

**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.07**  
**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ**  
**ПРОГРАММИРОВАНИЕ**  
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 .....	8
6	Лабораторные работы .....	9
7	Практические занятия .....	10
8	Самостоятельная работа .....	10
9	Примерная тематика курсовых работ (проектов) .....	11
10	Оценочные средства .....	11
11	Информационно-коммуникационные образовательные технологии .....	11
12	Учебно-методическое и информационное обеспечение дисциплины .....	12
13	Материально-техническое обеспечение дисциплины .....	13
14	Методические рекомендации по организации изучения дисциплины .....	13
15	Изучение дисциплины лицами с ограниченными возможностями здоровья .....	13
	Регламент дисциплины .....	154

## 1 Цели и задачи дисциплины

*Целью преподавания* дисциплины «Объектно-ориентированное программирование» (ООП) является обучение студентов методике разработки программных средств с использованием технологии объектно-ориентированного программирования.

*Задачами дисциплины* являются приобретение знаний в области ООП, построения приложений с привлечением ООП, изучения современных объектно-ориентированных языков программирования.

*Предметом изучения* являются объектно-ориентированные языки программирования.

### **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.

*Объектно-ориентированное программирование* – 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 (reference type)* – 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 (value type)* – 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.

## 2 Место дисциплины в структуре ООП ВПО

Дисциплина входит в вариативную часть образовательной программы бакалавра.

Изучение данной дисциплины базируется на следующих курсах:

«Программирование», «Информатика». Дисциплина является предшествующей для дисциплины «Технологии программирования».

## 3 Требования к результатам освоения содержания дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

а) общепрофессиональных

1. готовностью применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-3);

б) профессиональных

2. готовностью применять основные методы и инструменты разработки программного обеспечения (ПК-1);
3. владением навыками использования различных технологий разработки программного обеспечения (ПК-3).

В результате изучения дисциплины студент должен:

**ЗНАТЬ:**

- классы;
- экземпляры классов;
- конструкторы и деструкторы;

- жизненный цикл программы;
- наследование;
- полиморфизм;
- структуру программы;
- время жизни и область видимости программных объектов;
- способы конструирования и верификации программ;
- принципы объектно-ориентированного анализа.

**УМЕТЬ:**

- работать с компьютерной литературой;
- составлять алгоритмы решения задач;
- реализовать алгоритмы на языке объектно-ориентированном языке высокого уровня;
- проводить отладку и испытание программ.

**ВЛАДЕТЬ:**

- навыками объектно-ориентированного программирования;
- программными средствами разработки программ.

#### 4 Объем учебной дисциплины и виды учебной работы

##### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Вид учебной работы	Всего часов /зачетн. ед.	Семестр 4
<b>Аудиторные занятия, всего</b>	48/1,33	48/1,33
в том числе:		
лекции	16/0,44	16/0,44
практические занятия (ПЗ)	32/0,89	32/0,89
<b>Самостоятельная работа, всего</b>	60/1,67	60/1,67
в том числе:		
контрольные (домашние) работы	18/0,5	18/0,5
Изучение теоретического материала, подготовка к аудиторным занятиям	18/0,5	18/0,5
Самостоятельное решение задач. Подготовка к контрольным работам	24/0,67	24/0,67
Вид промежуточной аттестации		зачёт
<b>Общая трудоёмкость по дисциплине</b>	108	108
часы		
зачётные единицы	3	3

#### 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> Классы и объекты	Классы. Объекты. Конструкторы и деструкторы. Методы. Свойства. Интерфейсы	ЗПР
2	<b>Модуль 2</b>	Перезагрузка методов. Перегрузка	ЗПР

	Полиморфизм	операций. Виртуальные функции.	
3	<b>Модуль 3</b> Наследование	Наследование интерфейсов. Наследование реализации	ЗПР

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

Наименование обеспечиваемых (последующих) дисциплин	№/№ разделов данной дисциплины, необходимых для изучения обеспечиваемых (последующих) дисциплин
Технологии программирования (Б1.В.ОД.11)	Модуль 1, 2, 3

## 5.3 Разделы (модули) дисциплины и виды занятий

### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

4 семестр

№ раздела	Наименование разделов	Количество часов			
		Всего	Аудиторная работа	Внеауд. работа	
			Л	ПЗ	СР
1	2	3	4	5	6
1	<b>Модуль 1</b> Тема 1.1. Введение в дисциплину	5	1	0	4
2	<b>Модуль 1</b> Тема 1.2. Инкапсуляция	9	1	2	6
3	<b>Модуль 1</b> Тема 1.3. Функции-члены класса	14	2	4	8
4	<b>Модуль 2</b> Тема 2.1. Перегрузка функций	12	2	4	6
5	<b>Модуль 2</b> Тема 2.2. Перегрузка операций	14	2	4	8
6	<b>Модуль 3</b> Тема 3.1. Наследование реализации	14	2	4	8
7	<b>Модуль 3</b> Тема 3.2. Наследование интерфейсов	12	2	4	6
8	<b>Модуль 3</b> Тема 3.3. Отношения между классами	14	2	6	6
9	<b>Модуль 3</b> Тема 3.4. Диаграмма классов	14	2	4	8
	<i>Итого:</i>	108	16	32	60

## 6 Лабораторные работы

Лабораторный практикум для данной дисциплины не предусмотрен



## 7 Практические занятия

### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

4 семестр

№ занятия	Тема практического занятия	Количество часов
1	Разработка классов	4
2	Создание простого наследования	6
3	Создание виртуальных функций	6
4	Перегрузка методов и перегрузка классов	4
5	Отношения между классами	6
6	Диаграммы классов	6

## 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 «Объектно-ориентированное программирование».

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: «Инкапсуляция»	2
2	ПР	Обсуждение методом мозгового штурма на тему «Перегрузка функций»	2
3	ПР	Обсуждение методом мозгового штурма на тему «Перегрузка операций»	2
4	ПР	Круглый стол на тему: «Наследование	2

		интерфейсов»	
5	ПР	Обсуждение методом мозгового штурма на тему: «Наследование реализации»	2
6	ПР	Круглый стол на тему: «Отношения между классами»	2
7	ПР	Обсуждение методом мозгового штурма на тему: «Диаграмма классов»	2
8	ПР	Круглый стол на тему: «Наследование интерфейсов»	2
			16

4

## 12 Учебно-методическое и информационное обеспечение дисциплины

### а) основная литература

Дисциплина «Объектно-ориентированное программирование» обеспечивается необходимой учебной, учебно-методической и специализированной литературой.

1. Бужан В.В. Объектно-ориентированное программирование: учебно-методическое пособие для студентов очной и заочной форм обучения направления подготовки 09.03.01– Информатика и вычислительная техника. – Краснодар: ИМСИТ, 2013. – 52 с.
2. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2013. — 464 с. <http://www.studfiles.ru/preview/397655/>
3. Немцова Т.И. Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учебное пособие / Т.И. Немцова; Под ред. Л.Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2015. - 496 с. <http://znanium.com/catalog.php?bookinfo=472870>

### б) дополнительная литература

1. Васильев, А.Н. Java: Объектно-ориентированное программирование: базовый курс для магистров и бакалавров. – СПб: ПИТЕР, 2011. – 400 с.
2. 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
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., Щупак Ю.А. С++. Структурное и объектно-ориентированное программирование. Практикум. СПб.: Питер, 2010 — 352 с.
6. Нейгел К., Ивсен Б., Глиин 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 Visual C++ и Turbo C++ Explorer. М.:ИТ Пресс, 2007. – 352 с.
12. Пахомов Б.Н. C/C++ и MS Visual C++ 2005 для начинающих. – СПб.: БХВ-Петербург, 2007. – 464 с.
13. Сергеев А.П., Терен А.Н. Программирование в Microsoft Visual C++ 2005. – М.:Издательский дом «Вильямс», 2006. – 352 с.
14. Хортон А. Visual C++ 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: Microsoft Windows XP/Vista/7, Mac OS X Tiger/Leopard/Snow Leopard, Microsoft Visual Studio 2010, Xcode; браузеров для поиска информации в базах данных по дисциплине в глобальной сети: Microsoft Internet Explorer, 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 мультимедиа, доска): IBM-совместимый персональный компьютер или Apple Macintosh 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 «Объектно-ориентированное программирование» 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 Объектно-ориентированное программирование  
Преподаватель Бужан Виталий Викторович

Трудоёмкость дисциплины (из учебной программы дисциплины)

### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Курс2, семестр 4

Общее количество ауд. часов	48
Лекции	16
Семинарские (практические) задания	32
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 $n$ занятий
1	2	4
24	0,25	0,25 $n$

### Оценка текущей работы студента по дисциплине в семестре

#### ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Семестр 4

Виды работы	Количество баллов
Контрольная работа по модулю	18
Защита практической работы	36
Посещение занятий	6
Суммарный итог с учётом посещения занятий	60

*Примечание:* Текущий балл студента по работе в течение семестра = количество набранных баллов по посещаемости + количество набранных баллов по выполнению учебно-методической работы по дисциплине.

#### 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><i>Промежуточный контроль</i></b>			<b>30</b>
<b><i>Экзамен:</i></b>			
- « <b><i>Отлично</i></b> »		<b>90-100</b>	
- « <b><i>Хорошо</i></b> »		<b>70-89</b>	
- « <b><i>Удовлетворительно</i></b> »		<b>50-69</b>	
-« <b><i>Неудовлетворительно</i></b> »		<b>0-49</b>	