

Государственное образовательное учреждение
«Приднестровский государственный университет им. Т. Г. Шевченко»

Физико-математический факультет

Кафедра прикладной математики и информатики

«УТВЕРЖДАЮ»

Декан физ.-мат. факультета,
к. ф.-м. н., доцент



Коровой О. В.

«09» 2017 г.

РАБОЧАЯ ПРОГРАММА

на 2017/2018 учебный год

(2017 год набора)

учебной дисциплины

«Системы программирования»

Направление подготовки

01.03.02 «Прикладная математика и информатика»

Профиль подготовки

«Системное программирование и компьютерные технологии»

Квалификация (степень) выпускника

Бакалавр

Форма обучения

Очная

Тирасполь, 2017

Рабочая программа дисциплины «Системы программирования» / сост. В. И. Великодный — Тирасполь, ГОУ ПГУ им. Т. Г. Шевченко, 2017. — 14 с.

Рабочая программа предназначена для преподавания обязательной дисциплины вариативной части Блока 1 студентам очной формы обучения по направлению подготовки 01.03.02 «Прикладная математика и информатика».

Рабочая программа составлена с учетом Федерального Государственного образовательного стандарта высшего образования по направлению подготовки 01.03.02 «Прикладная математика и информатика», утвержденного приказом Министерства образования и науки РФ № 228 от 12.03.2015 г.

Составитель
ст. преподаватель



В. И. Великодный

1. Цели и задачи освоения дисциплины

Целями освоения дисциплины «Системы программирования» является приобретение компетенций в области написания компьютерных программ, применения парадигм программирования, разработки алгоритмов и структур данных.

Задачи изучения дисциплины:

- изучение различных подходов к программированию вычислительной техники;
- обучение основным приёмам написания компьютерных программ;
- формирование навыков практического использования инструментальных средств программирования для решения прикладных задач;
- формирование общекультурных и профессиональных компетенций, необходимых для осуществления профессиональной деятельности.

2. Место дисциплины в структуре ООП ВО бакалавриата

Дисциплина «Системы программирования» относится к обязательным дисциплинам вариативной части Блока 1 (Б1.В.ОД.5).

Для освоения дисциплины обучающиеся используют знания, умения и навыки, сформированные в ходе изучения предметов «Основы информатики» и «Архитектура компьютеров».

Дисциплина «Системы программирования» является предшествующей для изучения дисциплин «Вычислительные системы и параллельная обработка данных», «Технологии параллельного программирования», «Объектно-ориентированный анализ и проектирование», «Базы данных», «Конструирование компиляторов».

3. Требования к результатам освоения дисциплины

Изучение дисциплины направлено на формирование следующих компетенций:

Код компетенции	Формулировка компетенции
ОК-7	способность к самоорганизации и самообразованию
ОПК-1	способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой
ОПК-2	способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии
ОПК-3	способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям
ОПК-4	способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ПК-1	способность собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям
ПК-4	способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности
ПК-7	способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения

В результате освоения дисциплины студент должен:

3.1. Знать:

- основные парадигмы программирования;
- принципы функционирования компиляторов и интерпретаторов;
- основные конструкции языков программирования высокого уровня;
- базовые принципы структурного, объектно-ориентированного и событийно-ориентированного программирования;
- основные понятия параллельного программирования;
- наиболее распространённые структуры данных и алгоритмы работы с ними;
- методы и средства проектирования графических пользовательских интерфейсов.

3.2. Уметь:

- разрабатывать структуры данных, соответствующие поставленной задаче;
- использовать существующие или разрабатывать новые алгоритмы обработки данных;
- использовать инструментальное программное обеспечение для организации и повышения производительности труда;
- применять полученные навыки в других дисциплинах.

3.3. Владеть:

- приёмами решения типовых алгоритмических задач;
- методами и технологиями разработки программ на языках высокого уровня для решения прикладных задач;
- основными методами параллельного программирования для повышения производительности программ.

4. Структура и содержание дисциплины

4.1. Распределение трудоёмкости в ЗЕТ/часах по видам работы студентов по семестрам

Семестр	Количество часов						Форма итогового контроля
	Трудоёмкость, ЗЕТ/часы	В том числе					
		Аудиторных				СРС	
		Всего	Лекций	Лаб. работ	Практ. занятий		
1	5/180	99	36	63	—	81	зачёт, курс. работа
2	3/108	54	18	36	—	18	экзамен/36
Итого	8/288	153	54	99	—	99	36

4.2. Распределение видов учебной работы и их трудоёмкости по разделам дисциплины

№ раздела	Наименование раздела	Количество часов				
		Всего	Аудиторная работа			СРС
			Л	ЛР	ПЗ	
1-й семестр						
1.	Основные понятия	20	4	—	—	16
2.	Программирование	85	14	38	—	33
3.	Алгоритмы и структуры данных	46	10	14	—	22
4.	Пользовательские интерфейсы	29	8	11	—	10
Итого за семестр		180	36	63	—	81
2-й семестр						
5.	Разработка приложений	42	12	18	—	12
6.	Методы и технологии программирования	30	6	18	—	6

№ раздела	Наименование раздела	Количество часов				
		Всего	Аудиторная работа			СРС
			Л	ЛР	ПЗ	
Итого за семестр		72	18	36	—	18
Итого		252	54	99	—	99

4.3. Тематический план по видам учебной деятельности

Лекции

№ п/п	Раздел	Объём, часов	Тема лекции	Учебно-наглядные пособия
1-й семестр				
1.	1	2	Основные понятия	слайды
2.		2	Трансляция и выполнение программ	слайды
3.	2	2	Хранение данных	слайды
4.		4	Управление выполнением	слайды
5.		8	Объектно-ориентированное программирование	слайды
6.	3	4	Массивы и коллекции	слайды
7.		4	Обработка текстовых данных	слайды
8.		2	Управление памятью	слайды
9.	4	4	Пользовательские интерфейсы	слайды
10.		4	Графика и визуализация	слайды
Итого за семестр		36		
2-й семестр				
11.	5	4	Файлы и сериализация	слайды
12.		4	Параллельное программирование	слайды
13.	6	2	Асинхронное программирование	слайды
14.		2	Компонентное программирование	слайды
15.		4	Алгоритмы и структуры данных	слайды
16.		2	Технология программирования	слайды
Итого за семестр		18		
Итого		54		

Лабораторные работы

№ п/п	Раздел	Объём, часов	Тема лабораторного занятия	Лаборатория	Учебно- наглядные пособия
1-й семестр					
1.	2	4	Линейные алгоритмы	КУВТ	задачник

№ п/п	Раздел	Объём, часов	Тема лабораторного занятия	Лаборатория	Учебно-наглядные пособия
2.		6	Алгоритмы с ветвлением	КУВТ	задачник
3.		6	Циклические конструкции	КУВТ	задачник
4.		6	Структурное программирование	КУВТ	задачник
5.		8	Классы и исключения	КУВТ	задачник
6.		8	Наследование и полиморфизм	КУВТ	задачник
7.	3	8	Массивы и коллекции	КУВТ	задачник
8.		6	Обработка текстовых данных	КУВТ	задачник
9.	4	6	Пользовательские интерфейсы	КУВТ	задачник
10.		5	Графика и визуализация	КУВТ	задачник
Итого за семестр		63			
2-й семестр					
11.	5	6	Файлы и сериализация	КУВТ	задачник
12.	6	6	Параллельное программирование	КУВТ	задачник
13.		6	Компонентное программирование	КУВТ	задачник
14.		6	Рекурсивные структуры данных	КУВТ	задачник
15.		6	Обобщённое программирование	КУВТ	задачник
16.		6	Тестирование и диагностика	КУВТ	задачник
Итого за семестр		36			
Итого		99			

Практические (семинарские) занятия

Практические занятия по настоящей дисциплине не предусмотрены.

Самостоятельная работа студента

№ п/п	Раздел	Объём, часов	Тема и вид СРС
1-й семестр			
1.	1	4	Классификация языков программирования (Л)
2.		2	Алгоритмы и программы (К)
3.		4	Библиотеки (Л)
4.		4	Распространённые языки программирования (Л)
5.		2	Трансляция и выполнение программ (К)
6.	2	4	Виды памяти (Л)
7.		4	Адресное пространство процесса (Л)
8.		2	Хранение данных (К)
9.		4	Операции языка С# (Л)
10.		4	Применение рекурсии (Л)

№ п/п	Раздел	Объём, часов	Тема и вид СРС
11.		2	Управление выполнением (К)
12.		4	Основные принципы ООП (Л)
13.		2	Объектно-ориентированное программирование (К)
14.		7	Структурное и объектно-ориентированное программирование (Л)
15.	3	6	Коллекции .NET (Л)
16.		4	Распространённые структуры данных (Д)
17.		2	Специальные типы данных (К)
18.		4	Методы сортировки (Л)
19.		4	Регулярные выражения (Л)
20.		2	Обработка текстовых данных (К)
21.	4	6	Цифровая обработка изображений (Л)
22.		2	Графика и визуализация (К)
23.		2	Пользовательские интерфейсы (К)
Итого за семестр		81	
2-й семестр			
24.	5	2	Потоки ввода-вывода (Л)
25.		2	Файлы и сериализация (К)
26.		2	Параллельное программирование (К)
27.		2	Технологии параллельного программирования (Д)
28.		2	События и асинхронное программирование (К)
29.		2	Компонентное программирование (К)
30.	6	2	Алгоритмы и структуры данных (К)
31.		2	Функциональное программирование (К)
32.		2	Технология программирования (К)
Итого за семестр		18	
Итого		72	

Примечание:

К — подготовка к контрольной работе, тестированию;

Л — работа с литературой, подготовка к семинару;

Д — написание реферата, доклад.

5. Примерная тематика курсовых проектов

- 1) Язык программирования PHP
- 2) Математические библиотеки для языка Python
- 3) Разработка мобильных приложений для Android
- 4) Язык программирования JavaScript
- 5) Разработка компьютерных игр для командной строки
- 6) Язык программирования Swift
- 7) Программирование трёхмерной графики с использованием OpenGL
- 8) Язык программирования Rust

- 9) Язык программирования Scheme
- 10) Разработка программ для микроконтроллеров архитектуры AVR
- 11) Программирование игры Жизнь
- 12) Язык программирования Java
- 13) Язык программирования F#
- 14) Веб-дизайн и разработка Front-end
- 15) Программирование цифровой обработки изображений
- 16) Язык программирования Python
- 17) Фреймворк Unity3D
- 18) Язык программирования Ruby.

6. Образовательные технологии

В учебном процессе используются следующие образовательные технологии. По образовательным формам: лекции, лабораторные занятия, индивидуальные занятия, контрольные работы. По преобладающим методам и приёмам обучения: объяснительно-иллюстративные (объяснение, показ-демонстрация учебного материала и др.); активные (анализ учебной и научной литературы, составление схем и др.) и интерактивные, в том числе и групповые (взаимное обучение в форме подготовки и обсуждения докладов); информационные; компьютерные; мультимедийные (работа с сайтами академических структур, научно-исследовательских организаций, электронных библиотек и др., разработка презентаций, сообщений и докладов, работа с электронными обучающими программами и т. п.).

7. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Результаты освоения учебной дисциплины оцениваются следующими средствами текущего контроля успеваемости:

- решение тестовых заданий;
- выполнение контрольных работ;
- выполнение и защита лабораторных работ.

Методическое обеспечение текущего контроля:

1. Комплект тестовых заданий по вариантам.
2. Комплект контрольных заданий по вариантам.
3. Комплект индивидуальных заданий к лабораторным работам.

Отчет о выполненной лабораторной работе оформляется каждым студентом индивидуально.

Защита лабораторных работ производится по каждой работе в отдельности в виде индивидуального собеседования с каждым студентом по теоретической и практической частям выполненной работы.

Промежуточная аттестация по итогам освоения дисциплины проводится в форме зачёта (1 семестр) и экзамена (2 семестр).

7.1. Темы рефератов

Тема «Распространённые структуры данных»:

1. Связные списки.
2. Хеш-таблицы.
3. Деревья сортировки.
4. Красно-чёрные деревья.
5. B-деревья.
6. Суффиксное дерево.
7. Дерево Фенвика.
8. Биномиальные пирамиды.
9. Фибоначчиевы пирамиды.
10. Фильтр Блума.

Тема «Технологии параллельного программирования»:

1. Библиотека OpenMP.
2. Библиотека MPI.
3. Язык параллельного программирования Cilk.

4. Библиотека CUDA.
5. Библиотека OpenCL.
6. Модель акторов.
7. Программная транзакционная память.
8. Модель параллельных вычислений MapReduce.
9. Грид-вычисления.
10. Проекты «добровольных вычислений».

7.2. Примеры тестовых заданий текущего контроля

Тестирование проводится после каждой изученной темы. Время тестирования — 10 минут.

Тема 2. «Трансляция и выполнение программ»

1. Для какого из подходов характерна наименьшая программно-аппаратная переносимости?
 - компиляция;
 - интерпретация;
 - компиляция в промежуточный код.
2. На каком из этапов трансляции конструкции языка C++ преобразуются в машинные команды?
 - этап препроцессора;
 - связывание;
 - компиляция.
3. Дайте определение интерпретации.
4. Какая из перечисленных стадий компиляции выполняется первой?
 - семантический анализ;
 - лексический анализ;
 - оптимизация;
 - генерация кода;
 - синтаксический анализ.
5. При каких способах организации библиотеки требуемый участок кода подставляется непосредственно в исполняемый файл?
 - описание функций в заголовочных файлах;
 - статически связываемые библиотеки;
 - динамически связываемые библиотеки.

Тема 4. «Управление выполнением»

1. Какие из перечисленных команд можно использовать для выхода из цикла?
 - continue
 - goto
 - if
 - break
 - foreach
2. Как сгенерировать исключение Exception?
 - throw Exception;
 - throw Exception();
 - throw new Exception;
 - throw new Exception();
3. Что появится на экране в результате работы приведённой программы?
`Func<int, int, int> f = (x, y) => {return x + 2 * y;};`
`Console.WriteLine(f(5, 7));`
4. Что такое исключительная ситуация?
5. Что появится на экране в результате работы приведённой программы? (A и B — некоторые классы)


```
try {
    throw new A();
    throw new B();
} catch (A) {
    Console.WriteLine("A");
} catch (B) {
```

```
        Console.WriteLine("B");  
    }
```

7.3. Вопросы к зачёту

1. Языки программирования и алгоритмы.
2. Трансляция и выполнение программ.
3. Конструкции управления выполнением.
4. Объектно-ориентированное программирование.
5. Массивы и коллекции.
6. Специальные типы данных.
7. Обработка текстовых данных.
8. Управление памятью.
9. Пользовательские интерфейсы.
10. Графика и визуализация.

7.4 Экзаменационные вопросы

1. Понятие языка программирования.
2. Классификация языков программирования.
3. Алгоритмы и их сложность.
4. Трансляция и интерпретация.
5. Этапы трансляции программы.
6. Исполняемые файлы и библиотеки.
7. Типы данных и литералы.
8. Переменные и константы.
9. Организация памяти.
10. Операции и операторы.
11. Подпрограммы и функции.
12. Исключительные ситуации.
13. Абстракция данных и инкапсуляция.
14. Наследование.
15. Полиморфизм.
16. Массивы.
17. Коллекции.
18. Технология LINQ.
19. Строки.
20. Регулярные выражения.
21. Алгоритмы на строках.
22. Контроль времени жизни.
23. Указатели и адресная арифметика.
24. Выделение памяти.
25. Виды интерфейсов.
26. Графические пользовательские интерфейсы.
27. Веб-интерфейсы.
28. Растровая графика.
29. Векторная графика.
30. Визуализация числовых данных.
31. Потоки ввода/вывода.
32. Файлы.
33. Сериализация.
34. Потоки и процессы.
35. Синхронизация потоков. Блокировки.
36. Параллельное выполнение задач.
37. Событийно-ориентированное программирование.
38. Сопрограммы.
39. Асинхронное программирование.
40. Сборки .NET.
41. Разработка библиотек.
42. Ресурсы приложения.

43. Рекурсивные типы данных.
44. Списки и очереди.
45. Деревья.
46. Этапы разработки программного обеспечения.
47. Системы контроля версий.
48. Тестирование.

7.5. Примеры практических заданий экзамена

1. Запросить с клавиатуры четырёхзначное натуральное число. Вывести его сумму с числом, образованным записью цифр в обратном порядке.
2. Даны три числа. Определить, могут ли они быть длинами сторон треугольника. Если да, определить его вид (равносторонний, равнобедренный, разносторонний).
3. Запросить с клавиатуры число. Проверить, является ли оно простым, иными словами, есть ли у него делители, кроме 1 и самого числа.
4. Определить функцию, возвращающую количество цифр в заданном натуральном числе. С использованием этой функции написать программу, определяющую, в каком из двух введенных с клавиатуры чисел больше цифр.
5. Написать класс Progression, описывающий арифметическую прогрессию. Определить в нём два закрытых поля a_0 (первый член) и d (разность). Определить следующие открытые методы:
 - ToString() — перегруженный метод из класса object, возвращающий строку с описанием прогрессии с использованием формата « $(a_0 + d \cdot k)$ », где вместо a_0 и d подставлены конкретные значения полей.
 - Конструктор с двумя параметрами для инициализации закрытых полей.
 - Term(int k) — метод, возвращающий k -й член прогрессии.
 Написать краткий пример использования класса.
6. Написать программу, находящую свёртку квадратов положительных элементов двух заданных коллекций.
7. Написать программу для замены в указанном файле одного слова на другое. Спроектировать и реализовать графический интерфейс.
8. Написать программу вычисления приближённого значения интеграла методом прямоугольников. Для повышения быстродействия использовать несколько потоков выполнения.

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Основная литература

1. Албахари Д., Албахари Б. С# 5.0. Справочник. Полное описание языка. — М.: Вильямс, 2014.
2. Шилдт Г. С# 4.0: Полное руководство. — М.: Вильямс, 2011.
3. Троелсен Э. Язык программирования С# 5.0 и платформа .NET 4.5. — 6-е изд. — М.: Вильямс, 2013.
4. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#. 4-е изд. — СПб.: Питер, 2012.
5. Скит Дж. C#. Программирование для профессионалов. — М.: Вильямс, 2011.

8.2. Дополнительная литература

1. Макконнелл С. Совершенный код. — СПб.: Питер, 2007.
2. Фридл Дж. Регулярные выражения. — СПб.: Символ-Плюс, 2000.
3. Буч Г., Максимчук Р. А., Энгл М. У., Янг Б. Дж., Коналлен Дж., Хьюстон К. А. Объектно-ориентированный анализ и проектирование с примерами приложений (UML 2). — 3е издание. — М.: Вильямс, 2010.
4. Абельсон Х., Сассман Дж. Дж. Структура и интерпретация компьютерных программ. — М.: Добросвет, 2006.
5. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К. Алгоритмы: построение и анализ, 2-е изд. — М.: Вильямс, 2009..
6. Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989.

8.3. Программное обеспечение и Интернет-ресурсы

Требуемое программное обеспечение:

- Microsoft .NET Framework 4.6 либо Mono 4.2.
- Интегрированная среда разработки на языке C#: Microsoft Visual Studio 2015 либо MonoDevelop 5.7.
- Компилятор F#.
- Система контроля версий Git или Mercurial.

Интернет-ресурсы:

- MSDN / <http://msdn.microsoft.com/ru-ru/>
- Pro Git / <http://git-scm.com/book/>
- Лазин Е., Моисеев М., Сорокин Д. Введение в F# // Практика функционального программирования, 2010, № 5. / <http://fprog.ru/2010/issue5/>
- Полное руководство по языку программирования C# 6.0 и платформе .NET 4.6 / <http://metanit.com/sharp/tutorial/>
- C# 5.0 и платформа .NET 4.5 / http://professorweb.ru/my/csharp/charp_theory/level1/infocsharp.php

9. Материально-техническое обеспечение дисциплины

Лабораторные работы должны выполняться в специализированных классах, оснащенных современными персональными компьютерами и программным обеспечением в соответствии с тематикой изучаемого материала; число рабочих мест в классах должно быть таким, чтобы обеспечивалась индивидуальная работа студента на отдельном персональном компьютере.

Аудитория	Технические характеристики	На текущий момент
Аудитория 227, корпус № 2	• локальная сеть; • доступ к сети Интернет; • ОС Windows либо GNU/Linux.	• 1 сервер; • 12 раб. станций; • проектор; • проекционный экран.

10. Методические рекомендации по организации изучения дисциплины:

Изучение дисциплины осуществляется посредством освоения теоретического блока, выполнения заданий лабораторного практикума и выполнения самостоятельной работы.

Данный курс предполагает овладение студентами основными понятиями и методами написания программ на языках высокого уровня, получение практических навыков разработки, отладки и тестирования программного обеспечения. Полученные в рамках курса знания являются основой для изучения дисциплин профессионального цикла, связанных с информатикой и вычислительной техникой.

Рекомендуется для лучшего усвоения понятий вести терминологический словарь; посещать все занятия и вести подробный конспект; работать с основной и дополнительной литературой; пользоваться Интернет-ресурсами; самостоятельно выбирать и решать дополнительные задачи по изучаемым темам; подготовить реферат и к нему презентацию с последующей защитой перед группой.

Рабочая учебная программа по дисциплине «Системы программирования» составлена с учетом Федерального Государственного образовательного стандарта высшего образования по направлению подготовки 01.03.02 «Прикладная математика и информатика», утвержденного приказом Министерства образования и науки РФ № 228 от 12.03.2015 г. и учебного плана по профилю подготовки «Системное программирование и компьютерные технологии».

11. Технологическая карта дисциплины

Курс: 1, группа: 103, семестры: 1, 2.

Преподаватель-лектор: *ст. преп. В. И. Великодный.*

Преподаватель, ведущий лабораторные занятия: *ст. преп. Е. В. Калинова.*

Кафедра: *прикладной математики и информатики.*

Семестр 1

Семестр	Количество часов						Форма итогового контроля
	Трудо- емкость, з.е./часы	В том числе					
		Аудиторных				Самост. работы	
		Всего	Лекций	Лаб. раб.	Практ. зан.		
1	5/180	99	36	63	—	81	зачёт, курс. работа

Форма текущей аттестации	Расшифровка	Мин. число баллов	Макс. число баллов
Посещение лекционных занятий		0	10
Выполнение и защита лабораторных работ	3 балла за каждую лабораторную работу	0	30
Тесты по темам 1, 2 лекционных занятий	3 балла за каждый тест	0	6
Тесты по темам 3, 4 лекционных занятий	3 балла за каждый тест	0	6
Тесты по темам 5, 6 лекционных занятий	3 балла за каждый тест	0	6
Тесты по темам 7, 8 лекционных занятий	3 балла за каждый тест	0	6
Тесты по темам 9, 10 лекционных занятий	3 балла за каждый тест	0	6
Итого по текущей аттестации		45	70
Промежуточная аттестация	зачёт	10	30
Итого по семестру		55	100

Курсовая работа

Критерии оценки	Мин. число баллов	Макс. число баллов
Полнота изложения теоретической части	0	25
Корректность решения практических задач	0	30
Аккуратность оформления, стиль изложения	0	15
Итого по текущей аттестации	45	70
Промежуточная аттестация (защита курсовой работы)	10	30
Итого	55	100

Семестр 2

Семестр	Количество часов						Форма итогового контроля
	Трудо- емкость, з.е./часы	В том числе					
		Аудиторных				Самост. работы	
		Всего	Лекций	Лаб. раб.	Практ. зан.		
2	3/108	54	18	36	—	18	экзамен/36

Форма текущей аттестации	Расшифровка	Мин. число баллов	Макс. число баллов
Посещение лекционных занятий		0	10
Выполнение и защита лабораторных работ	4 балла за каждую лабораторную работу	0	24
Тесты по темам 11-13 лекционных занятий	4 балла за каждый тест	0	12
Тесты по темам 14-16 лекционных занятий	4 балла за каждый тест	0	12
Доклад по индивидуальной теме	6 баллов за доклад	0	12
Итого по текущей аттестации		45	70
Промежуточная аттестация	экзамен	10	30
Итого по семестру		55	100

Составитель,
ст. преподаватель

Зав. кафедрой,
доцент

Согласовано:

1. Зав. выпускающей кафедрой,
доцент

2. Декан физико-математического факультета,
доцент



В. И. Великодный



А. В. Коровай

А. В. Коровай

О. В. Коровай