

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра теоретической информатики

УТВЕРЖДАЮ

Декан факультета информатики и вычислительной техники

 Д.Ю.Чальи
(подпись)

« 18 » мая 20 21 г.

Рабочая программа дисциплины

«Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»

Направление подготовки

09.06.01 Информатика и вычислительная техника

Направленность (профиль)

«Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»

Форма обучения очная

Программа рассмотрена
на заседании кафедры теоретической информатики

от « 27 » апреля 20 21 года, протокол № 9

Зав.кафедрой _____ В.А.Соколов

Ярославль

1. Цели освоения дисциплины

Целью освоения дисциплины «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» является фундаментальная подготовка аспиранта по одноимённой специальности для повышения его общенаучного уровня, создания необходимой теоретической базы для выполнения диссертационных исследований и сдачи выпускного кандидатского экзамена по специальности, а также получение обучающимися теоретических знаний о средствах и методах программирования с последующим применением в профессиональной сфере и формирование практических навыков по разработке программных систем, способности решать задачи комплексной разработки ПО, используя современные инструментальные средства.

2. Место дисциплины в структуре программы аспирантуры

Учебная дисциплина «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» реализуется в вариативной части основной профессиональной образовательной программы высшего образования - программы подготовки научно-педагогических кадров в аспирантуре по направлению подготовки 09.06.01 «Информатика и вычислительная техника», профилю подготовки «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» очной формы обучения.

Изучение учебной дисциплины «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» базируется на знаниях и умениях, полученных обучающимися ранее в ходе освоения программного материала при изучении программ магистратуры.

3. Планируемые результаты обучения по дисциплине – знания, умения, навыки и (или) опыт деятельности, характеризующие этапы формирования компетенций и обеспечивающие достижение планируемых результатов освоения программы аспирантуры, и критерии их оценивания

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих универсальных, общепрофессиональных и профессиональных компетенций:

ОПК-3, ПК-1, ПК-2, ПК-3, ПК-4, ПК-6 в соответствии с основной профессиональной образовательной программой высшего образования - программой подготовки научно-педагогических кадров в аспирантуре по направлению подготовки 09.06.01

«Информатика и вычислительная техника», профилю подготовки «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

В результате освоения учебной дисциплины обучающийся должен демонстрировать следующие результаты:

Профессиональные компетенции:

Код компетенции	Планируемые результаты обучения	Критерии оценивания результатов обучения		
		Пороговый уровень	Продвинутый уровень	Высокий уровень
ОПК-3	Способность к разработке новых методов исследования и их применению в самостоятельной научно-исследовательской деятельности в области профессиональной	В целом успешное, но не систематическое умение разрабатывать новые методы исследования и их	В целом успешное, но сопровождающееся отдельными недостатками умение разрабатывать	Успешное и систематическое умение разрабатывать новые методы исследования и их применения в

	деятельности	применения в самостоятельной научно-исследовательской деятельности в области профессиональной деятельности.	новые методы исследования и их применения в самостоятельной научно-исследовательской деятельности в области профессиональной деятельности.	самостоятельной научно-исследовательской деятельности в области профессиональной деятельности.
ПК-1	Способность выполнять теоретические исследования процессов создания, накопления и обработки информации, включая анализ и создание моделей данных и знаний, языков их описания и манипулирования, разработку новых математических методов и средств поддержки интеллектуальной обработки данных	<ol style="list-style-type: none"> 1. Знание современных основ программирования. 2. Знание основных принципов организации вычислительных машин, систем и сетей. 3. Знание современных языков и систем программирования. 4. Владение основами организации операционных систем. 	<ol style="list-style-type: none"> 1. Знание математических основ программирования. 2. Знание основных принципов организации современных вычислительных машин, систем и сетей. 3. Знание технологий разработки программного обеспечения. 4. Знание принципов организации операционных систем. 5. Знание методов хранения данных и доступа к ним. организация баз данных и знаний. 6. Знание аппарата защиты данных и программных систем. 	<ol style="list-style-type: none"> 1. Знание и владение математическими основами программирования. 2. Знание основных принципов разработки и эксплуатации вычислительных машин, систем и сетей. 3. Знание языков, систем программирования и технологий разработки программного обеспечения. 4. Знание принципов организации современных операционных систем. 5. Владение методами хранения данных и доступа к ним. Знание современной организация баз данных и знаний. 6. Владение аппаратом защиты данных и программных систем.
ПК-2	Способность разрабатывать новые математические модели объектов и явлений, развивать аналитические и приближенные методы их исследования, выполнять реализацию эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения	В целом успешное, но не систематическое умение разрабатывать новые математические модели объектов и явлений, развивать аналитические и приближенные методы их	В целом успешное, но сопровождающееся отдельными недостатками умение разрабатывать новые математические модели объектов и явлений, развивать	Успешное и систематическое умение разрабатывать новые математические модели объектов и явлений, развивать аналитические и приближенные методы их

	вычислительного эксперимента	исследования, выполнять реализацию эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента.	аналитические и приближенные методы их исследования, выполнять реализацию эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента.	исследования, выполнять реализацию эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента.
ПК-3	Способность разрабатывать новые алгоритмы, уметь анализировать трудоемкость алгоритмов и их потребность в памяти, использовать различные языки программирования для разработки программ, уметь их тестировать, оценивать качество с учетом стандартов	В целом успешное, но не систематическое умение разрабатывать новые алгоритмы, уметь анализировать трудоемкость алгоритмов и их потребность в памяти, использовать различные языки программирования для разработки программ, уметь их тестировать, оценивать качество с учетом стандартов.	В целом успешное, но сопровождающееся отдельными недостатками умение разрабатывать новые алгоритмы, уметь анализировать трудоемкость алгоритмов и их потребность в памяти, использовать различные языки программирования для разработки программ, уметь их тестировать, оценивать качество с учетом стандартов.	Успешное и систематическое умение разрабатывать новые алгоритмы, уметь анализировать трудоемкость алгоритмов и их потребность в памяти, использовать различные языки программирования для разработки программ, уметь их тестировать, оценивать качество с учетом стандартов.
ПК-4	Способность объективно оценивать профессиональный уровень результатов научных исследований, в том числе с помощью международных баз данных публикационной активности	В целом успешное, но не систематическое умение объективно оценивать профессиональный уровень результатов научных исследований, в том числе с помощью международных баз данных публикационной активности.	В целом успешное, но сопровождающееся отдельными недостатками умение объективно оценивать профессиональный уровень результатов научных исследований, в том числе с помощью международных баз данных публикационной активности.	Успешное и систематическое умение объективно оценивать профессиональный уровень результатов научных исследований, в том числе с помощью международных баз данных публикационной активности.
ПК-6	Способность оформлять результаты своей научно-исследовательской деятельности в форме научно-квалификационной работы по научной специальности 05.13.11 Математическое и программное обеспечение	В целом успешное, но не систематическое умение оформлять результаты своей научно-исследовательской деятельности в	В целом успешное, но сопровождающееся отдельными недостатками умение оформлять результаты своей научно-	Успешное и систематическое умение оформлять результаты своей научно-

вычислительных машин, комплексов и компьютерных сетей	форме научно-квалификационной работы по научной специальности 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.	исследовательской деятельности в форме научно-квалификационной работы по научной специальности 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.	квалификационной работы по научной специальности 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.
---	--	---	---

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 акад. часов

Дисциплина изучается в течение четырех семестров (2, 3, 4, 5). Формой итоговой промежуточной аттестации по дисциплине в последнем семестре ее изучения является кандидатский экзамен.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий и их трудоемкость (в академических часах)					Формы текущего контроля успеваемости	
			лекции	практические	лабораторные	консультации	самостоятельная работа	Форма промежуточной аттестации (по семестрам)	
1.	Математические основы программирования	2	4			1	32		
2.	Вычислительные машины, системы и сети	2	4			1	30	Зачёт	
3.	Языки и системы программирования. Технология разработки программного обеспечения	3	6				30	Тест	
4.	Операционные системы	4	6			2	28	Зачёт	
5.	Методы хранения данных и доступа к ним, организация баз данных и знаний	5	4			1	40		
6.	Защита данных и программных систем	5	2			1	24	Экзамен	
Всего			26			6	184	216	

Содержание разделов дисциплины

1. Математические основы программирования

1. Понятие алгоритма и его уточнения: машины Тьюринга, нормальные алгоритмы Маркова, рекурсивные функции. Эквивалентность данных формальных моделей алгоритмов. Понятие об алгоритмической неразрешимости. Примеры алгоритмически неразрешимых проблем.
2. 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 приближённые комбинаторные алгоритмы.
3. 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).
4. Автоматы. Алгебры регулярных выражений. Теорема Клини о регулярных языках.
5. Алгебра логики. Булевы функции, канонические формы задания булевых функций. Понятие полной системы. Критерий полноты Поста.
6. Исчисление предикатов первого порядка. Понятие интерпретации. Выполнимость и общезначимость формулы первого порядка. Понятие модели. Теорема о полноте исчисления предикатов первого порядка.
7. Отношения и функции. Отношение эквивалентности и разбиения. Фактор множество. Отношения частичного порядка. Теоретико-множественное и алгебраическое определения решётки, их эквивалентность. Свойства решёток. Булевы решётки. Полные решётки.
8. Формальные языки и способы их описания. Классификация формальных грамматик. Их использование в лексическом и синтаксическом анализе.
9. Классы рекурсивных функций. Связь рекурсивных функций и машин Тьюринга.
10. Основы комбинаторного анализа. Метод производящих функций, метод включений и исключений. Примеры применения.
11. Коды с исправлением ошибок. Алфавитное кодирование. Методы сжатия информации.
12. Основы криптографии. Задачи обеспечения конфиденциальности и целостности информации. Теоретико-информационный и теоретико-сложностный 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.

Вычислительные машины, системы и сети

1. 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.
2. Классификация вычислительных систем (ВС) по способу организации параллельной обработки. Многопроцессорные и многомашинные комплексы. Вычислительные кластеры. Проблемно-ориентированные параллельные структуры: матричные ВС, систолические структуры, нейросети.
3. Назначение, архитектура и принципы построения информационно-вычислительных сетей (ИВС). Локальные и глобальные ИВС, технические и программные средства объединения различных сетей.
4. Методы и средства передачи данных в ИВС, протоколы передачи данных.
5. Особенности архитектуры локальных сетей (Ethernet, Token Ring, FDDI).
6. Сеть Internet, доменная организация, семейство протоколов TCP/IP. Информационно-вычислительные сети и распределенная обработка информации.

Языки и системы программирования. Технология разработки программного обеспечения

1. Языки программирования. Процедурные языки программирования (Фортран, Си), Функциональные языки программирования (Лисп), логическое программирование (Пролог), объектно-ориентированные языки программирования (Ява).
2. 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.
3. Объектно-ориентированное программирование. Классы и объекты, наследование, интерфейсы. Понятие об объектном окружении. Рефлексия. Библиотеки классов. Средства обработки объектов (контейнеры и итераторы).
4. 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.
5. 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.
6. 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 (LL(1)-грамматики) и восходящие (LR(1)-грамматики) 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.
7. 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.
8. Генерация объектного кода в компиляторах. Перенастраиваемые (retargetable) компиляторы, gcc (набор компиляторов Gnu). Переработка термов (term rewriting). Применение оптимизационных эвристик (целочисленное программирование, динамическое программирование) для автоматической генерации генераторов объектного кода (системы BEG, Iburg и др.).
9. Машинно-ориентированные языки, язык ассемблера. Представление машинных команд и констант. Команды транслятору. Их типы, принципы реализации. Макросредства, макровыводы, языки макроопределений, условная макрогенерация, принципы реализации.
10. Системы программирования, типовые компоненты СП: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы. Модульное программирование. Типы модулей. Связывание модулей по управлению и данным.
11. Пакеты прикладных программ (ППП). Системная часть и наполнение. Языки общения с ППП. Машинная графика. Средства поддержки машинной графики. Графические пакеты.
12. 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.

13. 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.

14. Методы спецификации программ. Методы проверки спецификации. Схемное, структурное, визуальное программирование. Разработка пользовательского интерфейса, стандарт CUA, мультимедийные среды интерфейсного взаимодействия.

Операционные системы

1. 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,

2. 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 её реализация в современных ОС.

3. Параллельные процессы, схемы порождения и управления. Организация взаимодействия между параллельными и асинхронными процессами: обмен сообщениями, организация почтовых ящиков. Критические участки, примитивы взаимоисключения процессов, семафоры Дейкстры и их расширения. Проблема тупиков при асинхронном выполнении процессов, алгоритмы обнаружения и предотвращения тупиков.

4. Операционные средства управления процессами при их реализации на параллельных и распределенных вычислительных системах и сетях: стандарты и программные средства PVM, MPI, OpenMP, POSIX .

5. Одноуровневые и многоуровневые дисциплины циклического обслуживания процессов на центральном процессоре, выбор кванта.

6. 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.

7. Управление внешними устройствами.

8. Оптимизация многозадачной работы компьютеров. Операционные системы Windows, Unix, Linux. Особенности организации, предоставляемые услуги пользовательского взаимодействия.

9. 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 NT. Семейство протоколов TCP/IP , структура и типы IP – адресов, доменная адресация в Internet .Транспортные протоколы TCP, UDP .

10. Удаленный доступ к ресурсам сети. Организация электронной почты, телеконференций. Протоколы передачи файлов FTP и HTTP, язык разметки гипертекста HTML, разработка WEB-страниц, WWW- серверы.

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

1. Концепция типа данных. Абстрактные типы данных. Объекты (основные свойства и отличительные признаки).

2. Основные структуры данных, алгоритмы обработки и поиска. Сравнительная характеристика методов хранения и поиска данных.

3. Основные понятия реляционной и объектной моделей данных.

4. Теоретические основы реляционной модели данных (РДМ). Реляционная алгебра, реляционное исчисление. Функциональные зависимости и нормализация отношений.

5. CASE- средства и их использование при проектировании БД.

6. Организация и проектирование физического уровня БД. Методы индексирования.

7. Обобщенная архитектура, состав и функции системы управления базой данных (СУБД). Характеристика современных технологий БД. Примеры соответствующих СУБД.
8. Основные принципы управления транзакциями, журнализацией и восстановлением.
9. Язык баз данных SQL. Средства определения и изменения схемы БД, определения ограничений целостности. Контроль доступа. Средства манипулирования данными.
10. Стандарты языков SQL. Интерактивный, встроенный, динамический SQL.
11. Основные понятия технологии клиент-сервер. Характеристика SQL-сервера и клиента. Сетевое взаимодействие клиента и сервера.
12. Информационно-поисковые системы. Классификация. Методы реализации и ускорения поиска.
13. Методы представления знаний: процедурные представления, логические представления, семантические сети, фреймы, системы продукций. Интегрированные методы представления знаний. Языки представления знаний. Базы знаний.
14. Экспертные системы (ЭС). Области применения ЭС. Архитектура ЭС. Механизмы вывода, подсистемы объяснения, общения, приобретения знаний ЭС. Жизненный цикл экспертной системы. Примеры конкретных ЭС.

Защита данных и программных систем

1. Аппаратные и программные методы защиты данных и программ. Защита данных и программ с помощью шифрования.
2. Защита от несанкционированного доступа в ОС Windows NT. Система безопасности и разграничения доступа к ресурсам в Windows NT. Файловая система NTFS и сервисы Windows NT.
3. Защита от несанкционированного копирования. Методы простановки не копируемых меток, настройка устанавливаемой программы на конкретный компьютер, настройка на конфигурацию оборудования.
4. Защита от разрушающих программных воздействий. Вредоносные программы и их классификация. Загрузочные и файловые вирусы, программы-закладки. Методы обнаружения и удаления вирусов, восстановления программного обеспечения.
5. Защита информации в вычислительных сетях Novell Netware, Windows NT и других.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

Освоение учебной дисциплины «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» предусматривает использование в учебном процессе активных и интерактивных форм проведения учебных занятий в форме: консультация, реферат, лекция-дискуссия в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

В рамках учебной дисциплины «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» предусмотрены встречи с руководителями и работниками организаций, деятельность которых связана с направленностью (профилем) реализуемой основной профессиональной образовательной программы.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости).

- } операционная система – Windows 8;
- } программное обеспечение Microsoft Office;
- } программное обеспечение C++, Java, C#.

7. Перечень основной и дополнительной учебной литературы, необходимых для освоения дисциплины

а) основная литература

1. [Тель, Ж., Введение в распределенные алгоритмы / Ж. Тель; пер. с англ. В. А. Захарова, М., МЦНМО, 2009, 616с](#)

б) дополнительная литература

3. Дейт К.Дж. Введение в системы баз данных. – М.: Вильямс, 1999.

4. Дейтел Г. Введение в операционные системы. М. Мир. 1987.

5. Кнут Д. Искусство программирования, т. 1 – 3. ИД «Вильямс» М., СПб., Киев 2000.

6. Когаловский М.Р. Энциклопедия технологий баз данных. – М.: "Финансы и статистика", 2002.

9. Котов В.Е., Сабельфельд В.К. Теория схем программ. М.Наука. 1991.

10. **Зыков С.В. Введение в теорию программирования. Объектно-ориентированный подход. - М.: Национальный Открытый Университет "ИНТУИТ", 2004. - 391 с.**

2. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для изучения учебной дисциплины «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» в рамках реализации основной профессиональной образовательной программы по направлению подготовки 09.06.01 «Информатика и вычислительная техника» используются:

Учебная аудитория для занятий лекционного типа оснащена специализированной мебелью (стол для преподавателя, парты, стулья, доска для написания мелом); техническими средствами обучения (видеопроекторное оборудование, средства звуковоспроизведения, экран и имеющие выход в сеть Интернет).

Учебная аудитория для занятий семинарского типа: оснащена специализированной мебелью (стол для преподавателя, парты, стулья, доска для написания мелом); техническими средствами обучения (видеопроекторное оборудование, средства звуковоспроизведения, экран и имеющие выход в сеть Интернет).

Автор:

Зав. кафедрой теоретической информатики,
профессор, д.ф.-м.н. Соколов В.А.

**Приложение к №1 рабочей программе дисциплины
«Математическое и программное обеспечение вычислительных машин, комплексов
и компьютерных сетей»**

**Оценочные средства
для проведения текущей и/или промежуточной аттестации аспирантов
по дисциплине**

Форма текущего контроля успеваемости и промежуточной аттестации обучающегося по учебной дисциплине.

Структурно-тематический план контроля уровня освоенности компетенций в части, предусмотренной рабочей программой дисциплины.

Структура дисциплины	Вид контроля	Номер оценочного средства
Текущий контроль		
Тема 1. Системы программирования	Зачёт	1
Тема 2.1. Технология разработки программного обеспечения	Зачёт	2
Тема 2.2. Технология разработки программного обеспечения	Тестовые задания	3
Промежуточный контроль		
Промежуточная аттестация	Экзамен (кандидатский)	4

**1. Типовые контрольные задания или иные материалы,
необходимые для оценки знаний, умений, навыков и (или) опыта деятельности,
характеризующих этапы формирования компетенций**

1.1 Список вопросов и (или) заданий для проведения промежуточной аттестации

Список вопросов к экзамену:

2. Процедурные языки программирования (Фортран, Си),
3. Функциональные языки программирования (Лисп), логическое программирование (Пролог), объектно-ориентированные языки программирования (Ява).
4. Основные управляющие конструкции, структура программы.
5. Работа с данными: переменные и константы, типы данных (булевский, целочисленные, плавающие, символьные, типы диапазона и перечисления, указатели), структуры данных (массивы и записи).
6. Процедуры (функции): вызов процедур, передача параметров (по ссылке, по значению, по результату), локализация переменных, побочные эффекты.
7. Обработка исключительных ситуаций. Библиотеки процедур и их использование.
8. Объектно-ориентированное программирование. Классы и объекты, наследование, интерфейсы.
9. Понятие об объектном окружении. Рефлексия. Библиотеки классов. Средства обработки объектов (контейнеры и итераторы).
10. Распределенное программирование. Процессы и их синхронизация. Семафоры, мониторы Хоара.
11. Объектно-ориентированное распределенное программирование. CORBA. Параллельное программирование над общей памятью. Нити. Стандартный интерфейс Open MP.
12. Распараллеливание последовательных программ. Параллельное программирование над распределенной памятью.
13. Парадигмы SPMD и MIMD. Стандартный интерфейс MPI.
14. Основы построения трансляторов. Структура оптимизирующего транслятора.
15. Промежуточные представления программы: последовательность символов, последовательность лексем, синтаксическое дерево, абстрактное синтаксическое дерево.
16. Уровни промежуточного представления: высокий, средний, низкий. Формы промежуточного представления.
17. Анализ исходной программы в компиляторе. Автоматные (регулярные) грамматики, контекстно-свободные грамматики и синтаксический анализ, организация таблицы символов программы, имеющей блочную структуру, хеш-функции.
18. Нисходящие и восходящие методы синтаксического анализа.
19. Семантические программы, построение абстрактного синтаксического дерева.
20. Автоматическое построение лексических и синтаксических анализаторов по формальным описаниям грамматик.
21. Оптимизация программ при их компиляции. Оптимизация базовых блоков, чистка циклов.
22. Анализ графов потока управления и потока данных. Отношение доминирования и его свойства, построение границы области доминирования

вершины, выделение сильно связанных компонент графа.

23. Построение графа зависимостей. Перевод программы в SSA-представление и обратно. Глобальная и межпроцедурная оптимизация.

24. Генерация объектного кода в компиляторах.

25. Машинно-ориентированные языки, язык ассемблера. Представление машинных команд и констант.

26. Команды транслятору. Их типы, принципы реализации. Макросредства, макровыводы, языки макроопределений, условная макрогенерация, принципы реализации.

27. Системы программирования (СП), типовые компоненты СП: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы.

28. Модульное программирование. Типы модулей. Связывание модулей по управлению и данным.

29. Пакеты прикладных программ (ППП). Системная часть и наполнение. Языки общения с ППП. Машинная графика.

30. Средства поддержки машинной графики. Графические пакеты.

31. Проблемы разработки программного обеспечения

32. Концепция программного средства как изделия, имеющего самостоятельное значение

33. Понятие жизненного цикла программного обеспечения

34. Процесс определения требований к ПО

35. Разработка целей создания программного обеспечения

36. Документирование требований. Техническое задание

37. Спецификации и их роль в разработке программ

38. Р- технология

39. Метод структурного анализа

40. Диаграммы потоков данных

41. Способы задания спецификаций процессов

42. Расширения реального времени в диаграммах потоков данных

43. Диаграммы переходов-состояний

44. Диаграммы сущность-связь

45. Основы структурного проектирования

46. Связность, целостность и другие характеристики модулей

47. Типовые модульные структуры

48. Восходящее и нисходящее проектирование

49. Общие правила структурного построения программного обеспечения

50. Правила связи программных модулей по управлению

51. Правила связи программных модулей по информации

52. Типовая структура программного модуля

53. Общая схема отладки ПО

54. Методы диагностики и локализации ошибок

55. Принципы и методы тестирования ПО

56. Принципы тестирования ПО

57. Классификация методов тестирования ПО

58. Методы статического тестирования ПО

59. Методы детерминированного динамического тестирования ПО

60. Тестирование модулей ПО

61. Испытания программного обеспечения

62. Критерии завершения тестирования ПО

Критерии и шкала оценивания экзамена в п.3.3. Приложения 1

1.2 Контрольные задания и иные материалы, используемые в процессе текущей аттестации

Контрольным мероприятием промежуточной аттестации обучающихся по учебной дисциплине является экзамен (кандидатский), который проводится в письменной форме.

Примерный перечень контрольных вопросов к текущему контролю успеваемости и промежуточной аттестации.

. Системы программирования

1. История развития языков программирования.
2. Логические языки программирования и искусственный интеллект.
3. Функциональные языки программирования.
4. Современные парадигмы программирования.
5. Объектно-ориентированное программирование.
6. Непроцедурные системы программирования.
7. Параллельное программирование.
8. Системное программирование. Язык Ассемблера.
9. Языки программирования в интернете.

Технология разработки программного обеспечения

1. Сравнительный анализ моделей ЖЦПО с точки зрения их применимости в методологиях RUP и XP.
2. Анализ опыта применения языка UML как инструмента бизнес-моделирования.
3. Сравнительный анализ жестких и гибких методологий разработки программ - XP, Scrum, RAD, RUP - с точки зрения автоматизации менеджмента проекта.
4. Теория и практика сопровождения ПО - анализ основных проблем
5. Сравнительный анализ и особенности применения моделей качества ISO, TQM, CMM, SPICE.
6. Метод бригады главного программиста и ролевое разделение работ в проекции на типовые методологии разработки программ.
7. Методы обеспечения надежности в распределенных системах
8. Безопасность информации в среде Интернет.
9. Методы и средства разработки систем офисной автоматизации.
10. Архитектурные решения и реинжиниринг в корпоративных системах.
11. Методы и инструменты визуализации информации при построении информационных систем.

Тесты для самопроверки

К формированию компетенции ОПК-3

1. Какие программы можно отнести к системному программному обеспечению:

операционные системы;

- 1) прикладные программы;
- 2) игровые программы.

2. Какие программы можно отнести к системному ПО:

драйверы;

- 1) текстовые редакторы;
- 2) электронные таблицы;
- 3) графические редакторы.

3. Специфические особенности ПО как продукта:

продажа по ценам ниже себестоимости (лицензирование);

- 1) низкие материальные затраты при создании программ;
- 2) возможность создание программ небольшие коллективом или даже одним человеком;
- 3) разнообразие решаемых задач с помощью программных средств.

4. Какие программы можно отнести к системному ПО:

- 1) программа расчета заработной платы;
- 2) электронные таблицы;

СУБД (системы управления базами данных).

5. Какие программы нельзя отнести к системному ПО:

игровые программы;

- 1) компиляторы языков программирования;
- 2) операционные системы;
- 3) системы управления базами данных.

6. Какие программы можно отнести к прикладному программному обеспечению:

электронные таблицы;

- 1) таблицы решений;
- 2) СУБД (системы управления базами данных).

7. Какие программы можно отнести к прикладному ПО:

программа расчета заработной платы;

- 1) диспетчер программ;
- 2) программа «Проводник» (Explorer).

8. Какие программы нельзя отнести к прикладному ПО:

компиляторы и (или) интерпретаторы;

- 1) текстовые и (или) графические редакторы;
- 2) электронные таблицы.

9. Можно ли отнести операционную систему к программному обеспечению:
да;

- 1) нет.

10. Можно ли отнести операционную систему к прикладному программному обеспечению:

- 1) да;

нет.

11. Специфические особенности ПО как продукта:

низкие затраты при дублировании;

- 1) универсальность;
- 2) простота эксплуатации;
- 3) наличие поддержки (сопровождения) со стороны разработчика.

12. Какие программы можно отнести к системному ПО:

утилиты;

- 1) экономические программы;
- 2) статистические программы;
- 3) мультимедийные программы.

13. Этап, занимающий наибольшее время, в жизненном цикле программы:

сопровождение;

- 1) проектирование;
- 2) тестирование;
- 3) программирование;
- 4) формулировка требований.

14. Этап, занимающий наибольшее время, при разработке программы:

тестирование;

- 1) сопровождение;
- 2) проектирование;
- 3) программирование;
- 4) формулировка требований.

15. Первый этап в жизненном цикле программы:

формулирование требований;

- 1) анализ требований;
- 2) проектирование;
- 3) автономное тестирование;
- 4) комплексное тестирование.

16. Один из необязательных этапов жизненного цикла программы:

оптимизация;

- 1) проектирование;
- 2) тестирование;
- 3) программирование;
- 4) анализ требований.

17. Самый большой этап в жизненном цикле программы:

эксплуатация;

- 1) изучение предметной области;
- 2) программирование;
- 3) тестирование;
- 4) корректировка ошибок.

18. Какой этап выполняется раньше:

отладка;

- 1) тестирование.

19. Какой этап выполняется раньше:

- 1) отладка;
- 2) оптимизация;

программирование;

- 3) тестирование.

20. Что выполняется раньше:

компиляция;

- 1) отладка;
- 2) компоновка;
- 3) тестирование.

21. Что выполняется раньше:

проектирование;

- 1) программирование;
- 2) отладка;
- 3) тестирование.

22. В стадии разработки программы не входит:

автоматизация программирования;

- 1) постановка задачи;
- 2) составление спецификаций;
- 3) эскизный проект;
- 4) тестирование.

23. Самый важный критерий качества программы:

работоспособность;

- 1) надежность;
- 2) эффективность;
- 3) быстрое действие;
- 4) простота эксплуатации.

24. Способы оценки качества:
сравнение с аналогами;
1) наличие документации;
2) оптимизация программы;
3) структурирование алгоритма.
25. Существует ли связь между эффективностью и оптимизацией программы:
да;
1) нет.
26. Наиболее важный критерий качества:
надежность;
1) быстроедействие;
2) удобство в эксплуатации;
3) удобный интерфейс;
4) эффективность.
27. Способы оценки надежности:
тестирование;
1) сравнение с аналогами;
2) трассировка;
3) оптимизация.
28. Повышает ли качество программ оптимизация:
да;
1) нет.
29. Существует ли связь между надежностью и быстрымдействием:
нет;
1) да.
30. В каких единицах можно измерить надежность:
отказов/час;
1) км/час;
2) Кбайт/сек;
3) операций/сек.
31. В каких единицах можно измерить быстроедействие:
1) отказов/час;
2) км/час;
3) Кбайт/сек;
операций/сек.
32. Что относится к этапу программирования:
написание кода программы;
1) разработка интерфейса;
2) работоспособность;
3) анализ требований.
33. Последовательность этапов программирования:
компилирование, компоновка, отладка;
1) компоновка, отладка, компилирование;
2) отладка, компилирование, компоновка;
3) компилирование, отладка, компоновка.
34. Инструментальные средства программирования:
компиляторы, интерпретаторы;
1) СУБД (системы управления базами данных);
2) BIOS (базовая система ввода-вывода);
3) ОС (операционные системы).
35. На языке программирования составляется:

исходный код;

- 1) исполняемый код;
- 2) объектный код;

алгоритм.

36. Правила, которым должна следовать программа это:

алгоритм;

- 1) структура;
- 2) спецификация;
- 3) состав информации.

37. Можно ли внутри цикла поместить еще один цикл:

да;

- 1) нет.

38. Можно ли внутри условного оператора поместить еще одно условие:

да;

- 1) нет.

39. Доступ, при котором записи файла читаются в физической последовательности, называется:

последовательным;

- 1) прямым;
- 2) простым;
- 3) основным.

40. Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:

прямым;

- 1) последовательным;
- 2) простым;
- 3) основным.

41. Методы программирования (укажите НЕ верный ответ):

логическое;

- 1) структурное;
- 2) модульное.

42. Что выполняется раньше:

разработка алгоритма;

- 1) выбор языка программирования;
- 2) написание исходного кода;
- 3) компиляция.

43. Можно ли переменным присваивать произвольные идентификаторы:

да;

- 1) нет.

44. Какие символы не допускаются в именах переменных:

пробелы;

- 1) цифры;

К формированию компетенции ПК-1

45. Какой методикой проектирования пользуются при структурном программировании:

сверху вниз;

- 1) снизу-вверх.

46. Какой этап проектирования может быть исключен:

эскизный проект;

- 1) технический проект;
- 2) рабочий проект.

47. Какие этапы проектирования можно объединять:

технический и рабочий;

- 1) эскизный и рабочий;
- 2) технический и эскизный.

48. Модульное программирование применимо при:

- 1) проектировании сверху вниз;

проектирование снизу-вверх;

49. Процесс преобразования постановки задачи в план алгоритмического или вычислительного решения это:

проектирование;

- 1) анализ требований;
- 2) программирование;
- 3) тестирование.

50. Составление спецификаций это:

формализация задачи;

- 1) эскизный проект;
- 2) поиск алгоритма;
- 3) отладка.

51. Что определяет выбор языка программирования:

область приложения;

- 1) знание языка;
- 2) наличие дополнительных библиотек.

52. Возможно ли комбинирование языков программирования в рамках одной задачи:

да;

- 1) нет.

53. Допустимо ли комбинирование языков программирования в рамках одной задачи:

да;

- 1) нет.

54. Этап разработки программы, на котором дается характеристика области применения программы:

техническое задание;

- 1) эскизный проект;
- 2) технический проект;
- 3) внедрение;
- 4) рабочий проект.

55. Укажите правильную последовательность создания программы:

Формулирование задачи, анализ требований, проектирование,

программирование;

- 1) анализ требований, проектирование, программирование, тестирование, отладка;
- 2) анализ требований, программирование, проектирование, тестирование;
- 3) анализ требований, проектирование, программирование, модификация, трассировка;
- 4) формулирование задачи, анализ требований, проектирование, программирование,

проектирование, отладка.

56. На каком этапе производится выбор языка программирования:

проектирование;

- 1) программирование;
- 2) отладка;
- 3) тестирование.

57. Нисходящее проектирование это:
последовательное уточнение (детализация);
1) составление блок-схем;
2) разделение программы на отдельные участки (блоки);
3) трассировка.
58. Какой методикой пользуются при структурном программировании:
сверху вниз;
1) снизу-вверх.
59. Проектирование сверху вниз это:
последовательное разбиение общих задач на более мелкие;
1) составление из отдельных модулей большой программы.
60. Проектирование снизу-вверх это:
составление из отдельных модулей большой программы;
1) последовательное разбиение общих задач на более мелкие.
61. Модульное программирование применимо при:
1) проектировании сверху вниз;
2) проектировании снизу-вверх;
и в том, и другом случае;
3) ни в коем случае.
62. Какой методикой проектирования пользуются при структурном программировании:
сверху вниз;
1) снизу-вверх.
63. Какой метод проектирования соответствует иерархическому подходу в решении задачи:
нисходящее (сверху вниз);
1) восходящее (снизу-вверх).
64. В каких единицах измеряются затраты на проектирование:
в человеко-днях;
1) в долларах;
2) в тенге;
3) в килобайтах.
65. Несуществующий метод проектирования:
алгоритмическое;
1) нисходящее;
2) восходящее.
66. Метод проектирования:
нисходящее;
1) алгоритмическое;
2) логическое;
3) использование языков программирования;
4) составление блок-схем.
67. Нисходящее проектирование это:
последовательное уточнение (детализация);
1) составление блок-схем;
2) разделение программы на отдельные участки (блоки);
3) трассировка.
68. Признаки нисходящего программирования:
последовательная детализация;
1) наличие оптимизации;
2) наличие тестирования;
3) автоматизация программирования.

К формированию компетенции ПК-2

69. Какой методикой пользуются при структурном программировании:
сверху вниз;

1) снизу-вверх.

70. Этап разработки программы, на котором дается характеристика области применения программы:

техническое задание;

1) эскизный проект;

2) технический проект;

3) внедрение;

4) рабочий проект.

71. Процедура поиска ошибки, когда известно, что она есть это:

отладка;

1) тестирование;

2) компоновка;

3) транзакция;

4) трансляция.

72. Программа для просмотра значений переменных при выполнении программы:

отладчик;

1) компилятор;

2) интерпретатор;

3) трассировка;

4) тестирование.

73. Отладка – это:

процедура поиска ошибок, когда известно, что ошибка есть;

1) определение списка параметров;

2) правило вызова процедур (функций);

3) составление блок-схемы алгоритма.

74. Когда программист может проследить последовательность выполнения команд программы:

при трассировке;

1) при тестировании;

2) при компиляции;

3) при выполнении программы;

4) при компоновке.

75. На каком этапе создания программы могут появиться синтаксические ошибки:

программирование;

1) проектирование;

2) анализ требований;

3) тестирование.

76. Когда приступают к тестированию программы:

когда программа уже закончена;

1) после постановки задачи;

2) на этапе программирования;

3) на этапе проектирования;

4) после составления спецификаций,

77. Тестирование бывает:

автономное;

1) инструментальное;

- 2) визуальное;
3) алгоритмическое.
78. Тестирование бывает:
комплексное;
1) инструментальное;
2) визуальное;
3) алгоритмическое.
79. Существует ли различие между отладкой и тестированием:
да;
1) нет.
80. При комплексном тестировании проверяются:
согласованность работы отдельных частей программы;
1) правильность работы отдельных частей программы;
2) быстродействие программы;
3) эффективность программы.
81. Чему нужно уделять больше времени, чтобы получить хорошую программу:
тестированию;
1) программированию;
2) отладке;
3) проектированию.
82. Процесс исполнения программы с целью обнаружения ошибок:
тестирование;
1) кодирование;
2) сопровождение;
3) проектирование.
83. Автономное тестирование это:
тестирование отдельных частей программы;
1) инструментальное средство отладки;
2) составление блок-схем;
3) пошаговая проверка выполнения программы.
84. Трассировка это:
проверка пошагового выполнения программы;
1) тестирование исходного кода;
2) отладка модуля;
3) составление блок-схемы алгоритма.
85. Локализация ошибки:
определение места возникновения ошибки;
1) определение причин ошибки;
2) обнаружение причин ошибки;
3) исправление ошибки.
86. Назначение тестирования:
повышение надежности программы;
1) обнаружение ошибок;
2) повышение эффективности программы;
3) улучшение эксплуатационных характеристик;
4) приведение программы к структурированному виду.
87. Назначение отладки:
поиск причин существующих ошибок;
1) поиск возможных ошибок;
2) составление спецификаций;
3) разработка алгоритма.

88. Инструментальные средства отладки (НЕ правильный ответ):
компиляторы;
1) отладчики;
2) трассировка.
89. Отладка программ это:
локализация и исправление ошибок;
1) алгоритмизация программирования;
2) компиляция и компоновка.
90. Что выполняется раньше, автономная или комплексная отладка:
автономная;
1) комплексная.
91. Что выполняется раньше, отладка или тестирование:
отладка;
1) тестирование.
92. Что такое автоматизация программирования:
создание исходного кода программными средствами;
1) создание исходного кода при помощи компилятора;
2) создание исходного кода без разработки алгоритма.
93. В чем сущность автоматизации программирования:
создание программы без написания ее текста;
1) получение готовой программы без выполнения компоновки;
2) в отсутствии компиляции.
94. Возможна ли автоматизация программирования:
да;
1) нет.
95. Создание исполняемого кода программы без написания исходного кода называется:
1) составлением спецификаций;
2) отладкой;
3) проектированием.
автоматизацией программирования;
96. Одно из преимуществ автоматизации программирования:
наглядное программирование с визуальным контролем;
1) получение стандартной программы;
2) создание программы с оптимальным кодом.
97. Один из методов автоматизации программирования:
1) структурное программирование;
2) модульное программирование;
визуальное программирование;
3) объектно-ориентированное программирование.
98. Влияет ли автоматизация программирования на эффективность программы:
нет;
1) да
99. Автоматизация программирования позволяет:
1) повысить надежность программы;
сократить время разработки программы;
2) повысить быстродействие программы.
100. Позволяет ли автоматизация программирования всегда создавать эффективные программы:
1) да.
нет;

К формированию компетенции ПК-3

101. Недостаток автоматизации программирования;

1) низкое быстродействие;

большой размер программы;

2) сложность программы.

102. Что легко поддается автоматизации:

интерфейс;

1) работа с файлами;

2) сложные логические задачи;

3) алгоритмизация.

103. Относится ли визуальное программирование к средствам автоматизации:

104. Уточнение структуры входных и выходных данных, разработка алгоритмов, определение элементов интерфейса входят в:

технический проект;

1) рабочий проект;

2) эскизный проект.

105. Что такое оптимизация программ:

улучшение работы существующей программы;

1) создание удобного интерфейса пользователя;

2) разработка модульной конструкции программы;

3) применение методов объектно-ориентированного программирования.

106. Критерии оптимизации:

время выполнения или размер требуемой памяти;

1) размер программы и ее эффективность;

2) независимость модулей;

3) качество программы, ее надежность.

107. Укажите правильную последовательность создания программы:

формулирование задачи, анализ требований, проектирование, программирование;

A) анализ требований, проектирование, программирование, тестирование, отладка;

B) анализ требований, программирование, проектирование, тестирование;

C) анализ требований, проектирование, программирование, модификация, трассировка;

D) формулирование задачи, анализ требований, программирование, проектирование, отладка.

108. Возможна ли оптимизация программ без участия программиста:

109. Возможна ли оптимизация циклов:

110. В чем заключается оптимизация условных выражений:

в изменении порядка следования элементов выражения;

A) в использовании простых логических выражений;

B) в использовании сложных логических выражений;

C) в использовании операций AND, OR и NOT.

111. Оптимизация циклов заключается в:

уменьшении количества повторений тела цикла;

A) просмотре задачи с другой стороны;

B) упрощение задачи за счет включения логических операций.

112. Оптимизация программы это:

модификация;

A) отладка;

B) повышение сложности программы;

C) уменьшение сложности программы.

113. Критерии оптимизации программы:
быстродействие или размер программы;
А) быстродействие и размер программы;
В) надежность или эффективность;
С) надежность и эффективность.
114. Результат оптимизации программы:
эффективность;
А) надежность;
В) машино-независимость;
С) мобильность.
115. Сущность оптимизации циклов:
сокращение количества повторений выполнения тела цикла;
А) сокращение тела цикла;
В) представление циклов в виде блок-схем;
С) трассировка циклов;
D) поиск ошибок в циклах.
116. В чем сущность модульного программирования:
в разбиении программы на отдельные функционально независимые части;
А) в разбиении программы на отдельные равные части;
В) в разбиение программы на процедуры и функции;
117. Можно ли сочетать модульное и структурное программирование: Варианты ответа:
да;
А) нет.
118. Может ли модуль включать несколько процедур или функций:
да;
А) нет.
119. Рекомендуемые размеры модулей:
небольшие;
А) большие;
В) равные;
С) фиксированной длины.
120. В чем заключается независимость модуля:
в написании, отладке и тестировании независимо от остальных модулей;
А) в разработке и написании независимо от других модулей;
В) в независимости от работы основной программы.
121. При модульном программировании желательно, чтобы модуль имел:
А) большой размер;
небольшой размер;
В) фиксированный размер;
С) любой размер.
122. Модульное программирование это:
разбиение программы на отдельные части;
А) структурирование;
В) использование стандартных процедур и функций.
123. Можно ли использовать оператор GO TO в модульных программах:
можно;
А) нельзя.
124. Разрешается ли использование циклов при модульном программировании:
да;
А) нет.

К формированию компетенции ПК-4

125. Разрешается ли использование условных операторов при модульном программировании:

да;

А) нет.

126. Сократится ли размер программы, если ее написать в виде набора модулей:

нет;

А) да.

127. Достоинство модульного программирования:

создание программы по частям в произвольном порядке;

А) не требует компоновки;

В) всегда дает эффективные программы;

С) снижает количество ошибок.

128. Недостаток модульного программирования:

А) увеличивает трудоемкость программирования;

усложняет процедуру комплексного тестирования;

В) снижает быстродействие программы;

С) не позволяет выполнять оптимизацию программы.

129. Достоинство модульного программирования:

возможность приступить к тестированию до завершения написания всей программы;

А) не требует комплексного тестирования;

В) уменьшает размер программы;

С) повышает надежность программы.

130. При структурном программировании задача выполняется:

поэтапным разбиением на более легкие задачи;

А) без участия программиста;

В) объединением отдельных модулей программы.

131. Разрешается ли использование оператора GO TO при структурном программировании:

нет;

А) да;

В) иногда.

132. Разрешается ли использование циклов при структурном программировании:

да;

А) нет.

133. Разрешается ли использование оператора IF при структурном программировании:

да;

А) нет.

134. Программирование без GO TO применяется при:

структурном программировании;

А) модульном программировании;

В) объектно-ориентированном программировании;

С) все ответы верные.

135. Достоинство структурного программирования:

можно приступить к комплексному тестированию на раннем этапе разработки;

А) можно приступить к автономному тестированию на раннем этапе разработки;

В) нет необходимости выполнять тестирование;

- С) можно пренебречь отладкой.
136. Достоинство структурного программирования:
облегчает работу над большими и сложными проектами;
А) повышает быстродействие программы;
В) снижает затраты на программирование.
137. Недостаток структурного программирования:
увеличивает размер программы;
А) снижает эффективность;
В) уменьшает количество ошибок;
С) не требует отладки.
138. Повышает ли читабельность программ структурное кодирование:
да;
А) нет.
- К формированию компетенции ПК-6**
139. Что такое объект, в объектно-ориентированном программировании:
тип данных;
А) структура данных;
В) событие;
С) обработка событий;
D) использование стандартных процедур.
140. Инкапсуляция это:
А) определение новых типов данных;
В) определение новых структур данных;
объединение переменных, процедур и функций в одно целое;
С) разделение переменных, процедур и функций;
D) применение стандартных процедур и функций.
141. Наследование это:
А) передача свойств экземплярам;
В) передача свойств предкам;
передача свойств потомкам;
С) передача событий потомкам.
142. Полиморфизм это:
изменение поведения потомков, имеющих общих предков;
А) передача свойств по наследству;
В) изменение поведения потомков на разные события;
С) изменение поведения экземпляров, имеющих общих предков;
143. Три "кита" объектно-ориентированного метода программирования:
А) предки, родители, потомки;
полиморфизм, инкапсуляция, наследование;
В) свойства, события, методы;
С) визуальные, не визуальные компоненты и запросы.
144. Какое утверждение верно:
А) предки наследуют свойства родителей;
В) родители наследуют свойства потомков;
С) потомки не могут иметь общих предков;
потомки наследуют свойства родителей.
145. Может ли дочерний элемент иметь двух родителей:
А) да;
нет;
В) только для визуальных элементов;
С) если их свойства совпадают.
146. Могут ли два визуальных компонента иметь общего предка:

да;

А) нет;

В) если их свойства совпадают;

С) если их методы совпадают.

147. Составление спецификаций это:

формализация задачи;

А) эскизный проект;

В) поиск алгоритма;

С) отладка.

148. Есть ли различие между объектом и экземпляром:

да;

А) нет;

В) если у них общий предок.

149. Есть ли различие в поведении объекта и экземпляра того же типа:

А) да;

В) если у них есть общий предок;

нет;

С) если у них нет общего предков.

150. Изменение свойств, приводит к изменению поведения экземпляра:

А) нет;

В) только для визуальных;

С) только НЕ для визуальных;

да.

151. Можно ли свойствам присваивать значения:

А) да (всегда);

не всегда;

В) нет.

152. Какой методикой проектирования пользуются при структурном программировании:

сверху вниз;

1) снизу-вверх.

165. Какой этап проектирования может быть исключен:

эскизный проект;

1) технический проект;

2) рабочий проект.

153. Какие этапы проектирования можно объединять:

технический и рабочий;

1) эскизный и рабочий;

2) технический и эскизный.

154. Модульное программирование применимо при:

1) проектировании сверху вниз;

проектирование снизу-вверх;

155. Процесс преобразования постановки задачи в план алгоритмического или вычислительного решения это:

проектирование;

1) анализ требований;

2) программирование;

3) тестирование.

Критерии и шкала оценивания тестирования в п.3.2. Приложения 1

2. Перечень компетенций, этапы их формирования,

описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1 Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных аспирантом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для аспиранта к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность аспиранта использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам,

в том числе по наличию более широких знаний о классических и основных теориях информации; углублённых знаний о принципах формирования и реализации информационных процессов; умению применять современные информационные технологии для поиска и обработки актуальной информации, демонстрацию владения современными моделями СППР и навыками программирования при решении социально значимых задач.

Высокий уровень - предполагает способность аспиранта использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам,

**2.2 Перечень компетенций, этапы их формирования,
описание показателей и критериев оценивания компетенций
на различных этапах их формирования**

Код компетенции	Этапы формирования (№ темы, задачи)	Критерии оценивания результатов обучения		
		Пороговый уровень	Продвинутый уровень	Высокий уровень
ОПК-3	Темы 1-6 Вопросы 1-44	В целом успешное, но не систематическое умение разрабатывать новые методы исследования и их применения в самостоятельной научно-исследовательской деятельности в области профессиональной деятельности.	В целом успешное, но сопровождающееся отдельными недостатками умение разрабатывать новые методы исследования и их применения в самостоятельной научно-исследовательской деятельности в области профессиональной деятельности.	Успешное и систематическое умение разрабатывать новые методы исследования и их применения в самостоятельной научно-исследовательской деятельности в области профессиональной деятельности.
ПК-1	Темы 1-6 Вопросы 45-68	<ol style="list-style-type: none"> 1. Знание современных основ программирования. 2. Знание основных принципов организации вычислительных машин, систем и сетей. 3. Знание современных языков и систем программирования. 4. Владение основами организации операционных систем. 	<ol style="list-style-type: none"> 1. Знание математических основ программирования. 2. Знание основных принципов организации современных вычислительных машин, систем и сетей. 3. Знание технологий разработки программного обеспечения. 4. Знание принципов организации операционных систем. 5. Знание методов хранения данных и доступа к ним. организация баз данных и знаний. 6. Знание аппарата защиты данных и программных систем. 	<ol style="list-style-type: none"> 1. Знание и владение математическими основами программирования. 2. Знание основных принципов разработки и эксплуатации вычислительных машин, систем и сетей. 3. Знание языков, систем программирования и технологий разработки программного обеспечения. 4. Знание принципов организации современных операционных систем. 5. Владение методами хранения данных и доступа к ним. Знание современной организация баз данных и знаний. 6. Владение аппаратом защиты данных и программных систем.
ПК-2	Темы 1-6 Вопросы 69-100	В целом успешное, но не систематическое умение разрабатывать новые	В целом успешное, но сопровождающееся отдельными недостатками умение разрабатывать новые математические модели	Успешное и систематическое умение разрабатывать новые математические модели объектов и явлений, развивать аналитические

		математические модели объектов и явлений, развивать аналитические и приближенные методы их исследования, выполнять реализацию эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента.	объектов и явлений, развивать аналитические и приближенные методы их исследования, выполнять реализацию эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента.	и приближенные методы их исследования, выполнять реализацию эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента.
ПК-3	Темы 1-6 Вопросы 101-124	В целом успешное, но не систематическое умение разрабатывать новые алгоритмы, уметь анализировать трудоемкость алгоритмов и их потребность в памяти, использовать различные языки программирования для разработки программ, уметь их тестировать, оценивать качество с учетом стандартов.	В целом успешное, но сопровождающееся отдельными недостатками умение разрабатывать новые алгоритмы, уметь анализировать трудоемкость алгоритмов и их потребность в памяти, использовать различные языки программирования для разработки программ, уметь их тестировать, оценивать качество с учетом стандартов.	Успешное и систематическое умение разрабатывать новые алгоритмы, уметь анализировать трудоемкость алгоритмов и их потребность в памяти, использовать различные языки программирования для разработки программ, уметь их тестировать, оценивать качество с учетом стандартов.
ПК-4	Темы 1-6 Вопросы 125-138	В целом успешное, но не систематическое умение объективно оценивать профессиональный уровень результатов научных исследований, в том числе с помощью международных баз данных публикационной активности.	В целом успешное, но сопровождающееся отдельными недостатками умение объективно оценивать профессиональный уровень результатов научных исследований, в том числе с помощью международных баз данных публикационной активности.	Успешное и систематическое умение объективно оценивать профессиональный уровень результатов научных исследований, в том числе с помощью международных баз данных публикационной активности.
ПК-6	Темы 1-6 Вопросы 139-155	В целом успешное, но не систематическое умение оформлять результаты своей научно-исследовательской деятельности в форме научно-квалификационной работы по научной специальности 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.	В целом успешное, но сопровождающееся отдельными недостатками умение оформлять результаты своей научно-исследовательской деятельности в форме научно-квалификационной работы по научной специальности 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.	Успешное и систематическое умение оформлять результаты своей научно-исследовательской деятельности в форме научно-квалификационной работы по научной специальности 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.

Оценочные средства по формам контроля

Текущий контроль	
Номер мероприятия	Название темы
1	Тема 1. Системы программирования
2	Тема 2.1. Технология разработки программного обеспечения
Задание для контрольного мероприятия	Написание реферата
Требования к выполнению тестового задания	<p>1. Написание реферата на любую из нижеперечисленных тем.</p> <p>2. Реферат сдается в бумажном и электронном виде</p> <p>3. При проверке реферата на антиплагиат (более 30% заимствований) работа не засчитывается.</p> <p>Критерии оценки по содержанию и качеству:</p> <p>0 – отсутствие реферата;</p> <p>71-89% своего текста – 4, выполнения тестового задания;</p> <p>90-100% своего текста – 5.</p> <p>При проверке реферата используется сайт www.antiplagiat.ru</p>
4	Письменный экзамен в соответствии с содержанием дисциплины

3	Тема 2.2. Технология разработки программного обеспечения
Содержание задания для проверочного мероприятия	Тестовые задания содержат 15 вопросов
Требования к выполнению тестового задания	<p>1. Выполняется письменно.</p> <p>2. Верное выполнение теоретических и практических тестовых заданий.</p> <p>3. Врем выполнения тестовых заданий – 30 мин.</p>
Критерии оценки по содержанию и качеству выполнения тестового задания	<p>1. Критерии оценки выполнения тестового задания: «верно»; «неверно».</p> <p>2. Верное выполнение оценивается в 1 (один) балл.</p> <p>3. Неверно выполненное отдельное тестовое задание оценивается в 0 баллов.</p>
Методика обработки и форматы представления результатов оценочных процедур	<p>При обработке результатов оценочной процедуры используются ключи:</p> <p>-верное выполнение оценивается в 1 (один) балл;</p> <p>-неверно выполненное отдельное тестовое задание оценивается в 0 баллов;</p> <p>-неявка оценивается в 0 баллов.</p>

3. Методические рекомендации преподавателю

по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения аспирантом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения аспирантом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе 1. «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;

- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2. Шкала оценивания успеваемости текущего контроля и промежуточной аттестации.

Оценивание результатов обучения аспирантов по дисциплине «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» осуществляется по регламенту текущего контроля и промежуточной аттестации.

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы аспирантов.

Текущий контроль проводится в виде внеаудиторной самостоятельной работы и тестирования.

Показатели	Критерии
Вопрос к кратким ответом	Правильный ответ – 1 балл Неверный ответ – 0 баллов
Вопрос с выбором одного ответа	Правильный ответ – 1 балл Неверный ответ – 0 баллов
Вопрос с множественным выбором	Каждый правильный ответ – 1 балл Неверный ответ – 0 баллов
Вопрос на соответствие	Все соответствия верные – 1 балл Иначе – 0 баллов

Оценка проставляется по количеству набранных баллов:

менее 60% от максимально возможного количества баллов - неудовлетворительно,
60-74% от максимально возможного количества баллов - удовлетворительно,
75-84% от максимально возможного количества баллов - хорошо,
85-100% от максимально возможного количества баллов – отлично.

Оценка сформированности компетенции (для теста)

Оценка	Уровень	Компетенция
5	Высокий	В тесте по компетенции ПК-1 аспирант отвечает правильно хотя бы на 85% вопросов.
4	Продвинутый	В тесте по компетенции ПК-1 аспирант отвечает правильно хотя бы на 75% вопросов.
3	Пороговый	В тесте по компетенции ПК-1 аспирант отвечает правильно хотя бы на 60% вопросов.
2	Не сформирована	В тесте по компетенции ПК-1 аспирант отвечает правильно менее чем на 60% вопросов.
Оценка	Уровень	ОПК-3
5	Высокий	В тесте по компетенции аспирант отвечает правильно хотя бы на 85% вопросов.
4	Продвинутый	В тесте по компетенции аспирант отвечает правильно хотя бы на 75% вопросов.
3	Пороговый	В тесте по компетенции аспирант отвечает правильно хотя бы на 60% вопросов.
2	Не сформирована	В тесте по компетенции аспирант отвечает правильно менее чем на 60% вопросов.
Оценка	Уровень	ПК-2
5	Высокий	В тесте по компетенции аспирант отвечает правильно

		хотя бы на 85% вопросов.
4	Продвинутый	В тесте по компетенции аспирант отвечает правильно хотя бы на 75% вопросов.
3	Пороговый	В тесте по компетенции аспирант отвечает правильно хотя бы на 60% вопросов.
2	Не сформирована	В тесте по компетенции аспирант отвечает правильно менее чем на 60% вопросов.
Оценка	Уровень	ПК-3
5	Высокий	В тесте по компетенции аспирант отвечает правильно хотя бы на 85% вопросов.
4	Продвинутый	В тесте по компетенции аспирант отвечает правильно хотя бы на 75% вопросов.
3	Пороговый	В тесте по компетенции аспирант отвечает правильно хотя бы на 60% вопросов.
2	Не сформирована	В тесте по компетенции аспирант отвечает правильно менее чем на 60% вопросов.
Оценка	Уровень	ПК-4
5	Высокий	В тесте по компетенции аспирант отвечает правильно хотя бы на 85% вопросов.
4	Продвинутый	В тесте по компетенции аспирант отвечает правильно хотя бы на 75% вопросов.
3	Пороговый	В тесте по компетенции аспирант отвечает правильно хотя бы на 60% вопросов.
2	Не сформирована	В тесте по компетенции аспирант отвечает правильно менее чем на 60% вопросов.
Оценка	Уровень	ПК-6
5	Высокий	В тесте по компетенции аспирант отвечает правильно хотя бы на 85% вопросов.
4	Продвинутый	В тесте по компетенции аспирант отвечает правильно хотя бы на 75% вопросов.
3	Пороговый	В тесте по компетенции аспирант отвечает правильно хотя бы на 60% вопросов.
2	Не сформирована	В тесте по компетенции аспирант отвечает правильно менее чем на 60% вопросов.

3.3. Критерии и шкала оценивания успеваемости итоговой аттестации.

Критерии оценки ответа на экзамене

Ответ обучающегося на экзамене оцениваются преподавателем по **четырёхбалльной системе**, оценка по учебной дисциплине выставляется в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации студентов/аспирантов в ЯрГУ.

Оценка «отлично» — глубокие, исчерпывающие знания всего программного материала, понимание сущности и взаимосвязи рассматриваемых процессов и явлений, твердое знание основных положений смежных дисциплин: логически последовательные, содержательные, полные, правильные и конкретные ответы на все вопросы экзаменационного билета.

Оценка «хорошо» — твердые и достаточно полные знания всего программного материала, правильное понимание сущности и взаимосвязи рассматриваемых процессов и явлений; последовательные, правильные,

конкретные ответы на поставленные вопросы; при ответах на вопросы могут быть допущены отдельные незначительные неточности, но в целом ответ дан верный.

Оценка «удовлетворительно» — твердое знание и непонимание основных вопросов программы; правильные и конкретные, без грубых ошибок ответы на поставленные вопросы, при ответах на отдельные вопросы допущены серьезные неточности.

Оценка «неудовлетворительно» — неправильный ответ хотя бы на один из основных вопросов, грубые ошибки в ответе, непонимание сущности излагаемых вопросов.

Оценка сформированности компетенций при итоговой аттестации (экзамен).

оценка	уровень	компетенции
отлично	высокий,	ОПК-3, ПК-1, ПК-2, ПК-3, ПК-4, ПК-6
хорошо	продвинутый,	ОПК-3, ПК-1, ПК-2,
удовлетворительно	пороговый	ОПК-3, ПК-1
неудовлетворительно	Не сформирован	Не сформированы

**Приложение № 2 к рабочей программе дисциплины
«Математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей»**

Методические указания для аспирантов по освоению дисциплины

Освоение обучающимся учебной дисциплины «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» предполагает изучение материалов дисциплины на аудиторных занятиях и в ходе самостоятельной работы. Аудиторные занятия проходят в форме лекций, семинаров, лабораторных работ и практических занятий. Самостоятельная работа включает разнообразный комплекс видов и форм работы обучающихся.

Для успешного освоения учебной дисциплины и достижения поставленных целей необходимо внимательно ознакомиться с настоящей рабочей программой учебной дисциплины. Ее может представить преподаватель на вводной лекции или самостоятельно обучающийся использует информацию на официальном Интернет-сайте Университета.

Следует обратить внимание на список основной и дополнительной литературы, которая имеется в электронной библиотечной системе Университета, на предлагаемые преподавателем ресурсы информационно-телекоммуникационной сети Интернет. Эта информация необходима для самостоятельной работы обучающегося.

При подготовке к аудиторным занятиям необходимо помнить особенности каждой формы его проведения.

Подготовка к учебному занятию лекционного типа заключается в следующем.

С целью обеспечения успешного обучения обучающийся должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

- } знакомит с новым учебным материалом;
- } разъясняет учебные элементы, трудные для понимания;
- } систематизирует учебный материал;
- } ориентирует в учебном процессе. С этой целью:
 - } внимательно прочитайте материал предыдущей лекции;
 - } ознакомьтесь с учебным материалом по учебнику и учебным пособиям с темой прочитанной лекции;
 - } внесите дополнения к полученным ранее знаниям по теме лекции на полях лекционной тетради;
 - } запишите возможные вопросы, которые вы зададите лектору на лекции по материалу изученной лекции;
 - } постарайтесь уяснить место изучаемой темы в своей подготовке;
 - } узнайте тему предстоящей лекции (по тематическому плану, по информации лектора) и запишите информацию, которой вы владеете по данному вопросу

Подготовка к занятию семинарского типа

При подготовке и работе во время проведения занятий семинарского типа следует обратить внимание на следующие моменты: на процесс предварительной подготовки, на работу во время занятия, обработку полученных результатов, исправление полученных замечаний.

Предварительная подготовка к учебному занятию семинарского типа заключается в изучении теоретического материала в отведенное для самостоятельной работы время, ознакомление с инструктивными материалами с целью осознания задач практического занятия, техники безопасности при работе с приборами, веществами.

Работа во время проведения учебного занятия семинарского типа включает несколько моментов:

} консультирование аспирантов преподавателями и вспомогательным персоналом с целью предоставления исчерпывающей информации, необходимой для самостоятельного выполнения предложенных преподавателем задач, ознакомление с правилами техники безопасности при работе в лаборатории;

} самостоятельное выполнение заданий согласно обозначенной учебной программой тематики;

Обработка, обобщение полученных результатов заданий проводится обучающимися самостоятельно или под руководством преподавателя (в зависимости от степени сложности поставленных задач). В результате оформляется индивидуальный отчет. Подготовленная к сдаче на контроль и оценку работа сдается преподавателю. Форма отчетности может быть письменная, устная или две одновременно. Главным результатом в данном случае служит получение положительной оценки по практическому занятию. Это является необходимым условием при проведении рубежного контроля и допуска к экзамену. При получении неудовлетворительных результатов обучающийся имеет право в дополнительное время пересдать преподавателю работу до промежуточной аттестации.

Самостоятельная работа.

Для более углубленного изучения темы задания для самостоятельной работы рекомендуется выполнять параллельно с изучением данной темы. При выполнении заданий по возможности используйте наглядное представление материала. Более подробная информация о самостоятельной работе представлена в разделах «Учебно- методическое обеспечение самостоятельной работы по дисциплине (модулю)»,

«Методические указания к самостоятельной работе по дисциплине (модулю)».

Подготовка к экзамену.

К экзамену необходимо готовится целенаправленно, регулярно, систематически и с первых дней обучения по данной дисциплине. Попытки освоить учебную дисциплину в период зачетно-экзаменационной сессии, как правило, приносят не слишком удовлетворительные результаты.

При подготовке к экзамену обратите внимание на защиту практических заданий на основе теоретического материала.

После предложенных указаний у обучающихся должно сформироваться четкое представление об объеме и характере знаний и умений, которыми надо будет овладеть по дисциплине.

Учебно-методическое обеспечение самостоятельной работы аспирантов по дисциплине

Тема 1. Системы программирования

Цель: заключается в получении обучающимися теоретических знаний о средствах и методах программирования с последующим применением в

профессиональной сфере и формирование практических навыков программирования, используя современные инструментальные средства.

Перечень изучаемых элементов содержания

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 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. Построение абстрактного синтаксического дерева. Автоматическое построение лексических и синтаксических анализаторов по формальным описаниям грамматик.

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.

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 (СП), типовые компоненты СП: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы. Модульное программирование. Типы модулей. Связывание модулей по управлению и данным.

Вопросы для самоподготовки

1. Языки программирования.
2. Процедурные языки программирования.
3. Объектно-ориентированное программирование.
4. Распределенное программирование.
5. Основы построения трансляторов.
6. Анализ исходной программы в компиляторе.
7. Оптимизация программ при их компиляции.
8. Генерация объектного кода в компиляторах.
9. Машинно-ориентированные языки, язык ассемблера.
10. Системы программирования (СП), типовые компоненты СП: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы.
11. Модульное программирование. Типы модулей. Связывание модулей по управлению и данным.

Формы контроля самостоятельной работы обучающихся: реферат.

Литература по теме:

1. Зыков С.В. Введение в теорию программирования. Объектно-ориентированный подход. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. - 189 с. <http://www.book.ru/book/917673>

Список полезных Интернет-ресурсов:

proklondike.com. Отличный сайт, предлагающий множество учебников по всем разделам программирования: языки программирования, веб-технологии, разработка приложений, базы данных, операционные системы. Все книги выложены на сайте, их можно быстро и удобно скачать. Также публикуются статьи по программированию.

ph4s.ru. Множество книг и учебников по программированию и компьютерным технологиям, доступные для бесплатного скачивания (на этом же сайте еще учебники по другим наукам).

programm.ws. Небольшой, но весьма полезный сайт с книгами и уроками по Java, C++, Delphi и Assembler. Почему полезный? Выложены полнотекстовые сборники задач и примеров на разных языках: постановка задачи, разбор, исходник, скриншот.

godlab.ru. Статьи и учебники по программированию, бесплатные видеоуроки, скрипты, шаблоны, программы и многое другое.

Тема 2. Технология разработки программного обеспечения

Цель: заключается в получении обучающимися теоретических знаний о средствах и методах разработки программного обеспечения с последующим применением в профессиональной сфере и формирование практических навыков по разработке программных систем, способности решать задачи комплексной разработки ПО, используя современные инструментальные средства

Перечень изучаемых элементов содержания

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

Вопросы для самоподготовки

1. Технология разработки и сопровождения программ.
2. Жизненный цикл программы.
3. Декомпозиционные и сборочные технологии, механизмы наследования, инкапсуляции, задания типов.
4. Модули, взаимодействие между модулями, иерархические структуры программ.
5. 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.
6. Методы спецификации программ.
7. Методы проверки спецификации.
8. Схемное, структурное, визуальное программирование.
9. Разработка пользовательского интерфейса, мультимедийные среды интерфейсного взаимодействия

Форма контроля самостоятельной работы обучающихся: реферат.

Критериями оценки реферата являются: новизна текста, обоснованность выбора источников литературы, степень раскрытия сущности вопроса, научность; соблюдения требований к оформлению.

Оценка **«отлично»**: выполнены все требования к написанию реферата: обозначена проблема и обоснована её актуальность, сделан анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция; сформулированы выводы, тема раскрыта полностью, выдержан объём; соблюдены требования к внешнему оформлению.

Оценка **«хорошо»**: основные требования к реферату выполнены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объём реферата; имеются упущения в оформлении.

Оценка **«удовлетворительно»**: имеются существенные отступления от требований к реферированию. В частности: тема освещена лишь частично; допущены фактические ошибки в содержании реферата; отсутствуют выводы.

Оценка **«неудовлетворительно»**: тема реферата не раскрыта, обнаруживается существенное непонимание проблемы или реферат не представлен вовсе.

Оценка сформированности компетенций при самостоятельной работе аспиранта (реферат).

оценка	уровень	компетенции
отлично	высокий,	ОПК-3, ПК-1, ПК-2, ПК-3, ПК-4, ПК-6
хорошо	продвинутый,	ОПК-3, ПК-1, ПК-2, ПК-3
удовлетворительно	пороговый	ОПК-3, ПК-1, ПК-2
неудовлетворительно	Не сформирован	Не сформированы

Литература по теме

1. Бурков А.В. Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. - 311 с. <http://www.book.ru/book/918135/>

2. Грекул В.И. Проектирование информационных систем. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. - 571 с. <http://www.book.ru/book/918134>

3. Золотов С. Ю. Проектирование информационных систем: учебное пособие. Томск: ЭльКонтент, 2013. - 88 с. http://biblioclub.ru/index.php?page=book_red&id=208706&sr=1

Список полезных Интернет-ресурсов:

<http://www.grand-fair.ru/cat/PO/> Методический сайт по курсу Технология разработки программного обеспечения

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», рекомендованных к использованию при освоении дисциплины

proklondike.com. Отличный сайт, предлагающий множество учебников по всем разделам программирования: языки программирования, веб-технологии, разработка приложений, базы данных, операционные системы. Все книги выложены на сайте, их можно быстро и удобно скачать. Также публикуются статьи по программированию.

ph4s.ru. Множество книг и учебников по программированию и компьютерным технологиям, доступные для бесплатного скачивания (на этом же сайте еще учебники по другим наукам).

programm.ws. Небольшой, но весьма полезный сайт с книгами и уроками по Java, C++, Delphi и Assembler. Почему полезный? Выложены полнотекстовые сборники задач и примеров на разных языках: постановка задачи, разбор, исходник, скриншот.

godlab.ru. Статьи и учебники по программированию, бесплатные видеоуроки, скрипты, шаблоны, программы и многое другое.

<http://www.grand-fair.ru/cat/PO/> Методический сайт по курсу Технология разработки программного обеспечения

<http://agilerussia.ru/> Сайт посвящен различным аспектам гибкой (Agile) разработки программного обеспечения

информационно-поисковая система Google www.google.ru;

всемирная свободная Интернет-энциклопедия Wikipedia ru.wikipedia.org;

Web-портал компании Microsoft для поддержки программы Access office.microsoft.com/ru-ru/access;

академическая программа Softline Academy Alliance www.it-academy.ru; Web-портал по информационным технологиям CIT Forum citforum.ru; Национальный открытый интернет-университет «ИНТУИТ» www.intuit.ru;

Web-ресурс по информационным технологиям INTERFACE.RU www.interface.ru; интернет-портал «Информационно-коммуникационные технологии в образовании»

www.ict.edu.ru;

Российский портал открытого образования www.openet.edu.ru; Web-ресурс разработчиков информационных систем www.rsdn.ru; электронная библиотека стандартов IT-GOST.RU it-gost.ru;

Web-ресурс журнала «Открытые системы» www.osp.ru;

Web-ресурс журнала «Хакер» www.xakep.ru/articles/magazine/default.asp; Web-сайт по базам данных archae-dev.com;

Web-форум SQL.RU www.sql.ru/forum; Web-ресурс SQL.RU www.sql.ru;

Центр Инноваций компании Microsoft в УрФУ mic.usu.ru;

Web-ресурс Академической образовательной программы Softline Academy Alliance www.it-academy.ru;

Web-портал компании Microsoft в России www.microsoft.com/ru-ru; Web-портал по продуктам компании Microsoft msdn.microsoft.com/ru-ru; Web-портал компании Oracle в России www.oracle.com/ru;

Web-портал компании SyBase в России www.sybase.ru; Web-ресурс по СУБД MySQL mysql.ru;

Web-ресурс по СУБД PostgreSQL www.postgresql.org.