовый курс – М.: И.Д. «Вильямс», 2007. – 1152 с.
15. Культин Н. С# в задачах и примерах. – СПб.: БХВ-Петербург, 2007. – 240 с.

#### **в) программное обеспечение**

Преподавание и подготовка студентов предполагает использование стандартного программного обеспечения для персонального компьютера: MicrosoftWindows XP/Vista/7, Mac OS X Tiger/Leopard/SnowLeopard, MicrosoftVisualStudio 2010, Xcode; браузеров для поиска информации в базах данных по дисциплине в глобальной сети: MicrosoftInternetExplorer, Opera, Safari.

#### **г) базы данных, информационно-справочные и поисковые системы для освоения дисциплины**

Рекомендуется использование баз данных, информационно-справочных систем:  
<http://www.intuit.ru/> Интернет университет информационных технологий  
<http://www.openet.edu.ru/> Российский портал открытого образования « Российский образовательный портал».  
<http://www.en.edu.ru/> Естественно-научный образовательный портал  
<http://fcior.edu.ru/> Федеральный центр информационно-образовательных ресурсов  
<http://window.edu.ru/> Единое окно доступа к образовательным ресурсам  
<http://www.studfiles.ru/> Все для учебы  
<http://www.google.ru/> Google – поисковый сервер  
<http://www.imsit.ru/> электронные информационно-справочные ресурсы электронной библиотеки ИМСИТ

### **13 Материально-техническое обеспечение дисциплины**

Освоение дисциплины предполагает использование академической аудитории для проведения лекционных и семинарских занятий с необходимыми техническими средствами (компьютер, оборудование мульти-медиа, доска): IBM-совместимый персональный компьютер или AppleMacintosh на базе процессора Intel.

#### **14 Методические рекомендации по организации изучения дисциплины**

В соответствии с требованиями ФГОС ВПО по направлению подготовки реализация компетентностного подхода предусматривает использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных задач, проведение блиц-опросов, исследовательские работы) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Удельный вес занятий, проводимых в интерактивной форме, составляет не менее 15% аудиторных занятий (определяется ФГОС с учетом специфики ООП).

Лекционные занятия дополняются ПЗ и различными формами СРС с учебной и научной литературой. В процессе такой работы студенты приобретают навыки «глубокого чтения» - анализа и интерпретации текстов по методологии и методике дисциплины.

Учебный материал по дисциплине «Технологии параллельного программирования» разделен на логически завершённые части (модули), после изучения, которых предусматривается аттестация в форме письменных тестов, контрольных работ.

Работы оцениваются в баллах, сумма которых дает рейтинг каждого обучающегося. В баллах оцениваются не только знания и навыки обучающихся, но и их творческие возможности: активность, неординарность решений поставленных проблем. Каждый модуль учебной дисциплины включает обязательные виды работ – лекции, ПЗ, СРС (выполнение домашних заданий по решению задач).

Форма текущего контроля знаний – работа студента на практическом занятии. Форма промежуточных аттестаций – письменная (домашняя) работа и доклад на заданную тему. Итоговая форма контроля знаний по дисциплине – контрольная работа с задачами по всему материалу курса.

#### **15 Изучение дисциплины лицами с ограниченными возможностями здоровья**

Для студентов из числа лиц с ограниченными возможностями здоровья обучение проводится Академией с учетом особенностей их психофизического развития, их индивидуальных возможностей и состояния здоровья.

При проведении обучения по дисциплине обеспечивается соблюдение следующих общих требований:

- проведение обучения для лиц с ограниченными возможностями здоровья в одной аудитории совместно со студентами, не имеющими ограниченных возможностей здоровья, если это не создает трудностей для них в процессе обучения;
- присутствие в аудитории ассистента, оказывающего обучающимся необходимую техническую помощь с учетом их индивидуальных особенностей (занять рабочее место, передвигаться, прочитать и оформить задание, общаться с преподавателем);
- пользование необходимыми обучающимся техническими средствами при выполнении практических и других работ в соответствии с учебным планом с учетом их индивидуальных особенностей;

В зависимости от индивидуальных особенностей обучающихся с ограниченными возможностями здоровья образовательная среда Академии обеспечивает выполнение следующих требований при обучении и проведении промежуточной и итоговой аттестации:

а) для слепых:

- задания и иные материалы для аттестации зачитываются ассистентом;

- письменные задания надиктовываются обучающимся ассистенту;
- б) для слабовидящих:
  - задания и иные учебно-методические материалы оформляются увеличенным шрифтом;
  - обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
  - при необходимости обучающимся предоставляется увеличивающее устройство, допускается использование увеличивающих устройств, имеющихся у обучающихся;
- в) для глухих и слабослышащих, с тяжелыми нарушениями речи:
  - обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования;
  - по их желанию аттестационные испытания проводятся в письменной форме;
- г) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):
  - письменные задания надиктовываются ассистенту;
  - по их желанию все аттестационные испытания проводятся в устной форме.

### **Регламент дисциплины**

Дисциплина Технологии параллельного программирования

Преподаватель Бужан Виталий Викторович

Трудоёмкость дисциплины (из учебной программы дисциплины)

### **ОЧНАЯ ФОРМА ОБУЧЕНИЯ**

Курс 4, семестр 8

Общее количество ауд. часов	48
Лекции	24
Семинарские (практические) задания	24
Лабораторные работы	
Самостоятельная работа студентов	60
Форма рубежного контроля по дисциплине	зачёт

### **Оценка текущей работы студента по дисциплине в семестре**

#### **Посещение занятий**

Общее количество занятий	Балл одного занятия	Количество баллов, начисляемых за посещение занятий
1	2	4
24	0,25	0,25 n

### **Оценка текущей работы студента по дисциплине в семестре**

#### **ОЧНАЯ ФОРМА ОБУЧЕНИЯ**

Семестр 8

Виды работы	Количество баллов
Защита практической работы	36
Контрольная работа по модулю	18
Посещение занятий	6

Суммарный итог с учётом посещения занятий	60
---	----

### Рубежный контроль

Зачет/Экзамен	Количество баллов, набранных по соответствующей шкале (100 баллов)
Зачёт	30 баллов

### Премиальные баллы по дисциплине (до 10 баллов):

Студенту начисляют бонусные баллы за прилежание, нестандартные решения, умение быстро выполнять практические задания, применение системного подхода в процессе изучения дисциплины, грамотную речь, выполнение научно-исследовательской работы по дисциплине.

Итоговая балльная оценка студента Q рассчитывается по формуле

$$Q=N+M+R,$$

где N, M и R соответственно количество баллов, набранных за текущую работу по дисциплине в течение семестра; количество баллов по промежуточной аттестации, премиальные баллы.

Шкала итоговых оценок экзамена (зачёта) в зависимости от набранных баллов

<b>Промежуточный контроль</b>			<b>30</b>
<b>Зачёт:</b>			
- «Зачтено»		<b>50-100</b>	
- «Незачтено»		<b>0-49</b>	