

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

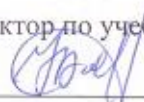
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

**«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра вычислительной математики и кибернетики

УТВЕРЖДАЮ

Проректор по учебной работе



Н.Г.Зарипов

« 12 » 09 20 15 г.

РАБОЧАЯ ПРОГРАММА

УЧЕБНОЙ ДИСЦИПЛИНЫ

**«МОДУЛЬ: МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ
МАШИН, КОМПЛЕКСОВ И КОМПЬЮТЕРНЫХ СЕТЕЙ»**

Уровень подготовки: высшее образование – подготовка кадров высшей квалификации

Направление подготовки научно-педагогических кадров высшей квалификации (аспирантура)

09.06.01 Информатика и вычислительная техника
(код и наименование направления подготовки)

Направленность подготовки

**Математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей**
(наименование программы подготовки)

Квалификация (степень) выпускника

Исследователь. Преподаватель-исследователь

Форма обучения

Очная

Уфа 20 15

Содержание

1.	Место дисциплины в структуре образовательной программы.....	3
2.	Перечень результатов обучения.....	5
3.	Содержание и структура дисциплины (модуля).....	7
4.	Учебно-методическое обеспечение самостоятельной работы.....	10
5.	Фонд оценочных средств.....	11
6.	Учебно-методическое и информационное обеспечение дисциплины (модуля).	18
7.	Образовательные технологии.....	19
8.	Методические указания по освоению дисциплины.....	20
9.	Материально-техническое обеспечение дисциплины.....	27
10.	Адаптация рабочей программы для лиц с ОВЗ.....	27
	Лист согласования рабочей программы дисциплины.....	28
	Дополнения и изменения в рабочей программе дисциплины.....	29

1. Место дисциплины в структуре образовательной программы

Дисциплина *Модуль: Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей* является дисциплиной вариативной части.

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки научно-педагогических кадров высшей квалификации (аспирантура) *09.06.01 Информатика и вычислительная техника*, утвержденного приказом Министерства образования и науки Российской Федерации от "30" июля 2014 г. № 875 и приказа Министерства образования и науки Российской Федерации от 30.04.2015 N 464 "О внесении изменений в федеральные государственные образовательные стандарты высшего образования (уровень подготовки кадров высшей квалификации)". Является неотъемлемой частью основной образовательной профессиональной программы (ОПОП).

Целью освоения дисциплины является формирование у аспирантов общепрофессиональных и профессиональных компетенций в соответствии с ФГОС ВО по направлению подготовки *09.06.01 Информатика и вычислительная техника* и ОПОПВО по направлению подготовки научно-педагогических кадров высшей квалификации (аспирантура).

Задачи: углубленное изучение теоретических и методологических основ математической теории программирования, создания, сопровождения и эксплуатации программных средств различного назначения для вычислительных машин и вычислительных систем, а также построенных на их основе комплексов, компьютерных и нейронных сетей, формирование практических навыков в области программных средств организации и управления обработкой данных и знаний, создания прикладного математического обеспечения, программных средств автоматизации разработки программ и связи человека с ЭВМ.

Дисциплина «Модуль: Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» опирается на дисциплины, входящие в план подготовки студентов по соответствующим программам специалитета, бакалавриата и магистратуры.

Входные компетенции:

№	Компетенция	Код	Уровень освоения, определяемый этапом формирования компетенции*	Название дисциплины (модуля), практики, научных исследований, сформировавших данную компетенцию
1	Владение методологией теоретических и экспериментальных исследований в области профессиональной деятельности;	ОПК1	пороговый	На предыдущем уровне высшего образования (специалитет, магистратура)
2	Способность разрабатывать математическое обеспечение в виде математических моделей объектов, процессов и систем различного типа и современных математических методов, включая методы с применением элементов искусственного интеллекта и его реализация	ПК1	пороговый	На предыдущем уровне высшего образования (специалитет, магистратура)
3	способность создавать, унифицировать и оптимизировать программный код и с целью	ПК2	пороговый	На предыдущем уровне высшего образования

	повышения эффективности процессов обработки данных и знаний			(специалитет, магистратура)
4	способность анализировать качество, надежность программного обеспечения и его соответствия требованиям, спецификациям и стандартам	ПК3	пороговый	На предыдущем уровне высшего образования (специалитет, магистратура)
5	способность проектировать и анализировать архитектуру программных систем	ПК4	пороговый	На предыдущем уровне высшего образования (специалитет, магистратура)

- **пороговый уровень дает общее представление о виде деятельности, основных закономерностях функционирования объектов профессиональной деятельности, методов и алгоритмов решения практических задач;*

*-**базовый уровень** позволяет решать типовые задачи, принимать профессиональные и управленческие решения по известным алгоритмам, правилам и методикам;*

*-**повышенный уровень** предполагает готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные и управленческие решения в условиях неполной определенности, при недостаточном документальном, нормативном и методическом обеспечении.*

Исходящие компетенции:

№	Компетенция	Код	Уровень освоения, определяемый этапом формирования компетенции	Название дисциплины (модуля), практики, научных исследований для которых данная компетенция является входной
1	Владение методологией теоретических и экспериментальных исследований в области профессиональной деятельности;	ОПК1	базовый	Выбор 1 Научно-исследовательская практика
2	Способность разрабатывать математическое обеспечение в виде математических моделей объектов, процессов и систем различного типа и современных математических методов, включая методы с применением элементов искусственного интеллекта и его реализация	ПК1	базовый	Научные исследования
3	способность создавать, унифицировать и оптимизировать программный код и с целью повышения эффективности процессов обработки данных и знаний	ПК2	базовый	Выбор 1 Научно-исследовательская практика Научные исследования
4	способность анализировать качество, надежность программного обеспечения и его соответствия требованиям,	ПК3	базовый	Выбор 1 Научно-исследовательская практика Научные исследования

	спецификациям и стандартам			
5	способность проектировать и анализировать архитектуру программных систем	ПК4	базовый	Выбор 1 Научно-исследовательская практика Научные исследования

2. Перечень результатов обучения

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций.

Планируемые результаты обучения по дисциплине

№	Формируемые компетенции	Код	Знать	Уметь	Владеть
1	владением методологией теоретических и экспериментальных исследований в области профессиональной деятельности;	ОПК1	цели и задачи исследования, основные методологические подходы исследования процессов функционирования объектов профессиональной деятельности; общие принципы и закономерности в построении, функционировании и развитии, управлении и моделировании процессов объектов исследования	использовать методологии и методы научного исследования на уровнях теоретического познания, использования общелогических методов и приемов исследования;	системными правилами выявления причин нарушения системных принципов функционирования объектов исследования
2	Способность разрабатывать математическое обеспечение в виде математических моделей объектов, процессов и систем различного типа и современных математических методов, включая методы с применением элементов искусственного интеллекта и его	ПК1	принципов и проблем создания математического обеспечения программных средств для систем различного назначения и его реализация; создания эффективного математического обеспечения, в том числе разработки алгоритмов;	решать поставленные задачи с использованием современных математических методов и инструментов; разрабатывать модели, методы, алгоритмы для организации взаимодействия программ и программных систем; разрабатывать модели, методы, алгоритмы для реализации машинной графики, визуализации, обработки изображений;	навыками создания математического обеспечения, включая разработку методов и алгоритмов, их модификацию и адаптацию к задаче;

	реализация				
3	способность создавать, унифицировать и оптимизировать программный код и с целью повышения эффективности процессов обработки данных и знаний	ПК2	методов и средств эффективной разработки программного кода; особенностей программирования обмена с окружающей средой; приемы и средства унификации программ с сохранением опыта разработки путем;	формировать требования к используемым технологиям и методикам, проводить их анализ;	владеть методами и инструментами анализа и проектирования программного обеспечения;
4	способность анализировать качество, надежность программного обеспечения и его соответствия требованиям, спецификациям и стандартам	ПК3	методов и технологий тестирования кода; основных принципов управления качеством программного обеспечения; средств верификации программного обеспечения; стандартов качества программного обеспечения; методы повышения надежности функционирования программ	использовать методы и технологии тестирования кода; оценивать качество программного кода; оценивать программный код на соответствие стандартам	владеть методами и инструментами анализа и проектирования; средствами верификации программного обеспечения
5	способность проектировать и анализировать архитектуру программных систем	ПК4	методов проектирования и анализа архитектуры систем; моделей и методов создания программ и программных систем для параллельной и распределенной обработки данных и знаний, языков и инструментальных средств параллельного программирования; моделей, методов, алгоритмов и программной инфраструктуры для организации глобально распределенной обработки данных	проектировать программные системы для параллельной и распределенной обработки данных и знаний; описывать архитектуру системы; применять специализированные методологии для построения архитектуры программных систем;	средствами проектирования программных систем; методами анализа архитектуры программного обеспечения;

3. Содержание и структура дисциплины (модуля)

Общая трудоемкость дисциплины составляет 9 зачетных единиц (324 часов).

Трудоемкость дисциплины по видам работ

Вид работы	Трудоемкость, час.		
	2 семестр	3 семестр	4 семестр
Лекции (Л)	4	6	4
Практические занятия (ПЗ)	6	8	6
Лабораторные работы (ЛР)			
КСР			
Курсовая проект работа (КР)			
Расчетно - графическая работа (РГР)			
Самостоятельная работа (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам, рубежному контролю и т.д.)	89	85	62
Подготовка и сдача экзамена			36
Подготовка и сдача зачета	9	9	
Вид итогового контроля (зачет, экзамен)	3	3	э

Содержание разделов и формы текущего контроля

№	Наименование и содержание раздела	Количество часов						Литература, рекомендуемая студентам*	Виды интерактивных образовательных технологий**
		Аудиторная работа				СРС	Всего		
		Л	ПЗ	ЛР	КСР				
1	Математические основы программирования	4	6			89	99	6.1.1, 6.2.2, 6.3.1	Опережающая самостоятельная работа, лекция-визуализация
2	Вычислительные машины, системы и сети	2	2			30	34	6.1.3, 6.2.1	Опережающая самостоятельная работа, лекция-визуализация
3	Языки и системы программирования. Технологии разработки программного обеспечения.	2	4			25	31	6.1.7.	Опережающая самостоятельная работа, лекция-визуализация
4	Операционные системы	2	2			30	34	6.1.4	Опережающая самостоятельная работа, лекция-визуализация
5	Методы хранения данных и доступа к ним, организация баз данных и знаний	2	4			30	36	6.1.2	Опережающая самостоятельная работа, лекция-визуализация
6	Защита данных и программных систем	2	2			32	36	6.1.3.,6.1.6	Опережающая самостоятельная работа, лекция-визуализация

*Указывается номер источника из соответствующего раздела рабочей программы, раздел (например, Р 6.1 №1, гл.3)

**Указываются образовательные технологии, используемые при реализации различных видов работы.

Занятия, проводимые в интерактивной форме, составляют 100% от общего количества аудиторных часов по дисциплине _____.

Практические занятия (семинары)

№ занятия	№ раздела	Тема	Кол-во часов
1	1	Математические основы программирования	6
2	2	Вычислительные машины, системы и сети	2
3	3	Языки и системы программирования. Технологии разработки программного обеспечения.	4
4	4	Операционные системы	2
5	5	Методы хранения данных и доступа к ним, организация баз данных и знаний	4
6	6	Защита данных и программных систем	2

4. Учебно-методическое обеспечение самостоятельной работы студентов

Тема 1 Математические основы программирования.

Вопросы для самостоятельного изучения (подготовке к обсуждению):

1. Автоматы. Алгебра логики.
2. Формальные языки и способы их описания.
3. Коды с исправлением ошибок. Методы сжатия информации.
1. Основы криптографии.

Тема 2 Вычислительные машины, системы и сети.

Вопросы для самостоятельного изучения:

1. Архитектура современных компьютеров.
2. Классификация вычислительных систем.
3. Протоколы передачи данных.
4. Особенности архитектуры локальных сетей (Ethernet, Token Ring, FDDI).

Тема 3 Языки и системы программирования. Технологии разработки программного обеспечения.

Вопросы для самостоятельного изучения:

1. Языки программирования.
2. Распределенное программирование.
3. Пакеты прикладных программ.
1. Отладка, тестирование, верификация и оценивание сложности программ.

Тема 4 Операционные системы.

Вопросы для самостоятельного изучения:

1. Оптимизация многозадачной работы компьютеров.
2. Операционные средства управления сетями.
3. Удаленный доступ к ресурсам сети.

Тема 5 Методы хранения данных и доступа к ним, организация баз данных и знаний.

Вопросы для самостоятельного изучения:

1. Язык баз данных SQL.
2. Информационно-поисковые системы.
3. Методы представления знаний.

4. Экспертные системы.

Тема 6 Защита данных и программных систем.

Вопросы для самостоятельного изучения:

1. Аппаратные и программные методы защиты данных и программ.
2. Защита информации в вычислительных сетях.

5. Фонд оценочных средств

Оценка уровня освоения дисциплины осуществляется в виде текущего и промежуточного контроля успеваемости аспирантов университета, и на основе критериев оценки уровня освоения дисциплины.

Активность обучающегося на занятиях оценивается на основе выполненных работ и заданий, предусмотренных ФОС дисциплины.

Оценивание проводится преподавателем независимо от наличия или отсутствия обучающегося (по уважительной или неуважительной причине) на занятии. Оценка носит комплексный характер и учитывает достижения обучающегося по основным компонентам учебного процесса за текущий период.

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Уровень освоения, определяемый этапом формирования компетенции	Наименование оценочного средства*
1	Математические основы программирования	ПК1	Базовый	Круглый стол, комплексное задание, ответы на вопросы
2	Вычислительные машины, системы и сети	ОПК1, ПК4	Базовый, Базовый	Круглый стол, ответы на вопросы
3	Языки и системы программирования. Технологии разработки программного обеспечения.	ОПК1 ПК2 ПК3	Базовый, Базовый, Базовый	Круглый стол, комплексное задание, ответы на вопросы
4	Операционные системы	ОПК1	Базовый	Круглый стол, ответы на вопросы
5	Методы хранения данных и доступа к ним, организация баз данных и знаний	ОПК1 ПК1	Базовый, Базовый	Круглый стол, комплексное задание, ответы на вопросы
6	Защита данных и программных систем	ПК1	Базовый	Круглый стол, ответы на вопросы

* Планируемые формы контроля: защита лабораторной работы (ЗЛР), курсовой работы (КР), расчетно-графической работы (РГР), домашнего задания (ДЗ) написание реферата (Р), эссе (Э), тестирование, ответы на вопросы (Т), кейс-анализ (КА) и т.д.

1. Понятие алгоритма и его уточнения: машины Тьюринга, нормальные алгоритмы Маркова, рекурсивные функции. Эквивалентность формальных моделей алгоритмов.
2. Понятие об алгоритмической неразрешимости. Примеры алгоритмически неразрешимых проблем.
3. Понятие сложности алгоритмов. Классы P и NP. Полиномиальная сводимость задач. Теорема Кука об NP-полноте задачи выполнимости булевой формулы. Примеры NP-полных задач, подходы к их решению.
4. Точные и приближенные комбинаторные алгоритмы. Примеры эффективных (полиномиальных) алгоритмов: быстрые алгоритмы поиска и сортировки.
5. Полиномиальные алгоритмы для задач на графах и сетях (поиск в глубину и ширину, о минимальном остове, о кратчайшем пути, о назначениях).
6. Автоматы. Эксперименты с автоматами.
7. Алгебры регулярных выражений. Теорема Клини о регулярных языках.
8. Алгебра логики. Булевы функции, канонические формы задания булевых функций.
9. Понятие полной системы. Критерий полноты Поста. Минимизация булевых функций в классах нормальных форм.
10. Исчисление предикатов первого порядка. Понятие интерпретации. Выполнимость и общезначимость формулы первого порядка. Понятие модели. Теорема о полноте исчисления предикатов первого порядка.
11. Отношения и функции. Отношение эквивалентности и разбиения.
12. Фактор множество. Отношения частичного порядка.
13. Теоретико-множественное и алгебраическое определения решетки, их эквивалентность. Свойства решеток. Булевы решетки. Полные решетки.
14. Формальные языки и способы их описания. Классификация формальных грамматик. Их использование в лексическом и синтаксическом анализе.
15. λ -исчисление, правила редукции, единственность нормальной формы и правила ее достижения, представление рекурсивных функций.
16. Основы комбинаторного анализа. Метод производящих функций, метод включений и исключений. Примеры применения.
17. Коды с исправлением ошибок. Алфавитное кодирование. Методы сжатия информации.
18. Архитектура современных компьютеров. Организации памяти и архитектура процессора современных вычислительных машин.
19. Страничная и сегментная организация виртуальной памяти. Кэш-память.
20. Командный и арифметический конвейеры, параллельное выполнение независимых команд, векторные команды. Специализированные процессоры.
21. Машины, обеспечивающие выполнение вычислений, управляемых потоком данных. Организация ввода-вывода, каналы и процессоры ввода-вывода, устройства сопряжения с объектами.

22. Классификация вычислительных систем (ВС) по способу организации параллельной обработки. Многопроцессорные и многомашинные комплексы. Вычислительные кластеры.
23. Проблемно-ориентированные параллельные структуры: матричные ВС, систолические структуры, нейросети. Назначение, архитектура и принципы построения информационно - вычислительных сетей (ИВС).
24. Локальные и глобальные ИВС, технические и программные средства объединения различных сетей. Методы и средства передачи данных в ИВС, протоколы передачи данных.
25. Особенности архитектуры локальных сетей (Ethernet, TokenRing, FDDI). Сеть Internet, доменная организация, семейство протоколов TCP/IP. Информационно-вычислительные сети и распределенная обработка информации.
26. Языки программирования. Процедурные языки программирования (Фортран, Си). Функциональные языки программирования (Лисп). Логическое программирование (Пролог), Объектно-ориентированные языки программирования (Ява).
27. Процедурные языки программирования. Основные управляющие конструкции, структура программы. Работа с данными: переменные и константы, типы данных (Булевский, целочисленные, плавающие, символьные, типы диапазона и перечисления, указатели), структуры данных (массивы и записи).
28. Процедуры (функции): вызов процедур, передача параметров (по ссылке, по значению, по результату), локализация переменных, побочные эффекты. Обработка исключительных ситуаций. Библиотеки процедур и их использование.
29. Объектно-ориентированное программирование. Классы и объекты, наследование, интерфейсы. Понятие об объектном окружении.
30. Рефлексия. Библиотеки классов. Средства обработки объектов (контейнеры и итераторы).
31. Распределенное программирование. Процессы и их синхронизация. Семафоры, мониторы Хоара.
32. Объектно-ориентированное распределенное программирование. CORBA.
33. Параллельное программирование над общей памятью. Нити.
34. Стандартный интерфейс Open MP. Распараллеливание последовательных программ. Параллельное программирование над распределенной памятью.
35. Парадигмы SPMD и MIMD. Стандартный интерфейс MPI.
36. Основы построения трансляторов. Структура оптимизирующего транслятора.
37. Промежуточные представления программы: последовательность символов, последовательность лексем, синтаксическое дерево, абстрактное синтаксическое дерево.
38. Уровни промежуточного представления: высокий, средний, низкий. Формы промежуточного представления.
39. Анализ исходной программы в компиляторе. Автоматные (регулярные) грамматики и сканирование, контекстно-свободные грамматики и синтаксический анализ, организация таблицы символов программы, имеющей блочную структуру, хеш-функции.
40. Нисходящие и восходящие методы синтаксического анализа.

41. Атрибутные грамматики и семантические программы, построение абстрактного синтаксического дерева.
42. Автоматическое построение лексических и синтаксических анализаторов по формальным описаниям грамматик. Системы lex и yacc. Система Gentle.
43. Оптимизация программ при их компиляции. Оптимизация базовых блоков, чистка циклов.
44. Анализ графов потока управления и потока данных. Отношение доминирования и его свойства, построение границы области доминирования вершины, выделение сильно связанных компонент графа. Построение графа зависимостей.
45. Перевод программы в SSA-представление и обратно. Глобальная и межпроцедурная оптимизация.
46. Генерация объектного кода в компиляторах. Перенастраиваемые (retargetable) компиляторы, gcc (набор компиляторов Gnu).
47. Переработка термов (termrewriting).
48. Применение оптимизационных эвристик (целочисленное программирование, динамическое программирование) для автоматической генерации генераторов объектного кода (системы BEG, Iburg и др.).
49. Машинно-ориентированные языки, язык ассемблера.
50. Представление машинных команд и констант. Команды транслятору. Их типы, принципы реализации.
51. Макросредства, макровыводы, языки макроопределений, условная макрогенерация, принципы реализации.
52. Системы программирования, типовые компоненты СП: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы.
53. Модульное программирование. Типы модулей. Связывание модулей по управлению и данным.
54. Машинная графика. Средства поддержки машинной графики. Графические пакеты.
55. Технология разработки и сопровождения программ. Жизненный цикл программы. Этапы разработки, степень и пути их автоматизации. Обратная инженерия.
56. Декомпозиционные и сборочные технологии, механизмы наследования, инкапсуляции, задания типов.
57. Модули, взаимодействие между модулями, иерархические структуры программ.
58. Отладка, тестирование, верификация и оценивание сложности программ. Генерация тестов. Системы генерации тестов.
59. Срезы программ (slice, chop) и их применение при отладке программ и для генерации тестов.
60. Методы спецификации программ. Методы проверки спецификации.
61. Схемное, структурное, визуальное программирование.
62. Разработка пользовательского интерфейса, стандарт CUA, мультимедийные среды интерфейсного взаимодействия.
63. Режимы функционирования вычислительных систем, структура и функции операционных систем. Основные блоки и модули.

64. Основные средства аппаратной поддержки функций ОС: система прерываний, защита памяти, механизмы преобразования адресов в системах виртуальной памяти, управление каналами и периферийными устройствами.
65. Виды процессов и управления ими в современных ОС. Представление процессов, их контексты, иерархии порождения, состояния и взаимодействие. Многозадачный (многопрограммный) режим работы. Команды управления процессами. Средства взаимодействия процессов.
66. Модель клиент-сервер и ее реализация в современных ОС.
67. Параллельные процессы, схемы порождения и управления.
68. Организация взаимодействия между параллельными и асинхронными процессами: обмен сообщениями, организация почтовых ящиков.
69. Критические участки, примитивы взаимного исключения процессов, семафоры Дейкстры и их расширения.
70. Проблема тупиков при асинхронном выполнении процессов, алгоритмы обнаружения и предотвращения тупиков.
71. Операционные средства управления процессами при их реализации на параллельных и распределенных вычислительных системах и сетях: стандарты и программные средства PVM, MPI, OpenMP, POSIX.
72. Одноуровневые и многоуровневые дисциплины циклического обслуживания процессов на центральном процессоре, выбор кванта.
73. Управление доступом к данным. Файловая система, организация, распределение дисковой памяти. Управление обменом данными между дисковой и оперативной памятью. Рабочее множество страниц (сегментов) программы, алгоритмы его определения.
74. Управление внешними устройствами.
75. Оптимизация многозадачной работы компьютеров. Операционные системы Windows, Unix, Linux. Особенности организации, предоставляемые услуги пользовательского взаимодействия.
76. Операционные средства управления сетями. Эталонная модель взаимодействия открытых систем ISO/OSI. Маршрутизация и управление потоками данных в сети.
77. Локальные и глобальные сети. Сетевые ОС, модель “клиент - сервер”, средства управления сетями в ОС UNIX, Windows.
78. Семейство протоколов TCP/IP, структура и типы IP – адресов, доменная адресация в Internet. Транспортные протоколы TCP, UDP .
79. Удаленный доступ к ресурсам сети. Организация электронной почты, телеконференций. Протоколы передачи файлов FTP и http. Язык разметки гипертекста HTML, разработка WEB-страниц, WWW- серверы.
80. Концепция типа данных. Абстрактные типы данных. Объекты (основные свойства и отличительные признаки).
81. Основные структуры данных, алгоритмы обработки и поиска.
82. Сравнительная характеристика методов хранения и поиска данных. Основные понятия реляционной и объектной моделей данных. Теоретические основы реляционной модели данных (РДМ). Реляционная алгебра, реляционное исчисление.
83. Функциональные зависимости и нормализация отношений. CASE- средства и их использование при проектировании БД.

84. Организация и проектирование физического уровня БД. Методы индексирования. Обобщенная архитектура, состав и функции системы управления базой данных (СУБД).
85. Характеристика современных технологий БД. Примеры соответствующих СУБД.
86. Основные принципы управления транзакциями, журнализацией и восстановлением.
87. Язык баз данных SQL. Средства определения и изменения схемы БД, определения ограничений целостности. Контроль доступа. Средства манипулирования данными. Стандарты языков SQL. Интерактивный, встроенный, динамический SQL.
88. Основные понятия технологии клиент-сервер. Характеристика SQL-сервера и клиента. Сетевое взаимодействие клиента и сервера.
89. Информационно-поисковые системы. Классификация. Методы реализации и ускорения поиска.
90. Методы представления знаний: процедурные представления, логические представления, семантические сети, фреймы, системы продукций. Интегрированные методы представления знаний.
91. Языки представления знаний. Базы знаний. Экспертные системы (ЭС). Области применения ЭС. Архитектура ЭС. Механизмы вывода, подсистемы объяснения, общения, приобретения знаний ЭС. Жизненный цикл экспертной системы. Примеры конкретных ЭС.
92. Аппаратные и программные методы защиты данных и программ. Защита данных и программ с помощью шифрования.
93. Защита от несанкционированного доступа в ОС Windows. Система безопасности и разграничения доступа к ресурсам в Windows. Файловая система и сервисы Windows.
94. Защита от несанкционированного копирования.
95. Методы простановки не копируемых меток, настройка устанавливаемой программы на конкретный компьютер, настройка на конфигурацию оборудования.
96. Защита от разрушающих программных воздействий. Вредоносные программы и их классификация. Загрузочные и файловые вирусы, программы-закладки.
97. Методы обнаружения и удаления вирусов, восстановления программного обеспечения. Защита информации в вычислительных сетях Novell Netware, Windows и других.
98. Основы криптографии. Задачи обеспечения конфиденциальности и целостности информации. Теоретико-информационный и теоретико-сложностной подходы к определению криптографической стойкости.
99. Американский стандарт шифрования DES и российский стандарт шифрования данных ГОСТ 28147-89.
100. Системы шифрования с открытым ключом (RSA). Цифровая подпись. Методы генерации и распределения ключей.

Критерии оценки:

- оценка «отлично» выставляется студенту, если была проявлена дискуссионная активность в рамках круглого стола, получены полные ответы на вопросы (теоретические), корректно решены задания;
- оценка «хорошо» - была проявлена дискуссионная активность в рамках круглого стола, не более половины ответов на теоретические вопросы не совсем полные, корректно решены задания;
- оценка «удовлетворительно» - была проявлена дискуссионная активность в рамках круглого стола, ответы на более чем половина теоретических вопросов не совсем полные, часть заданий решена корректно;
- оценка «неудовлетворительно» - отсутствовала дискуссионная активность в рамках круглого стола, ответы на теоретические вопросы неправильные, задания решены некорректно;
- оценка «зачтено» выставляется студенту, если была проявлена дискуссионная активность в рамках круглого стола, задания решены корректно;
- оценка «не зачтено» - отсутствовала дискуссионная активность в рамках круглого стола, задания решены некорректно.

Типовые оценочные материалы

Раздел (тема) дисциплины Математические основы программирования

Перечень дискуссионных тем для круглого стола (дискуссии, полемики, диспута, дебатов)

1. Автоматы. Алгебра логики.
2. Формальные языки и способы их описания.
3. Коды с исправлением ошибок. Методы сжатия информации.
4. Основы криптографии.

Комплексное задание: решить задачи с использованием методов проверки на четность по вертикали и горизонтали, линейный код.

1. Найдите пример комбинации из 6 ошибок, которая не может быть обнаружена при использовании горизонтальных и вертикальных проверок на четность. (Каждая строка с ошибками и каждый столбец с ошибками будут содержать ровно 2 ошибки).
2. Рассмотрим код с проверкой на четность. Предположим, что есть три кодовых слова 1001011, 0101101 и 0011110.

Найдите правило, по которому выполняется проверка на четность, и все восемь кодовых слов.

Критерии оценки:

- оценка «зачтено» выставляется студенту, если была проявлена дискуссионная активность в рамках круглого стола;
- оценка «не зачтено» - отсутствовала дискуссионная активность в рамках круглого стола.

Раздел (тема) дисциплины Вычислительные машины, системы и сети

Перечень дискуссионных тем для круглого стола (дискуссии, полемики, диспута, дебатов)

1. Архитектура современных компьютеров.
2. Классификация вычислительных систем.
3. Протоколы передачи данных.
4. Особенности архитектуры локальных сетей (Ethernet, TokenRing, FDDI).

Критерии оценки:

- оценка «зачтено» выставляется студенту, если была проявлена дискуссионная активность в рамках круглого стола;
- оценка «не зачтено» - отсутствовала дискуссионная активность в рамках круглого стола.

Раздел (тема) дисциплины Языки и системы программирования. Технологии разработки программного обеспечения

Перечень дискуссионных тем для круглого стола (дискуссии, полемики, диспута, дебатов)

1. Языки программирования.
2. Распределенное программирование.
3. Пакеты прикладных программ.
4. Отладка, тестирование, верификация и оценивание сложности программ.

Комплексное задание: оценить надежность программы.

Модель Коркорэна. Задача.

Оттестировать и оценить надежность по модели Коркорэна. Было проведено 100 испытаний программы. 20 из 100 испытаний прошли безуспешно, а в остальных случаях получились следующие данные:

№	Тип ошибки	Вероятность появления	Вероятность появления ошибки при испытании N_i
1	Ошибки вычисления	0,09	5
2	Логические ошибки	0,26	25
3	Ошибки ввода /вывода	0,16	3
4	Ошибки манипулирования данными	0,18	-
5	Ошибки сопряжения	0,17	11
6	Ошибки определения данных	0,08	3
7	Ошибки в БД	0,06	4

Модель Шумана

Пример. В программе имеется $It=4381$ оператор. В процессе последовательных тестовых прогонов были получены следующие данные:

№ прогона	1	2А	3	4	5	6	7	8В	9	10
Количество ошибок	1	2	1	1	1	1	1	2	1	1
Время (м)	5	8	2	1	5	1	1	2	5	5

Критерии оценки:

- оценка «зачтено» выставляется студенту, если была проявлена дискуссионная активность в рамках круглого стола;
- оценка «не зачтено» - отсутствовала дискуссионная активность в рамках круглого стола.

Раздел (тема) дисциплины Операционные системы**Перечень дискуссионных тем для круглого стола (дискуссии, полемики, диспута, дебатов)**

1. Оптимизация многозадачной работы компьютеров.
2. Операционные средства управления сетями.
3. Удаленный доступ к ресурсам сети.

Критерии оценки:

- оценка «зачтено» выставляется студенту, если была проявлена дискуссионная активность в рамках круглого стола;
- оценка «не зачтено» - отсутствовала дискуссионная активность в рамках круглого стола.

Раздел (тема) дисциплины Методы хранения данных и доступа к ним, организация баз данных и знаний

Перечень дискуссионных тем для круглого стола (дискуссии, полемики, диспута, дебатов)

1. Язык баз данных SQL.
2. Информационно-поисковые системы.
3. Методы представления знаний.
4. Экспертные системы.

Комплексное задание

Представить декларативное знание о понятии «Квартира» двумя моделями представления знаний:

1. в виде семантической сети.
2. в виде продукционной модели.

Квартира состоит из:

Кухня.
Гостиная.
Прихожая.
Спальня.
Детская.
Санитарный узел (туалет).
Ванная комната.
Кладовка.
Гардеробная.
Комната отдыха (игровая комната).
Спортивная комната (тренажерная).
Бытовая комната.

Критерии оценки:

- оценка «зачтено» выставляется студенту, если была проявлена дискуссионная активность в рамках круглого стола;
- оценка «не зачтено» - отсутствовала дискуссионная активность в рамках круглого стола.

Раздел (тема) дисциплины Защита данных и программных систем

Перечень дискуссионных тем для круглого стола (дискуссии, полемики, диспута, дебатов)

1. Аппаратные и программные методы защиты данных и программ.
2. Защита информации в вычислительных сетях.

Комплексное задание

1. Реализуйте на одном из языков программирования следующую задачу: известен ключ – 1001001110111101110. Закодировать последовательность 1001110011100111001 для передачи по каналу связи.
2. Реализуйте на одном из языков программирования следующую задачу: известен ключ – 1111001101110010110. Раскодировать последовательность 1001110011100111001, принятую по каналу связи.
3. Реализуйте на одном из языков программирования следующую задачу: известна переданная последовательность 1111100000, известна принятая последовательность 0101010101. Найдите ключ, при помощи которого было закодировано сообщение.
4. Реализуйте на одном из языков программирования следующую задачу: последовательность закодировать и вывести результат на экран вместе с ключом, предварительно составив любую последовательность и ключ. Затем последовательность раскодировать при помощи ключа. В конце произведенных действий проверить правильность дешифровки.

Критерии оценки:

- оценка «зачтено» выставляется студенту, если была проявлена дискуссионная активность в рамках круглого стола;
- оценка «не зачтено» - отсутствовала дискуссионная активность в рамках круглого стола.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

6.1 Основная литература

1. Ершов, Ю. Л. Математическая логика: учебное пособие / Ю. Л. Ершов, Е. А. Палютин. - 4-е изд., стер. - СПб; М.: Лань: 2005. - 336 с.
2. Советов, Б. Я. Базы данных / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской; Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В. И. Ульянова (Ленина). - 2-е изд. - Москва: Юрайт, 2015 - 463 с.
3. Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации / В. Л. Бройдо, О. П. Ильина. - 4-е изд. - Санкт-Петербург [и др.]: Питер, 2011. - 554.
4. Партыка, Т. Л. Операционные системы, среды и оболочки: [учебное пособие] / Т. Л. Партыка, И. И. Попов. - 5-е изд., перераб. и доп. — Москва: ИНФРА-М: ФОРУМ, 2013. - 560 с.
5. Юсупова Н.И., Шахмаметова Г.Р., Сметанина О.Н., Гаянова М.М., Еникеева К.Р. / Экспертные системы Учебное пособие. – Уфа: УГАТУ, 2014 – 89 с.
6. Гузаиров М.М., Юсупова Н.И., Сметанина О.Н., Шахмаметова Г.Р. / Информационные системы и технологии. – Москва: Машиностроение, 2013. – 319 с.
7. Орлов, С. А. Технологии разработки программного обеспечения. Современный курс по программной инженерии : [учебник для студ. вузов, обуч. по спец. "Программное обеспечение вычислит. техники и автоматизир. систем" напр. подготовки дипломирован. спец. "Информатика и вычислительная техника"] / С. А. Орлов, Б. Я. Цилькер. - 4-е изд. - СПб.: Питер, 2012. http://www.library.ugatu.ac.ru/pdf/teach/Orlov_Tehnolog_razrab_progr_obespech_Sovr_4izd_2012.pdf

6.2 Дополнительная литература

1. Гусева, А. И. Вычислительные системы, сети и телекоммуникации / А. И. Гусева, В. С. Киреев. - Москва: Академия, 2014. - 288 с.
2. Орехов, Э. Ю. Введение в теорию сложности решения задач: [учебное пособие для студентов всех форм обучения, обучающихся по специальности 010503 - "Математическое обеспечение и администрирование информационных систем"] / Э. Ю. Орехов, Ю. В. Орехов; ГОУ ВПО УГАТУ. - Уфа: УГАТУ, 2008. – 87 С.
3. Орехов, Ю. В. Математическая логика: учебное пособие / Ю. В. Орехов, Э. Ю. Орехов; Уфимский государственный авиационный технический университет; науч. ред. Э. А. Мухачева. — Уфа: УГАТУ, 2006. - 161 с.

6.3. Интернет-ресурсы (электронные учебно-методические издания, лицензионное программное обеспечение)

На сайте библиотеки <http://library.ugatu.ac.ru/> в разделе «Информационные ресурсы», подраздел «Доступ к БД» размещены ссылки на интернет-ресурсы.

1. Козлов, А.Н. Интеллектуальные информационные системы: учебник / А.Н. Козлов; Мин-во с-х. РФ, ФГБОУ ВПО Пермская ГСХА. – Пермь: Изд-во ФГБОУ ВПО Пермская ГСХА, 2013. – 278 с. <http://pgsha.ru:8008/books/study.pdf>

6.4 Методические указания к практическим занятиям

1. О.Н. Сметанина. Методические указания к практическим занятиям по курсу «Модуль: математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» (электр. вариант).

7. Образовательные технологии

№	Наименование	Доступ, количество одновременных пользователей	Реквизиты договоров с правообладателями
Программного комплекса			
1	операционная система Microsoft Windows	1800 компьютеров	договор ЭФ-193/0503-14
2	Microsoft Office	1800 компьютеров	договор ЭФ-193/0503-14

8. Методические указания по освоению дисциплины

Раздел Математические основы программирования

Знания: Понятие алгоритма и его уточнения. Понятие сложности алгоритмов. Классы P и NP. Автоматы. Алгебра логики. Отношения и функции. Отношения частичного порядка. Формальные языки и способы их описания. Коды с исправлением ошибок. Методы сжатия информации. Основы криптографии.

Лекция: №1-2, **Объем лекций:** 4 часа

На изучение материала данной темы отводится 4 часа лекционных занятий, 6 часа практических занятий и 89 часов самостоятельной работы.

При самостоятельной проработке материалов темы 1 необходимо использовать:

- учебное пособие 6.1.1, 6.2.3, 6.2.4;
- презентации № 1, 2 лекционного курса.

При изучении материалов темы 1 необходимо акцентировать внимание на следующих понятиях:

- Понятие алгоритма и его уточнения: машины Тьюринга, нормальные алгоритмы Маркова, рекурсивные функции.
- Эквивалентность формальных моделей алгоритмов.
- Понятие об алгоритмической неразрешимости. Примеры алгоритмически неразрешимых проблем.
- Понятие сложности алгоритмов.
- Классы P и NP.
- Полиномиальная сводимость задач.
- Теорема Кука об NP-полноте задачи выполнимости булевой формулы.
- Примеры NP-полных задач, подходы к их решению.
- Точные и приближенные комбинаторные алгоритмы.
- Примеры эффективных (полиномиальных) алгоритмов: быстрые алгоритмы поиска и сортировки.
- Полиномиальные алгоритмы для задач на графах и сетях (поиск в глубину и ширину, о минимальном остове, о кратчайшем пути, о назначениях).
- Автоматы.
- Эксперименты с автоматами.
- Алгебры регулярных выражений.
- Теорема Клини о регулярных языках.
- Алгебра логики.
- Булевы функции, канонические формы задания булевых функций.
- Понятие полной системы.

- Критерий полноты Поста.
- Минимизация булевых функций в классах нормальных форм.
- Исчисление предикатов первого порядка.
- Понятие интерпретации.
- Выполнимость и общезначимость формулы первого порядка.
- Понятие модели.
- Теорема о полноте исчисления предикатов первого порядка.
- Отношения и функции.
- Отношение эквивалентности и разбиения.
- Фактор множество.
- Отношения частичного порядка.
- Теоретико-множественное и алгебраическое определения решетки, их эквивалентность.
- Свойства решеток.
- Булевы решетки.
- Полные решетки.
- Формальные языки и способы их описания.
- Классификация формальных грамматик. Их использование в лексическом и синтаксическом анализе.
- λ -исчисление, правила редукции, единственность нормальной формы и правила ее достижения, представление рекурсивных функций.
- Основы комбинаторного анализа.
- Метод производящих функций, метод включений и исключений.
- Примеры применения.
- Коды с исправлением ошибок.
- Алфавитное кодирование.
- Методы сжатия информации.

При выполнении практического задания по теме 1 необходимо подготовиться для проведения дискуссии в рамках круглого стола по следующим темам:

1. Автоматы. Алгебра логики.
2. Формальные языки и способы их описания.
3. Коды с исправлением ошибок. Методы сжатия информации.
4. Основы криптографии.

Комплексное задание: решить задачи с использованием методов проверок на четность по вертикали и горизонтали, линейный код.

1. Найдите пример комбинации из 6 ошибок, которая не может быть обнаружена при использовании горизонтальных и вертикальных проверок на четность. (Каждая строка с ошибками и каждый столбец с ошибками будут содержать ровно 2 ошибки).

Бит проверки на четность имеет значение 1, если число единиц в последовательности битов нечетное, и значение 0 – в противном случае, иначе можно говорить о том, что контрольный бит равен сумме по модулю 2 значений битов в исходной последовательности битов.

Каждый бит проверки на четность по горизонтали проверяет свою строку, каждый бит проверки по вертикали проверяет свой столбец. Если каждый выделенный бит изменяет значение на противоположное, то четность в каждой строке и в каждом столбце, тем не менее сохраняется.

							Проверки по горизонталям
1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
1	0	1	1	1	1	1	0
Проверки по вертикалям							

А)

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
1	0	1	1	1	1	1	0

Б)

Бит проверки на четность в правом нижнем углу может относиться к строке или столбцу битов проверки на четность или же к массиву данных.

Если четное число ошибок находится в пределах одной строки, то каждую из них можно обнаружить с помощью соответствующего бита проверки на четность по столбцу; аналогично ошибки только в одном столбце можно обнаружить с помощью битов проверки на четность по строкам.

К сожалению, любой набор из четырех ошибок, принадлежащих двум строкам и двум столбцам, обнаружить невозможно.

Решение:

	V	V			
		V		V	
	V			V	

Одна из комбинаций из 6 ошибок, которые не будут обнаружены.

2. Рассмотрим код с проверкой на четность. Предположим, что есть три кодовых слова 1001011, 0101101 и 0011110.

Найдите правило, по которому выполняется проверка на четность, и все восемь кодовых слов.

Преобразование последовательности битов данных в последовательность битов данных и битов проверки на четность называется кодом с проверкой на четность или линейным кодом.

Код с проверкой на четность определяется конкретным набором подмножеств, используемых для выполнения проверок на четность. Слово код относится к самому преобразованию; закодированную последовательность битов (данные плюс проверки на четность) называют кодовым словом.

S ₁	S ₂	S ₃	C ₁	C ₂	C ₃	C ₄	
1	0	0	1	1	1	0	C ₁ =S ₁ +S ₃
0	1	0	0	1	1	1	C ₂ =S ₁ +S ₂ +S ₃
0	0	1	1	1	0	1	C ₃ =S ₁ +S ₂
1	1	0	1	0	0	1	C ₄ =S ₂ +S ₃
1	0	1	0	0	1	1	
1	1	1	0	1	0	0	
0	0	0	0	0	0	0	

Решение:

1001011

0101101

0011110

C₁ – во все трех случаях представлено 1. При сложении по модулю 2 символов кода можно получить, если C₁=S₁+S₂+S₃. Получить 0 в первой комбинации и 1 во второй и третьей можно, если C₂=S₂+S₃. Получить 1 в первой и в третьей комбинациях и 0 во второй можно, если C₃=S₁+S₃. Получить 1 в первой и во второй комбинациях и 0 в третьей можно, если C₄=S₁+S₂.

Восемь (2³) кодовых слов можно получить добавляя к кодам (000, 001, 010, 011, 100, 101, 110, 111) биты проверки на четность по правилам:

0000000

0011110

0101101

0110011

1001011

1010101

1100110

1111000.

Раздел Вычислительные машины, системы и сети.

Знания: Архитектура современных компьютеров. Классификация вычислительных систем. Назначение, архитектура и принципы построения информационно-вычислительных сетей. Протоколы передачи данных. Особенности архитектуры локальных сетей (Ethernet, TokenRing, FDDI). Сеть Internet, доменная организация, семейство протоколов TCP/IP.

Лекция: №3, Объем лекций: 2 часа

На изучение материала данной темы отводится 2 часа лекционных занятий, 2 часа практических занятий и 30 часов самостоятельной работы.

При самостоятельной проработке материалов темы 2 необходимо использовать:

- учебное пособие 6.1.3, 6.2.1;
- презентации № 3 лекционного курса.

При изучении материалов темы 2 необходимо акцентировать внимание на следующих понятиях:

- Архитектура современных компьютеров.
- Организации памяти и архитектура процессора современных вычислительных машин.
- Страничная и сегментная организация виртуальной памяти.
- Кэш-память.
- Командный и арифметический конвейеры, параллельное выполнение независимых команд, векторные команды.
- Специализированные процессоры.
- Машины, обеспечивающие выполнение вычислений, управляемых потоком данных.

- Организация ввода-вывода, каналы и процессоры ввода-вывода, устройства сопряжения с объектами.
- Классификация вычислительных систем (ВС) по способу организации параллельной обработки.
- Многопроцессорные и многомашинные комплексы.
- Вычислительные кластеры.
- Проблемно-ориентированные параллельные структуры: матричные ВС, систолические структуры, нейросети.
- Назначение, архитектура и принципы построения информационно - вычислительных сетей (ИВС).
- Локальные и глобальные ИВС, технические и программные средства объединения различных сетей.
- Методы и средства передачи данных в ИВС, протоколы передачи данных.
- Особенности архитектуры локальных сетей (Ethernet, TokenRing, FDDI).
- Сеть Internet, доменная организация, семейство протоколов TCP/IP.
- Информационно-вычислительные сети и распределенная обработка информации.

При выполнении практического задания по теме 2 необходимо подготовиться для проведения дискуссии в рамках круглого стола по следующим темам:

1. Архитектура современных компьютеров.
2. Классификация вычислительных систем.
3. Протоколы передачи данных.
4. Особенности архитектуры локальных сетей (Ethernet, TokenRing, FDDI).

Раздел Языки и системы программирования. Технологии разработки программного обеспечения.

Знания: Языки программирования. Распределенное программирование. Параллельное программирование над общей памятью. Параллельное программирование над распределенной памятью. Основы построения трансляторов. Пакеты прикладных программ. Отладка, тестирование, верификация и оценивание сложности программ. Методы спецификации программ.

Лекция: №4, Объем лекций: 4 часа

На изучение материала данной темы отводится 2 часа лекционных занятий, 4 часа практических занятий и 25 часов самостоятельной работы.

При самостоятельной проработке материалов темы 3 необходимо использовать:

- учебное пособие 6.1.7;
- презентации № 4 лекционного курса.

При изучении материалов темы 3 необходимо акцентировать внимание на следующих процессах:

- Языки программирования.
- Процедурные языки программирования (Фортран, Си).
- Функциональные языки программирования (Лисп).
- Логическое программирование (Пролог).
- Объектно-ориентированные языки программирования (Ява).
- Процедурные языки программирования.
- Основные управляющие конструкции, структура программы.
- Работа с данными: переменные и константы, типы данных (Булевский, целочисленные, плавающие, символьные, типы диапазона и перечисления, указатели), структуры данных (массивы и записи).
- Процедуры (функции): вызов процедур, передача параметров (по ссылке, по значению, по результату), локализация переменных, побочные эффекты.
- Обработка исключительных ситуаций. Библиотеки процедур и их использование.
- Объектно-ориентированное программирование.
- Классы и объекты, наследование, интерфейсы. Понятие об объектном окружении.

- Рефлексия. Библиотеки классов. Средства обработки объектов (контейнеры и итераторы).
- Распределенное программирование.
- Процессы и их синхронизация.
- Семафоры, мониторы Хоара.
- Объектно-ориентированное распределенное программирование. CORBA.
- Параллельное программирование над общей памятью. Нити.
- Стандартный интерфейс Open MP.
- Распараллеливание последовательных программ.
- Параллельное программирование над распределенной памятью.
- Парадигмы SPMD и MIMD.
- Стандартный интерфейс MPI.
- Основы построения трансляторов.
- Структура оптимизирующего транслятора.
- Промежуточные представления программы: последовательность символов, последовательность лексем, синтаксическое дерево, абстрактное синтаксическое дерево.
- Уровни промежуточного представления: высокий, средний, низкий. Формы промежуточного представления.
- Анализ исходной программы в компиляторе.
- Автоматные (регулярные) грамматики и сканирование, контекстно-свободные грамматики и синтаксический анализ, организация таблицы символов программы, имеющей блочную структуру, хеш-функции.
- Нисходящие и восходящие методы синтаксического анализа.
- Атрибутные грамматики и семантические программы, построение абстрактного синтаксического дерева.
- Автоматическое построение лексических и синтаксических анализаторов по формальным описаниям грамматик.
- Системы lex и yacc. Система Gentle.
- Оптимизация программ при их компиляции.
- Оптимизация базовых блоков, чистка циклов.
- Анализ графов потока управления и потока данных.
- Отношение доминирования и его свойства, построение границы области доминирования вершины, выделение сильно связанных компонент графа.
- Построение графа зависимостей.
- Перевод программы в SSA-представление и обратно. Глобальная и межпроцедурная оптимизация.
- Генерация объектного кода в компиляторах.
- Перенастраиваемые (retargetable) компиляторы, gcc (набор компиляторов Gnu).
- Переработка термов (termrewriting).
- Применение оптимизационных эвристик (целочисленное программирование, динамическое программирование) для автоматической генерации генераторов объектного кода (системы BEG, Iburg и др.).
- Машинно-ориентированные языки, язык ассемблера.
- Представление машинных команд и констант.
- Команды транслятору. Их типы, принципы реализации.
- Макросредства, макровыводы, языки макроопределений, условная макрогенерация, принципы реализации.
- Системы программирования, типовые компоненты СП: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы.
- Модульное программирование.
- Типы модулей. Связывание модулей по управлению и данным.
- Машинная графика. Средства поддержки машинной графики. Графические пакеты.

- Технология разработки и сопровождения программ.
- Жизненный цикл программы.
- Этапы разработки, степень и пути их автоматизации.
- Обратная инженерия.
- Декомпозиционные и сборочные технологии, механизмы наследования, инкапсуляции, задания типов.
- Модули, взаимодействие между модулями, иерархические структуры программ.
- Отладка, тестирование, верификация и оценивание сложности программ.
- Генерация тестов.
- Системы генерации тестов.
- Срезы программ (slice, chop) и их применение при отладке программ и для генерации тестов.
- Методы спецификации программ.
- Методы проверки спецификации.
- Схемное, структурное, визуальное программирование.
- Разработка пользовательского интерфейса, стандарт CUA, мультимедийные среды интерфейсного взаимодействия.

При выполнении практического задания по теме 3 необходимо подготовиться для проведения дискуссии в рамках круглого стола по следующим темам:

1. Языки программирования.
2. Распределенное программирование.
3. Пакеты прикладных программ.
4. Отладка, тестирование, верификация и оценивание сложности, надежности программ.

Комплексное задание: оценить надежность программы.

Модель Коркорэна

Задача

Оттестировать и оценить надежность по модели Коркорэна. Было проведено 100 испытаний программы. 20 из 100 испытаний прошли безуспешно, а в остальных случаях получились следующие данные:

№	Тип ошибки	Вероятность появления	Вероятность появления ошибки при испытании N_i
1	Ошибки вычисления	0,09	5
2	Логические ошибки	0,26	25
3	Ошибки ввода /вывода	0,16	3
4	Ошибки манипулирования данными	0,18	-
5	Ошибки сопряжения	0,17	11
6	Ошибки определения данных	0,08	3
7	Ошибки в БД	0,06	4

Применение модели предполагает знание следующих ее показателей:

– модель содержит изменяющуюся вероятность отказов для различных источников ошибок и соответственно разную вероятность их исправления;

- в модели используются такие параметры, как результат только N испытаний, в которых наблюдается N_i ошибок i -го типа;
- выявление в ходе N испытаний ошибки i -го типа появляется с вероятностью a_i .

Показатель уровня надежности R вычисляют по следующей формуле:

$$R = \frac{N_0}{N} + \sum_{i=1}^k \frac{Y_i * (N_i - 1)}{N},$$

Где N_0 – число безотказных (или безуспешных) испытаний, выполненных в серии из N испытаний,

k – известное число типов ошибок,

Y_i – вероятность появления ошибок,

при $N_i > 0$, $Y_i = a_i$,

при $N_i = 0$, $Y_i = 0$.

Модель Шумана

Модель Шумана относится к динамическим моделям дискретного времени, данные для которой собираются в процессе тестирования программного обеспечения в течение фиксированных или случайных интервалов времени. Модель Шумана предполагает, что тестирование проводится в несколько этапов. Каждый этап представляет собой выполнение программы на полном комплексе разработанных тестовых данных. Выявленные ошибки регистрируются, но не исправляются. В конце этапа рассчитываются количественные показатели надежности, исправляются найденные ошибки, корректируются тестовые наборы и проводится следующий этап тестирования. В модели Шумана предполагается, что число ошибок в программе постоянно и в процессе корректировки новые ошибки не вносятся. Скорость обнаружения ошибок пропорциональна числу оставшихся ошибок. Предполагается, что до начала тестирования имеется Et ошибок. В течение времени тестирования τ обнаруживается ε_c ошибок в расчете на одну команду в машинном языке. Таким образом, удельное число ошибок на одну машинную команду, оставшихся в системе после τ времени тестирования, равно:

$$\varepsilon_r(\tau) = Et / It * \varepsilon_c(\tau),$$

где It – общее число машинных команд, которое предполагается постоянным в рамках этапа тестирования.

Предполагается, что значение функции частоты отказов $Z(t)$ пропорционально числу ошибок, оставшихся в программе после израсходованного на тестирование времени τ .

$$Z(t) = C * \varepsilon_r(\tau),$$

Где C – некоторая постоянная;

t – время работы программы без отказов.

Тогда, если время работы программы без отказа t отсчитывается от точки $t = 0$, а t остается фиксированным, функция надежности, или вероятность безотказной работы на интервале от 0 до t , равна

$$R(t, \tau) = \exp \{ - C * [Et / It - \varepsilon_c(\tau)] * t \} (1) t_{cp} = 1 / \{ C * [Et / It - \varepsilon_c(\tau)] \}. \quad (2)$$

Нам необходимо найти начальное значение ошибок Et и коэффициент пропорциональности – C . В процессе тестирования собирается информация о

времени и количестве ошибок на каждом прогоне, т.е. общее время тестирования τ складывается из времени каждого прогона $\tau = \tau_1 + \tau_2 + \tau_3 + \dots + \tau_n$.

Предполагая, что интенсивность появления ошибок постоянна и равна λ , можно вычислить ее как число ошибок в единицу времени

$$\lambda = \frac{\sum_{i=1}^k A_i}{\tau}, \quad (3)$$

где A_i – количество ошибок на i -ом прогоне.

Имея данные для двух различных моментов тестирования τ_a и τ_b , которые выбираются произвольно с учетом требования, чтобы $\varepsilon_c(\tau_b) > \varepsilon_c(\tau_a)$, можно сопоставить уравнения (2) и (4) при τ_a и τ_b .

$$t_{cp} = \frac{\tau}{\sum_{i=1}^k A_i} \cdot (4) \frac{1}{\lambda_{\tau_a}} = \frac{1}{C[Et/It - \varepsilon_c(\tau_a)]}; \quad (5) \frac{1}{\lambda_{\tau_b}} = \frac{1}{C[Et/It - \varepsilon_c(\tau_b)]}. \quad (6)$$

Неизвестный параметр C (8) получается путем подстановки Et (7) в выражение (5). Вычисляя соотношения (5) и (6) получим

$$Et = \frac{It[\lambda_{\tau_b}/\lambda_{\tau_a}\varepsilon_c(\tau_b) - \varepsilon_c(\tau_b)]}{(\lambda_{\tau_b}/\lambda_{\tau_a}) - 1}. \quad (7) \quad C = \frac{\lambda_{\tau_a}}{[Et/It - \varepsilon_c(\tau_a)]}. \quad (8)$$

Получив неизвестные Et и C , можно рассчитать надежность программы по формуле (1), проведем расчеты применительно к учебной программе.

Например, в программе имеется $It = 4381$ оператор. В процессе последовательных тестовых прогонов были получены следующие данные:

№ прогона	1	2А	3	4	5	6	7	8В	9	10
Количество ошибок	1	2	1	1	1	1	1	2	1	1
Время (м)	5	8	2	1	5	1	1	2	5	5

Выберем две точки, исходя из требования, чтобы число ошибок, найденных на интервале $A \div B$, и было больше, чем на интервале $0 \div A$. За точку A возьмем 2 прогон, а за точку B – 8 прогон. Тогда ошибки, найденные на этапах тестирования на интервалах $0 \div A$ и $A \div B$, будут равны соответственно:

$$\varepsilon_c(\tau_a) = 3/4381 = 0,0007$$

$$\varepsilon_c(\tau_b) = 7/4381 = 0,0015$$

Время тестирования на интервалах равно: $\tau_a = 13$, $\tau_b = 12$.

Рассчитаем интенсивности появления ошибок на двух интервалах:

$$\lambda_a = 3/13 = 0,23$$

$$\lambda_b = 7/12 = 0,58$$

$$Et = \frac{4381(0,58/0,23 * 0,0007 - 0,0015)}{0,58/0,23 - 1} = 0,763 \approx 1 \text{ ошибка. Тогда число имеющихся до}$$

$$\text{начала тестирования ошибок равно } C = \frac{0,23}{0,76/4381 - 0,0007} = -460.$$

Рассчитаем вероятность безотказной работы в течение времени t при $\tau = 35 \text{ мин.}$

$$R(t, 35) = \exp \{460 * [0,763/4381 - \varepsilon_c(35)]t\}$$

Возьмем $t = 60 \text{ мин.}$

$$R(60,35) = \exp \{460 * [0,0002 - 0,0027]60\} \approx 0,9$$

Таким образом, надежность безотказной работы достаточно велика и вероятность сбоев и возникновения ошибок небольшая.

Раздел Операционные системы.

Знания: Виды процессов и управления ими в современных ОС. Параллельные процессы, схемы порождения и управления. Управление доступом к данным. Управление внешними устройствами. Оптимизация многозадачной работы компьютеров. Операционные средства управления сетями. Удаленный доступ к ресурсам сети.

Лекция: №5, Объем лекций: 2 часа

На изучение материала данной темы отводится 2 часа лекционных занятий, 2 часа практических занятий и 30 часов самостоятельной работы.

При самостоятельной проработке материалов темы 4 необходимо использовать:

- учебное пособие 6.1.4;
- презентации № 5 лекционного курса.

При изучении материалов темы 4 необходимо акцентировать внимание на следующих понятиях:

Режимы функционирования вычислительных систем, структура и функции операционных систем. Основные блоки и модули.

- Основные средства аппаратной поддержки функций ОС: система прерываний, защита памяти, механизмы преобразования адресов в системах виртуальной памяти, управление каналами и периферийными устройствами.
- Виды процессов и управления ими в современных ОС.
- Представление процессов, их контексты, иерархии порождения, состояния и взаимодействие.
- Многозадачный (многопрограммный) режим работы.
- Команды управления процессами.
- Средства взаимодействия процессов.
- Модель клиент-сервер и ее реализация в современных ОС.
- Параллельные процессы, схемы порождения и управления.
- Организация взаимодействия между параллельными и асинхронными процессами: обмен сообщениями, организация почтовых ящиков.
- Критические участки, примитивы взаимного исключения процессов, семафоры Дейкстры и их расширения.
- Проблема тупиков при асинхронном выполнении процессов, алгоритмы обнаружения и предотвращения тупиков.
- Операционные средства управления процессами при их реализации на параллельных и распределенных вычислительных системах и сетях: стандарты и программные средства PVM, MPI, OpenMP, POSIX.
- Одноуровневые и многоуровневые дисциплины циклического обслуживания процессов на центральном процессоре, выбор кванта.
- Управление доступом к данным.
- Файловая система, организация, распределение дисковой памяти.
- Управление обменом данными между дисковой и оперативной памятью. Рабочее множество страниц (сегментов) программы, алгоритмы его определения.
- Управление внешними устройствами.
- Оптимизация многозадачной работы компьютеров.
- Операционные системы Windows, Unix, Linux.
- Особенности организации, предоставляемые услуги пользовательского взаимодействия.
- Операционные средства управления сетями.

- Эталонная модель взаимодействия открытых систем ISO/OSI. Маршрутизация и управление потоками данных в сети.
- Локальные и глобальные сети.
- Сетевые ОС, модель “клиент - сервер”, средства управления сетями в ОС UNIX, Windows.
- Семейство протоколов TCP/IP, структура и типы IP – адресов, доменная адресация в Internet.
- Транспортные протоколы TCP, UDP .
- Удаленный доступ к ресурсам сети.
- Организация электронной почты, телеконференций.
- Протоколы передачи файлов FTP и http.
- Язык разметки гипертекста HTML, разработка WEB-страниц, WWW- серверы.

При выполнении практического задания по теме 4 необходимо подготовиться для проведения дискуссии в рамках круглого стола по следующим темам:

1. Оптимизация многозадачной работы компьютеров.
2. Операционные средства управления сетями.
3. Удаленный доступ к ресурсам сети.

Раздел Методы хранения данных и доступа к ним, организация баз данных и знаний.

Знания: Концепция типа данных. Основные структуры данных, алгоритмы обработки и поиска. Основные понятия реляционной и объектной моделей данных. Характеристика современных технологий БД. Язык баз данных SQL. Информационно-поисковые системы. Методы представления знаний. Экспертные системы.

Лекция: №6, **Объем лекций:** 2 часа

На изучение материала данной темы отводится 2 часа лекционных занятий, 4 часа практических занятий и 30 часов самостоятельной работы.

При самостоятельной проработке материалов темы 4 необходимо использовать:

- учебное пособие 6.1.2;
- презентации № 6 лекционного курса.

При изучении материалов темы 5 необходимо акцентировать внимание на следующих понятиях:

- Концепция типа данных.
 - Абстрактные типы данных.
 - Объекты (основные свойства и отличительные признаки).
 - Основные структуры данных, алгоритмы обработки и поиска.
 - Сравнительная характеристика методов хранения и поиска данных.
 - Основные понятия реляционной и объектной моделей данных.
 - Теоретические основы реляционной модели данных (РДМ).
 - Реляционная алгебра, реляционное исчисление.
 - Функциональные зависимости и нормализация отношений.
 - CASE- средства и их использование при проектировании БД.
 - Организация и проектирование физического уровня БД.
 - Методы индексирования. Обобщенная архитектура, состав и функции системы управления базой данных (СУБД).
 - Характеристика современных технологий БД.
 - Примеры соответствующих СУБД.
 - Основные принципы управления транзакциями, журнализацией и восстановлением.
 - Язык баз данных SQL.
 - Средства определения и изменения схемы БД, определения ограничений целостности.
- Контроль доступа.
- Средства манипулирования данными.

- Стандарты языков SQL.
- Интерактивный, встроенный, динамический SQL.
- Основные понятия технологии клиент-сервер. Характеристика SQL-сервера и клиента.
- Сетевое взаимодействие клиента и сервера.
- Информационно-поисковые системы. Классификация.
- Методы реализации и ускорения поиска.
- Методы представления знаний: процедурные представления, логические представления, семантические сети, фреймы, системы продукций.
- Интегрированные методы представления знаний.
- Языки представления знаний.
- Базы знаний.
- Экспертные системы (ЭС).
- Области применения ЭС.
- Архитектура ЭС.
- Механизмы вывода, подсистемы объяснения, общения, приобретения знаний ЭС.
- Жизненный цикл экспертной системы.
- Примеры конкретных ЭС.

При выполнении практического задания по теме 5 необходимо подготовиться для проведения дискуссии в рамках круглого стола по следующим темам:

1. Язык баз данных SQL.
2. Информационно-поисковые системы.
3. Методы представления знаний.
4. Экспертные системы.

Комплексное задание

Представить декларативное знание о понятии «Квартира» двумя моделями представления знаний:

1. в виде семантической сети.
2. в виде продукционной модели.

Квартира состоит из:

1. Кухня.
2. Гостиная.
3. Прихожая.
4. Спальня.
5. Детская.
6. Санитарный узел (туалет).
7. Ванная комната.
8. Кладовка.
9. Гардеробная.
10. Комната отдыха (игровая комната).
11. Спортивная комната (тренажерная).
12. Бытовая комната.

Решение:

Фрагмент семантической сети (рис.)



Пример продукционного правила:

Если есть и «гардеробная» и «спальня» и «кухня» и «прихожая» и «ванная комната» и «санитарный узел» то «квартира двухкомнатная»

Раздел Защита данных и программных систем.

Знания: Аппаратные и программные методы защиты данных и программ. Защита от несанкционированного копирования. Защита от разрушающих программных воздействий. Защита информации в вычислительных сетях.

Лекция: №7, Объем лекций: 2 часа

На изучение материала данной темы отводится 2 часа лекционных занятий, 2 часа практических занятий и 32 часа самостоятельной работы.

При самостоятельной проработке материалов темы 4 необходимо использовать:

- учебное пособие 6.1.3, 6.1.6;
- презентации № 7 лекционного курса.

При изучении материалов темы 6 необходимо акцентировать внимание на следующих понятиях:

- Аппаратные и программные методы защиты данных и программ.
- Защита данных и программ с помощью шифрования.
- Защита от несанкционированного доступа в ОС Windows.
- Система безопасности и разграничения доступа к ресурсам в Windows.
- Файловая система и сервисы Windows.
- Защита от несанкционированного копирования.
- Методы простановки не копируемых меток, настройка устанавливаемой программы на конкретный компьютер, настройка на конфигурацию оборудования.
- Защита от разрушающих программных воздействий.
- Вредоносные программы и их классификация.
- Загрузочные и файловые вирусы, программы-закладки.
- Методы обнаружения и удаления вирусов, восстановления программного обеспечения.
- Защита информации в вычислительных сетях NovellNetware, Windows и других.
- Основы криптографии.
- Задачи обеспечения конфиденциальности и целостности информации. Теоретико-информационный и теоретико-сложностной подходы к определению криптографической стойкости.

- Американский стандарт шифрования DES и российский стандарт шифрования данных ГОСТ 28147-89.
- Системы шифрования с открытым ключом (RSA).
- Цифровая подпись.
- Методы генерации и распределения ключей.

При выполнении практического задания по теме 6 необходимо подготовиться для проведения дискуссии в рамках круглого стола по следующим темам:

1. Аппаратные и программные методы защиты данных и программ.
2. Защита информации в вычислительных сетях.

Комплексное задание

Выполнение алгоритмашифрования с открытым ключом

Шифрование по алгоритму производится следующим образом:

1. Получателем сообщений производится генерация открытого ключа (пара чисел n и e) и закрытого ключа (число d). Для этого:
 - выбираются два простых числа p и q ;
 - определяется первая часть открытого ключа $n := p * q$;
 - определяется вторая часть открытого ключа – выбирается небольшое нечетное число e , взаимно простое с числом $(p-1)*(q-1)$ (заметим, что $(p-1)*(q-1) = p*q*(1-1/p)*(1-1/q) = \varphi(n)$);
 - определяется закрытый ключ: $d := e^{-1} \bmod ((p-1)(q-1))$;

После чего открытый ключ (числа n и e) сообщается всем отправителям сообщений.

2. Отправитель шифрует сообщение (разбивая его, если нужно, на слова S_i длиной менее $\log_2 n$ разрядов): $C_i := (S_i)^e \bmod n$ и отправляет получателю.

3. Получатель расшифровывает сообщение с помощью закрытого ключа d :

$$P_i := (C_i)^d \bmod n.$$

Будем считать верным следующее утверждение:

Шифрование с открытым ключом корректно, то есть в предыдущих обозначениях $P_i = S_i$.

Пример.

Генерация ключей:

1. $p := 3, q := 11$;
2. $n := p * q = 3 * 11 = 33$;
3. $(p-1)*(q-1) = 2 * 10, e := 7$;
4. $d := 7^{-1} \bmod 20 = 3, (7 * 3 \bmod 20 = 1)$.

Пусть $S_1 := 3, S_2 := 1, S_3 := 2$ ($S_1, S_2, S_3 < n = 33$). Тогда код определяется следующим образом.

1. $C_1 := 3^7 \bmod 33 = 2187 \bmod 33 = 9$;
2. $C_2 := 1^7 \bmod 33 = 1 \bmod 33 = 1$;
3. $C_3 := 2^7 \bmod 33 = 128 \bmod 33 = 29$;

При расшифровке имеем:

1. $P_1 := 9^3 \bmod 33 = 729 \bmod 33 = 3$;
2. $P_2 := 1^3 \bmod 33 = 1 \bmod 33 = 1$;
3. $P_3 := 29^3 \bmod 33 = 24389 \bmod 33 = 2$.

Более того, обычно бывает несложно указать алгоритм, который выполняет стоящую перед противником задачу, но не практически, а в принципе.

Вообще говоря, криптосистема с открытым ключом полностью определяется тремя алгоритмами: генерации ключей, шифрования и дешифрования.

Алгоритм генерации ключей G общедоступен; всякий желающий может подать ему на вход случайную строку (K_1, K_2) надлежащей длины и получить пару ключей (K_1, K_2) . Открытый ключ K_1 публикуется, а секретный ключ K_2 и случайная строка r хранятся в секрете. Алгоритмы шифрования E_{K_1} и дешифрования D_{K_2} таковы, что если (K_1, K_2) пара ключей, сгенерированная алгоритмом G , то $D_{K_2}(E_{K_1}(M(m))) = m$ для любого открытого текста m . Для простоты изложения предполагаем, что открытый текст и криптограмма имеют одинаковую длину n . Кроме того, считаем, что открытый текст, криптограмма и оба ключа являются строками в двоичном алфавите.

Предположим теперь, что противник атакует эту криптосистему. Ему известен открытый ключ K_1 , но неизвестен соответствующий секретный ключ K_2 .

Противник перехватил криптограмму d и пытается найти сообщение m , где $d = E_{K_1}(m)$. Поскольку алгоритм шифрования общеизвестен, противник может просто последовательно перебрать все возможные сообщения длины n , вычислить для каждого такого сообщения m_i криптограмму $d_i = E_{K_1}(m_i)$ и сравнить d_i с d . То сообщение, для которого $d_i = d$, и будет искомым открытым текстом. Если повезет, то открытый текст будет найден достаточно быстро. В худшем же случае перебор будет выполнен за время порядка $2^n T(n)$, где $T(n)$ - время, требуемое для вычисления функции E_{K_1} от сообщений длины n . Если сообщения имеют длину порядка 1000 битов, то такой перебор неосуществим на практике ни на каких самых мощных компьютерах.

9. Материально-техническое обеспечение дисциплины

- лекционные аудитории с современными средствами демонстрации 6-415, 6-416, 6-213.

- кафедральные лаборатории, обеспечивающих реализацию ОПОП ВО: 6-218 Учебно-научная лаборатория «Технологии искусственного интеллекта в социально-экономических исследованиях, 6-417 Лаборатория информатики и

программирования, 6-417а Учебно-научная лаборатория «Интеллектуальных технологий проектирования сложных систем».

10. Адаптация рабочей программы для лиц с ОВЗ

При инклюзивном обучении лиц с ОВЗ предоставляется возможность использовать следующие материально-технические средства:

- для студентов с ОВЗ по зрению предусматривается применение средств преобразования визуальной информации в аудио и тактильные сигналы, таких как, брайлевская компьютерная техника, электронные лупы, видеоувеличители, программы невидимого доступа к информации, программы-синтезаторов речи;

- для студентов с ОВЗ по слуху предусматривается применение сурдотехнических средств, таких как, системы беспроводной передачи звука, техники для усиления звука, видеотехника, мультимедийная техника и другие средства передачи информации в доступных формах;

для студентов с нарушениями опорно-двигательной функции предусматривается применение специальной компьютерной техники с соответствующим программным обеспечением, в том числе, специальные возможности операционных систем, таких, как экранная клавиатура и альтернативные устройства ввода информации.

ЛИСТ

согласования рабочей программы

Направление подготовки: **09.06.01 Информатика и вычислительная техника**
код и наименование

Направленность подготовки (программа): **Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей**
наименование

Дисциплина: Модуль: Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Учебный год 2014/2015

РЕКОМЕНДОВАНА заседанием кафедры Вычислительной математики и кибернетики
наименование кафедры

протокол № 15 от "25" августа 2015 г.
Заведующий кафедрой _____ Н. И. Юсупова
подпись расшифровка подписи

Исполнители: проф. _____ О.Н. Сметанина
должность подпись расшифровка подписи

СОГЛАСОВАНО:

Заведующий кафедрой¹
Вычислительной математики
и кибернетики _____ Н. И. Юсупова
наименование кафедры личная подпись расшифровка подписи дата

Председатель НМС по УГСН _____
протокол № 3 от "25" августа 2015 г.
09.06.01 Информатика и вычислительная техника
личная подпись расшифровка подписи

Библиотека, директор _____ С.Ф. Мухоморова 27.08.15
личная подпись расшифровка подписи дата

Начальник отдела аспирантуры _____ Мухоморов РМ 28.09.15
личная подпись расшифровка подписи дата

Рабочая программа зарегистрирована в ООПМА и внесена в электронную базу
данных

Начальник _____ Лок. МАННА 31.08.15
личная подпись расшифровка подписи дата

¹ Согласование осуществляется с выпускающими кафедрами (для рабочих программ, подготовленных на кафедрах, обеспечивающих подготовку для других направлений и специальностей)