



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

«СОГЛАСОВАНО»

Руководитель образовательной программы

А.С. Величко

«15» июля 2017 г.

«УТВЕРЖДАЮ»

Заведующий кафедрой
математических методов в экономике

А.С. Величко

«15» июля 2017 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Теоретические основы компьютерных наук

Направление подготовки 01.03.04 Прикладная математика

профиль «Математические методы в экономике»

Форма подготовки очная

курс 1 семестр 1, 2

лекции 36 час.

практические занятия не предусмотрены

лабораторные работы 36 час.

в том числе с использованием МАО лек. 0 час. / пр. 0 час. / лаб. 36 час.

всего часов аудиторной нагрузки 72 час.

в том числе с использованием МАО 36 час.

самостоятельная работа 144 час.

в том числе на подготовку к экзамену 54 час.

контрольные работы (количество) 6

курсовая работа / курсовой проект не предусмотрены

зачет не предусмотрен

экзамен 1,2 семестр

Рабочая программа учебной дисциплины (РПУД) составлена в соответствии с требованиями образовательного стандарта по направлению 01.03.04 «Прикладная математика», самостоятельно устанавливаемого ДВФУ, утвержденного приказом ректора от 18.02.2016 № 12-13-235

Рабочая программа обсуждена на заседании кафедры математических методов в экономике, протокол № 16 от «15» июля 2017 г.

Врио заведующего кафедрой математических методов в экономике, к.ф.-м.н., доцент А.С. Величко

Составитель:

старший преподаватель кафедры математических методов в экономике Е.А. Воронцова

АННОТАЦИЯ

Дисциплина «Теоретические основы компьютерных наук» предназначена для студентов направления подготовки 01.03.04 «Прикладная математика», профиль «Математические методы в экономике».

Общая трудоемкость освоения дисциплины составляет 6 зачётных единиц (2160 часов). Дисциплина реализуется на 1 курсе в 1-м и в 2-м семестрах. Дисциплина входит в обязательные дисциплины базовой части блока «Дисциплины (модули)».

Особенности построения курса: лекции (36 часов), лабораторные работы (36 часа), самостоятельная работа (90 часов), подготовка к экзамену (54 часа).

Содержание дисциплины охватывает следующий круг вопросов: теория информации, аппаратные средства информатики, технология разработки программ, системы счисления, виды сортировок массива, адресная арифметика, работа с указателями и массивами, отладка и тестирование программного обеспечения.

Цель - дать представление об основах информационной культуры; изучить аппаратные средства реализации современных компьютеров и вычислительных систем; изучить основные виды программных продуктов и технологии их разработки; дать представления о возможностях реализации программных продуктов.

Задачи:

- развитие способности освоить основные понятия теории информации;
- развитие готовности демонстрировать знания назначения основных аппаратных компьютерных средств;
- развитие способности проектировать, разрабатывать, отлаживать и тестировать программы на языке программирования C++;
- развитие готовности применять знания и навыки управления информацией;

- развитие способности и готовности демонстрировать знания современного языка программирования;
- развитие способности отлаживать и тестировать прикладное программное обеспечение;
- развитие способности настраивать, тестировать и осуществлять проверку вычислительной техники и программных средств;
- развитие готовности настраивать, тестировать и осуществлять проверку вычислительной техники и программных средств;
- развитие готовности осваивать современные технологии программирования;
- развитие способности демонстрировать знания современных технологий программирования.

Для успешного изучения дисциплины «Теоретические основы компьютерных наук» у обучающихся должны быть сформированы следующие предварительные компетенции:

- способность использовать для работы операционную систему ЭВМ Microsoft Windows и информационно-телекоммуникационную сеть "Интернет".

Уровень подачи материала курса в достаточной степени опирается на школьный курс информатики.

В результате изучения данной дисциплины у обучающихся формируются следующие общепрофессиональные, профессиональные компетенции (элементы компетенций).

Код и формулировка компетенции	Этапы формирования компетенции	
ОК-5 -способностью использовать современные методы и технологии (в том числе информационные) в профессиональной деятельности	Знает	современные языки программирования (в частности, языки C и C++)
	Умеет	применять знания современных языков программирования для разработки программных продуктов
	Владеет	навыками разработки программ на современных языках программирования

ОПК-2 - способность использовать современные математические методы и современные прикладные программные средства и осваивать современные технологии программирования	Знает	современные технологии программирования на языке С++
	Умеет	разрабатывать программные продукты с помощью современных технологий программирования
	Владеет	эффективными методами разработки программных продуктов с помощью современных технологий программирования
ПК-11 – готовность применять знания и навыки управления информацией	Знает	основные принципы управления информацией
	Умеет	применять знания основных принципов управления информацией
	Владеет	навыками управления информацией

Для формирования вышеуказанных компетенций в рамках дисциплины «Теоретические основы компьютерных наук» применяются следующие методы активного/интерактивного обучения: выполнение проектов при создании программного обеспечения с использованием компьютерных технологий.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Лекции (36 часов)

МОДУЛЬ 1 (18 ЧАСОВ)

Тема 1. Основы информационной культуры (2 часа)

Понятие информации. Информационные революции в истории развития человечества. Информационный потенциал общества. Информационные ресурсы и услуги.

Тема 2. Классификация и кодирование информации (2 часа)

Классификация информации. Системы кодирования информации. Меры информации.

Тема 3. Арифметические основы вычислительных машин (6 часов)

Системы счисления. Целые числа: знаковые и беззнаковые. Диапазоны целых. Арифметические действия с целыми и дробными числами в различных системах счисления. Кодирование символов. Таблица символов ASCII. Расширения таблицы ASCII. Представление чисел в памяти ЭВМ.

Тема 4. Аппаратное обеспечение ЭВМ (8 часов)

Материнская плата. Центральный процессор. Сопроцессор. Оперативная память и организация её работы. Накопители. Контроллеры. Устройства ввода-вывода. Flash-память.

МОДУЛЬ 2 (18 ЧАСОВ)

Тема 1. Основные этапы разработки ПО (4 часа)

Анализ требований. Проектирование. Макетирование. Кодирование. Отладка. Виды тестирования. Документирование ПО. Типология ПО. Алгоритм. Примеры. Блок-схема алгоритма.

Тема 2. Элементы языка Си++ (2 часа)

Динамические массивы. Контейнеры. Класс vector. Тип данных «структура». Тип данных «ссылка».

Тема 3. Алгоритмы сортировки (4 часа)

Прямой выбор. Пузырьковая сортировка. Шейкерная сортировка. Сортировка простым слиянием. Сортировка Шелла. Быстрая сортировка. Анализ сортировок. Оценка сложности. O-символика.

Тема 4. Динамические структуры данных (8 часов)

Список. Стек. Очередь. Бинарные деревья. Основные операции над элементами. Правила обхода деревьев. Примеры. Способы реализации на языке программирования C++. Хеш-таблицы. Применение.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Лабораторные работы (36 часов, в том числе 72 часа с использованием методов активного обучения)

Модуль 1 (18 часов, в том числе 18 часов с использованием методов активного обучения)

Лабораторная работа № 1. Среда разработки Microsoft Visual Studio (1 час)

Лабораторная работа № 2. Алгоритм. Блок-схемы. (2 часа)

Лабораторная работа № 3. Системы счисления (3 часа)

Лабораторная работа № 4. Разработка программного обеспечения. Простые программы. Ввод-вывод информации в программе (2 часа)

Лабораторная работа № 5. Организация памяти. Адресная арифметика (5 часов)

Лабораторная работа № 6. Работа с указателями и массивами в Си++ (5 часов)

Модуль 2 (18 часов, в том числе 18 часов с использованием методов активного обучения)

Лабораторная работа № 1. Проектирование программного обеспечения (4 часа)

Лабораторная работа № 2. Отладка и тестирование программного обеспечения (4 часа)

Лабораторная работа № 3. Виды сортировок массива (6 часов)

Лабораторная работа № 4. Динамические структуры данных. Реализация на С++ (4 часа)

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Теоретические основы компьютерных наук» представлено в Приложении 1 и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

Контролируемые разделы дисциплины, этапы формирования компетенций, виды оценочных средств, зачетно-экзаменационные материалы, комплекты оценочных средств для текущей аттестации, описание показателей и критериев оценивания компетенций на различных этапах их формирования, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература
(электронные и печатные издания)

1. Колдаев В.Д., Лупин С.А. Архитектура ЭВМ [Электронный ресурс] : учебное пособие. — М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. — 384 с. — Режим доступа: <http://znanium.com/go.php?id=375092>.

2. Дейл Н. Программирование на С++ [Электронный ресурс] : учебник / Дейл Н., Уимз Ч., Хедингтон М. — Электрон. дан. — М. : ДМК Пресс, 2007. — 672 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1219.

3. Тяпичев Г.А. Быстрое программирование на С++ [Электронный ресурс]. — Электрон. дан. — М.: СОЛОН-Пресс, 2008. — 373 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=13688.

4. Подбельский В.В. Курс программирования на языке Си [Электронный ресурс]: учебник / Подбельский В.В., Фомин С.С. — Электрон. текстовые данные. — М.: ДМК Пресс, 2012. — 384 с.— Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=4148.

Дополнительная литература *(печатные и электронные издания)*

1. Златопольский Д.М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс]. — Электрон. дан. — М.: "Лаборатория знаний" (ранее "БИНОМ. Лаборатория знаний"), 2015. — 224 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=70753.

2. Степанов В.П. Лабораторный практикум по программированию на языке Си [Электронный ресурс] : учебное пособие. — Электрон. дан. — М.: МГТУ им. Н.Э. Баумана, 2007. — 48 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=52383.

3. Арипова О.В. Программирование на языке высокого уровня: лабораторный практикум для вузов [Электронный ресурс] / Арипова О.В., Гуцин А.Н., Палехова О.А. — Электрон. дан. — СПб. : БГТУ "Военмех" им. Д.Ф. Устинова (Балтийский государственный технический университет

«Военмех» имени Д.Ф. Устинова), 2014. — 96 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=63671.

4. Грузина Э.Э. Практикум по программированию. – Ч. I [Электронный ресурс] : учебное пособие / Грузина Э.Э., Черноусова Н.Л.. — Электрон. дан. — Кемерово : Издательство КемГУ (Кемеровский государственный университет), 2013. — 100 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=58312.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Язык программирования C++. Учебник [Электронный ресурс].
URL: <http://cppstudio.com/cat/274/>
2. Visual C++ Guided Tour. URL:
<http://msdn.microsoft.com/en-us/library/vstudio/ms235630.aspx>
3. Онлайн-курс «Введение в программирование (C++)». Академия Яндекса, Высшая школа экономики (НИУ ВШЭ).
[https://stepic.org/course/Введение-в-программирование-\(C++\)-363/syllabus](https://stepic.org/course/Введение-в-программирование-(C++)-363/syllabus)
4. Онлайн-курс «Основы алгоритмов» University of California, San Diego, Higher School of Economics. <https://www.coursera.org/learn/algorithmic-toolbox>

Перечень дополнительных информационно-методических материалов

1. Таненбаум Э. Архитектура компьютера. 5-е изд. СПб.: Питер, 2007.
2. Подбельский В. В. Язык Си++. 5-е изд. М.: Финансы и статистика, 2007.
3. Петцольд Ч. Код. Тайный язык информатики. М.: Издательско-торговый дом «Русская Редакция», 2001.

4. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2003.
5. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. 3-е изд. М.: Вильямс, 2013.
6. Керниган Б., Пайк Р. Практика программирования. М.: Вильямс, 2004.
7. Прата С. Язык программирования С. Лекции и упражнения, 5-е издание. М.: Вильямс, 2013.
8. Вирт Н. Алгоритмы и структуры данных. М.: ДМК Пресс, 2010.

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса по дисциплине используется среда для разработки программ Microsoft Visual Studio одной из последних версий. Обязателен компьютерный класс с доступом в Интернет.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Рекомендации по планированию и организации времени, отведенного на изучение дисциплины, описание последовательности действий обучающихся

Освоение дисциплины следует начинать с изучения рабочей учебной программы, которая содержит основные требования к знаниям, умениям и навыкам. Обязательно следует учитывать рекомендации преподавателя, данные в ходе установочных занятий. Затем – приступать к изучению отдельных разделов и тем в порядке, предусмотренном программой.

Получив представление об основном содержании раздела, темы, необходимо изучить материал с помощью рекомендуемой основной литературы. Целесообразно составить краткий конспект или схему, отображающую смысл и связи основных понятий данного раздела и включенных в него тем. Обязательно следует записывать возникшие вопросы, на которые не удалось ответить самостоятельно.

Подготовку к началу обучения включает несколько необходимых пунктов:

1) Необходимо создать для себя рациональный и эмоционально достаточный уровень мотивации к последовательному и планомерному изучению дисциплины.

2) Необходимо изучить список рекомендованной основной и дополнительной литературы и убедиться в её наличии у себя дома или в библиотеке в бумажном или электронном виде.

3) Необходимо иметь «под рукой» специальные и универсальные словари, справочники и энциклопедии, для того, чтобы постоянно уточнять значения используемых терминов и понятий. Пользование словарями и справочниками необходимо сделать привычкой. Опыт показывает, что неудовлетворительное усвоение предмета зачастую коренится в неточном, смутном или неправильном понимании и употреблении понятийного аппарата учебной дисциплины.

4) Желательно в самом начале периода обучения возможно тщательнее спланировать время, отводимое на работу с источниками и литературой по дисциплине, представить этот план в наглядной форме (график работы с датами) и в дальнейшем его придерживаться, не допуская срывов графика индивидуальной работы и «аврала» в предсессионный период. Пренебрежение этим пунктом приводит к переутомлению и резкому снижению качества усвоения учебного материала.

Рекомендации по работе с литературой

1) Всю учебную литературу желательно изучать «под конспект». Чтение литературы, не сопровождаемое конспектированием, даже пусть самым кратким – бесполезная работа. Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала. Эти навыки обязательны для любого

специалиста с высшим образованием независимо от выбранной специальности.

2) Написание конспекта должно быть творческим – нужно не переписывать текст из источников, но пытаться кратко излагать своими словами содержание ответа, при этом максимально структурируя конспект, используя символы и условные обозначения. Копирование и «заучивание» неосмысленного текста трудоемко и по большому счету не имеет большой познавательной и практической ценности.

3) При написании конспекта используется тетрадь, поля в которой обязательны. Страницы нумеруются, каждый новый вопрос начинается с нового листа, для каждого экзаменационного вопроса отводится 1-2 страницы конспекта. На полях размещается вся вспомогательная информация – ссылки, вопросы, условные обозначения и т.д.

4) В итоге данной работы «идеальным» является полный конспект по программе дисциплины, с выделенными определениями, узловыми пунктами, примерами, неясными моментами, проставленными на полях вопросами.

5) При работе над конспектом обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении установочных лекций и консультаций, либо в индивидуальном порядке.

6) При чтении учебной и научной литературы всегда следить за точным и полным пониманием значения терминов и содержания понятий, используемых в тексте. Всегда следует уточнять значения по словарям или энциклопедиям, при необходимости записывать.

7) При написании учебного конспекта обязательно указывать все прорабатываемые источники, автор, название, дата и место издания, с указанием использованных страниц.

Подготовка к промежуточной аттестации по дисциплине: экзамену (зачёту)

К аттестации допускаются студенты, которые систематически в течение всего семестра посещали и работали на занятиях и показали уверенные знания в ходе выполнения практических заданий и лабораторных работ.

Непосредственная подготовка к аттестации осуществляется по вопросам, представленным в рабочей учебной программе. Тщательно изучите формулировку каждого вопроса, вникните в его суть, составьте план ответа. Обычно план включает в себя:

- определение сущности рассматриваемого вопроса, основных положений, утверждений, определение необходимости их доказательства;
- запись обозначений, формул, необходимых для полного раскрытия вопроса;
- графический материал (таблицы, рисунки, графики), необходимые для раскрытия сущности вопроса;
- роль и значение рассматриваемого материала для практической деятельности, примеры использования в практической деятельности.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для осуществления образовательного процесса по дисциплине необходим компьютерный класс с персональными компьютерами с доступом в сеть «Интернет» и установленной средой для разработки программ Microsoft Visual Studio. В компьютерном классе должно быть оборудование мультимедийного типа (мультимедийный проектор, настенный экран) и пластиковая доска.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ
РАБОТЫ ОБУЧАЮЩИХСЯ**
по дисциплине «Теоретические основы компьютерных наук»
Направление подготовки 01.03.04 Прикладная математика
профиль «Математические методы в экономике»
Форма подготовки очная

Владивосток
2017

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	Модуль 1. 4 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций; самостоятельный разбор заданий и задач, решаемых на лабораторных работах; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением	18 часов	Собеседование, вопросы 1-7, блок I; контрольная работа 1
2	Модуль 1. 8 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций; самостоятельный разбор заданий и задач, решаемых на лабораторных работах; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным	18 часов	Собеседование, вопросы 1-7, блок I;

		обеспечением		
3	Модуль 1. 12 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций; самостоятельный разбор заданий и задач, решаемых на лабораторных работах; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением	18 часов	Собеседование, вопросы 1-7, блок II; Контрольная работа 2-3
3	Модуль 1. 18 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций; самостоятельный разбор заданий и задач, решаемых на лабораторных работах; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением	27 часов	Собеседование, вопросы блока III, VI; защита проекта, тема 2
4	Модуль 1. Сессия	Подготовка к экзамену	27 часов	Экзамен
5	Модуль 2. 4	Повторение теоретического и	2 часа	Собеседование,

	неделя	практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением		блок IV; защита проекта, тема 1; контрольная работа 4
6	Модуль 2. 8 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением	2 часа	Собеседование, блок V; защита проекта, тема 3; контрольная работа 5
7	Модуль 2. 12 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ,	2 часа	Собеседование, блок VI; защита проекта, темы 4-5; контрольная работа 6

		в том числе при работе со специальным программным обеспечением		
8	Модуль 2. 18 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций	3 часа	Собеседование, все вопросы ; защита проектов, темы 1-5
9	Модуль 2. Сессия	Подготовка к экзамену	27 часов	Экзамен

Характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению

Задания и методические рекомендации по теме «Теория информации»

Необходимо изучить следующий материал и ответить на вопросы по данному материалу.

Теория информации

Источник: Кадырова Г.Р. Курс лекций по информатике. Часть 1. Ульяновск: УлГТУ, 2008.

Одним из ключевых понятий информатики является информация. Для определения информации нам очень важно понять, что информация образуется из данных, но ее содержательная часть зависит не только оттого, какие сигналы были зарегистрированы при образовании данных, но и оттого, каким методом данные воспроизводятся. Данные — это зарегистрированные сигналы.

Методы воспроизведения и обработки данных можно разделить на естественные и технические.

Естественные методы воспроизведения данных присущи человеку и другим организмам живой природы. Если мы говорим о человеке, то прежде всего к естественным методам относим все методы, основанные на его органах чувств (зрение, осязание, обоняние, слух и вкус).

Благодаря зрению человек получает отпечаток окружающей среды на сетчатке глаза. Сигналы регистрируются нервными окончаниями сетчатки, в результате чего образуются данные, которые впоследствии анализируются головным мозгом. Результатом этого анализа является наблюдаемый образ, то есть информация.

Вам, конечно, знакома разница между внимательным и невнимательным наблюдением. И в том и в другом случае на сетчатке глаза образуются совершенно одинаковые данные, но информацию мы получаем разную. Это связано с тем, что при внимательном наблюдении мозг применяет более сложные методы обработки данных.

Хороший пример того, как из одних и тех же данных образуется разная информация, представляют собой стереограммы. Их следует рассматривать так, чтобы левый и правый глаз фокусировались в разных точках рисунка. В этом случае мозг обрабатывает данные иным методом, и вместо регулярного узора мы можем наблюдать скрытое объемное изображение.

Кроме методов, основанных на органах чувств, человек обладает и другими методами обработки данных. К ним относится логическое мышление, воображение, сравнение, сопоставление, анализ, прогнозирование и другие.

В связи с бурным развитием вычислительной техники в последние годы в классе технических методов четко выделились два направления: аппаратные и программные методы, способные во многих случаях подменять или дополнять друг друга.

Широкое внедрение средств вычислительной техники позволяет автоматизировать обработку самых разных видов данных с помощью компьютеров. Компьютер — это прибор особого типа, в котором одновременно сочетаются аппаратные и программные методы обработки и представления информации.

До сих пор мы определили только данные как результат регистрации сигналов. Определить, что такое информация, не столь просто, хотя бы потому, что она, в отличие от данных, не является объектом материальной природы и образуется в результате взаимодействия данных с методами.

Несмотря на то что понятие информации очень широко используется и в науке, и в повседневной жизни, его строгого научного определения до последнего времени не существовало. По сей день разные научные дисциплины вводят это понятие по-разному. Здесь можно выделить три возможных подхода: антропоцентрический, техноцентрический и недетерминированный.

Суть антропоцентрического подхода состоит в том, что информацию отождествляют со сведениями или фактами, которые теоретически могут быть получены и усвоены, то есть преобразованы в знания. Этот подход в настоящее время применяется наиболее широко.

Суть техноцентрического подхода состоит в том, что информацию отождествляют с данными. Этот подход нашел очень широкое распространение в технических дисциплинах.

Недетерминированный подход к понятию информации встречается также достаточно широко. Он состоит в отказе от определения информации на том основании, что оно является фундаментальным, как, например, материя и энергия.

Лишь в последние годы информатика начала формироваться как естественнонаучная дисциплина, но она еще не вышла за рамки прикладной технической науки и потому до сих пор не ввела строгого понятия информации.

Научное определение информации дается достаточно просто, если предположить, что информация — это динамический объект, не существующий в природе сам по себе, а образующийся в ходе взаимодействия данных и методов. Он существует ровно столько, сколько длится это взаимодействие, а все остальное время пребывает в виде данных.

Информация — это продукт взаимодействия данных и методов, рассмотренный в контексте этого взаимодействия.

В нашем определении важным является пояснение «... рассмотренный в контексте этого взаимодействия». Приведем примеры, почему это действительно важно.

Анализируя информационную ценность газет, журналов, телепередач, мы можем прийти к выводу, что она зависит как от данных, так и от методов, которыми выполняется их потребление. Одно дело — внимательно просматривать телефильм, вслушиваясь в каждое слово, и совсем другое — смотреть его, одновременно разговаривая по телефону.

Попробуйте проанализировать свое участие в учебных занятиях. Вы заметите, что фактор внимательности влияет на содержание информации, полученной из данных, которые излагает преподаватель. Проявляя внимательность, мы расширяем возможности естественных методов, основанных на органах чувств, за счет методов логического мышления.

Характерными чертами информации являются следующие:

Это наиболее важный ресурс современного производства: он снижает потребность в земле, труде, капитале, уменьшает расход сырья и энергии.

Вызывает к жизни новые производства.

Является товаром, причем продавец информации ее не теряет после продажи.

Придает дополнительную ценность другим ресурсам, в частности, трудовым. Действительно, работник с высшим образованием ценится больше, чем со средним.

Информация может накапливаться.

Как следует из определения, с информацией всегда связывают три понятия:

- источник информации – тот элемент окружающего мира, сведения о котором являются объектом преобразования;
- потребитель информации – тот элемент окружающего мира, который использует информацию;
- сигнал – материальный носитель, который фиксирует информацию для переноса ее от источника к потребителю.

Одной из важнейших разновидностей информации является информация экономическая; ее отличительная черта – связь с процессами управления коллективами людей, организаций.

Экономическая информация – это совокупность сведений, возникающих в процессе производственно-хозяйственной, коммерческой и финансовой деятельности.

Совокупность экономической информации структурно можно свести к определенному набору экономических показателей (ЭП).

Простой, элементарной составляющей единицей экономической информации является реквизит.

Реквизит – логически неделимый элемент (нельзя разделить на более мелкие единицы – буквы, цифры – без потери смысла) показателя, соотносимый с определенным свойством отображаемого информацией объекта или процесса.

Каждый ЭП состоит из одного реквизита-основания и одного или нескольких реквизитов-признаков.

Реквизит-основание характеризует количественную сторону объекта или процесса и определяет значение показателя; реквизиты-признаки характеризуют качественную сторону и определяют наименование показателя (идентифицируют показатель).

Из определения информации вытекает важное свойство ее динамичности.

Дело в том, что информация существует крайне непродолжительное время — ровно столько, сколько продолжается взаимодействие данных и методов во время ее создания, потребления или преобразования. Как только это взаимодействие завершается, мы опять имеем данные, но уже представленные в другой форме.

Информационный процесс — это всегда цикл образования информации из данных и немедленного ее сохранения в виде новых данных. Информация существует крайне непродолжительное время, но сам информационный процесс длится столько, сколько существуют носители данных, представляющие информацию. Исследуя сегодня египетские иероглифы, ученые продолжают информационный процесс, начатый несколько тысяч лет назад.

В вычислительной технике, как и везде, информационный процесс протекает в ходе взаимодействия данных и методов. Однако он имеет особенность, связанную с тем, что некоторые этапы происходят автоматически, без участия человека. В ходе этих этапов данные, представленные зарегистрированными сигналами, взаимодействуют как с аппаратными методами (компьютерами и другими устройствами), так и с программными методами (компьютерными программами).

При этом важной особенностью компьютерных программ является их двойственная природа. С одной стороны, они проявляют себя как методы, а с другой стороны — как данные.

Компьютерные программы могут существовать в двух фазах: в активной и пассивной. В активной фазе программа работает совместно с оборудованием, ее команды управляют процессором компьютера, который под их воздействием обрабатывает данные и взаимодействует с другим оборудованием.

В пассивной фазе компьютерная программа ничем от данных не отличается. Ее точно так же можно хранить, транспортировать по каналам связи, воспроизводить в виде печатного текста или экранного изображения. Ее можно даже обрабатывать другими программами. Программу, представленную как данные, можно редактировать, то есть изменять ее содержание.

Классификацию информации выполняют по нескольким основаниям (эта классификация не является строгой и может меняться):

По времени возникновения:

а) априорная — известна потребителю заранее, до получения сигнала;

б) апостериорная — становится известной потребителю после получения сигнала.

Так, получаемая сейчас читателем информация является априорной, если он освоил азы информатики в школе, и апостериорной в противном случае.

По стабильности:

а) переменная – отражает фактические характеристики источника информации. Может меняться.

б) постоянная – неизменная и многократно используемая в течение длительного периода времени. Строго говоря, и эта информация может меняться, но с гораздо меньшей частотой, которой можно пренебречь. Весь понятийный аппарат, излагаемый по тексту, относится к постоянной информации и является тем понятийным базисом, который позволяет специалистам говорить на одном профессиональном языке.

По способу использования:

а) вспомогательная – необязательные данные;

б) закрытая – ее использование возможно с согласия определенных физических или юридических лиц;

в) избыточная – дублирует данные;

г) коммерческая – является объектом купли-продажи.

Будучи объектом преобразования и использования, информация характеризуется следующими свойствами:

- синтаксис – свойство, определяющее способ представления информации на носителе (в сигнале). Так, данная информация представлена на бумажном носителе с помощью определенного шрифта. Здесь же можно рассматривать такие параметры представления информации, как стиль и цвет шрифта, его размеры, формат бумаги и ее качество и т.д. Выделение нужных параметров как синтаксических свойств, очевидно, определяется предполагаемым способом преобразования. Например, для плохо видящего человека существенным является размер и цвет шрифта. Если предполагается вводить данный текст в компьютер через сканер, важен формат бумаги.

- семантика – свойство, определяющее смысл информации как соответствие сигнала реальному миру. Так, семантика сигнала «информатика» заключается в данном ранее определении. Семантика может рассматриваться как некоторое соглашение, известное потребителю информации, о том, что означает каждый сигнал (так называемое правило интерпретации). Например, именно семантику сигналов изучает начинающий автомобилист, штудирующий правила дорожного движения, познавая дорожные знаки (в этом случае сигналами выступают сами знаки). Семантику слов (сигналов) познаёт обучаемый какому-либо иностранному языку.

- прагматика – свойство, определяющее влияние информации на поведение потребителя.

Следует отметить, что различные по синтаксису сигналы могут иметь одинаковую семантику. Например, сигналы «ЭВМ» и «компьютер» означают электронное устройство для преобразования информации. В этом случае обычно говорят о синонимии сигналов. С другой стороны, один сигнал (т. е.,

информация с одним синтаксическим свойством) может иметь разную прагматику для потребителей и семантику. Так, дорожный знак, известный под названием «кирпич» и имеющий вполне определенную семантику («въезд запрещен»), означает для автомобилиста запрет на въезд, а на пешехода никак не влияет. В то же время, сигнал «ключ» может иметь разную семантику: скрипичный ключ, родниковый ключ, ключ для открытия замка или гаечный ключ (в этом случае говорят об омонимии сигнала).

Задания и методические рекомендации по темам «Работа с указателями и массивами», «Адресная арифметика», «Технология разработки программ», «Отладка и тестирование программного обеспечения»

Указатели и адреса

1. Указатели на действительные переменные

1.1. Размещение значений переменных и их указателей в памяти

```
#include <iostream>
int main() {
    int x = 10;
    int *px = &x;
    ... // отобразить x, используя x и px
    ... // изменить x и отобразить его значение, используя *px
    ... // изменить *px и отобразить его значение, используя x
    cout << " x содержит " << x << endl;
    return 0;
}
```

Задания

1. Дополните данную программу. Что такое переменная и указатель? Какими характеристиками и отличиями обладают?
2. Организуйте вывод содержимого указателя px, т.е. адреса переменной x двумя способами:
 - а) используя имя px;
 - б) используя обозначение адреса x: &x.

3. Что такое сегментный адрес и смещение? Начертите схему размещения в памяти переменной x и указателя px побайтно. Укажите их адреса и содержимое.

1.2. Указатели на указатели

```
#include <iostream>
(?) main() {
    int i = 88;
    int *pi = &i;
    int **ppi = &pi;
    int ***pppi = &ppi;
    cout << "\n ***pppi = " << ***pppi;
}
```

Задания

1. Объясните, как работает данная "конструкция" из указателей.
2. Начертите распределение указателей в памяти компьютера, их адреса, и что в них должно быть записано.
3. Что произойдет, если при выводе убрать одну из *?

-

1.3. Доступ к данным с использованием указателей

```
#include <iostream>
main ()
{
    unsigned long L = 0x12345678L;
    char *cp = (char *)&L;
    int *ip = (int *)&L;
    long *lp = (long *)&L;
    cout << hex;
    cout << "\Адрес L, т.е. &L = " << ...;
    cout << "\ncp = " << (void *)cp << "\t*cp = 0x" <<
(int *)cp;
    cout << "\nip = " << (void *)ip << "\t*ip = 0x" << *ip;
```

```

cout << "\nlp = " << (void *)lp << "\t*lp = 0x" << *lp;
return 0;
}

```

Задания

1. Дополните и объясните данную программу и результаты выведенных значений. Объясните лексему 0x12345678L.
2. Начертите схему размещения указателей, а также распределение цифр числа L по байтам памяти и сопоставьте им адреса.
3. Каково назначение типа void? Объясните применение операции преобразования типа (void*).
4. Примените вместо манипулятора hex манипулятор oct либо dec (последний применяется по умолчанию). Объясните результат.
5. Выведите с помощью указателя sr и поразрядной операции сдвига сочетание цифр 56 числа L (цифры второго байта числа L).

2. Вычитание указателей

Создайте программу, выводящую разности между двумя символьными и целочисленными значениями, а также между указателями на них в адресной и числовой форме.

Задания

1. Выведите адреса всех указателей и начертите схему распределения в памяти компьютера указателей и переменных в соответствии с их адресами.
2. В чем отличие адресной и числовой разности?

3. Вычитание адресов и указателей разных типов

```

#include <iostream>
(?) main() {
    double aa = 0.0, bb = 1.0;
    double *pda = &aa, *pdb = &bb;
    float *pfa = (float *)&aa, *pfb = (float *)&bb;
    int *pia = (int *)&aa, *pib = (int *)&bb;
    char *pca = (char *)&aa, *pcb = (char *)&bb;
    cout << "\nАдреса объектов: &aa = " << &aa << "\t&bb = " << &bb;
    cout << "\nРазность адресов: (&bb - &aa) = " << (&bb - &aa);
}

```

```

cout << "\nРазность значений адресов: " << "((int)&bb - (int)&aa) = "
    << ((int)&bb - (int)&aa);
cout << "\nРазности указателей:";
cout << "\n double *: (pdb - pda) = " << (pdb - pda);
cout << "\n float *: (pfb - pfa) = " << (pfb - pfa);
cout << "\n int *: (pib - pia) = " << (pib - pia);
cout << "\n char *: (pcb - pca) = " << (pcb - pca);
return 0;
}

```

Задания

1. Объясните действие программы.
2. Примените операцию **sizeof** к переменным и указателям разных типов.
3. Почему разности указателей отрицательны?

4. Изменение указателя на произвольную величину

```

#include <iostream>
int main() {
    long L1 = 12345678;
    int i = 6;
    double d = 66.6;
    long L2 = 87654321;
    cout << "\nНе кратные для long адреса: " << ...;
    cout << "\nРазность некратных адресов: " << ...;
    cout << "\n(&L2 + 5) = " << ...;
    int *pi;
    long *pl = &L1;
    cout << ...; // вывести значение и адрес pl
    pl = (long *)((long&L1 - sizeof(int) - sizeof(double) - sizeof(long)); // (1)
    cout << "\npl = " << pl << " *pl = " << *pl;
    pi = (int *)((long&L2 + sizeof(long) + sizeof(double)); // (2)
    cout << "\npi = " << pi << " *pi = " << *pi;
    return 0;
}

```

```
}
```

Задания

1. Дополните и объясните эту программу. Объясните понятие кратных и некратных адресов.
2. Начертите схему распределения объявленных переменных в памяти компьютера побайтно, с указанием их адресов. На ее примере объясните, как действуют следующие операторы: (1) и (2).

Массивы

1. Одномерные массивы

1.1. Обращение к элементам массива по номеру элемента

```
#include <iostream>
const int MAX = 10;
int main() {
    double array[MAX] = { /* список из 10-ти значений элементов массива */ };
    int num_elem = MAX;
    double sum = 0;
    for (int ix = 0; ix < num_elem; ix++)
    {
        /* найти сумму элементов массива и вывести сам элемент */
    }
    cout << endl << "Average: " << sum / num_elem << endl;
    return 0;
}
```

Задания

1. Дополните и запустите программу на своем компьютере. Что произойдет, если задать значение формальной константы **MAX**:
 - а) больше 10;
 - б) меньше 10.
2. Перестройте программу для ввода элементов массива с клавиатуры.
3. Если элементы массива $p[4]$ равны соответственно $\{4, 2, 1, 3\}$, то чему равно значение выражения $p[p[4] - 2 + p[5 - p[2]]]$?

1.2. Обращение к элементам массива через операцию *разыменования*

```
#include <iostream>
```

```

void main()
{ int m[] = {40, 30, 20, 10};
  int j = 1;
  cout << "\nm[j] = " << m[j];
  cout << " *(m + j++) = " << *(m + j++);
  cout << "\n(++j + m) = " << *(++j + m);
  cout << " j[m] = " << j[m];
  cout << "\n*(j-- + m) = " << *(j-- + m);
  cout << " j--[m] = " << j--[m];
  cout << "\n*(-j + m) = " << *(-j + m);
  cout << " --j[m] = " << --j[m];
  cout << "\n3[m] = " << 3[m] << "\n2[m] = " << 2[m]
      << "\n1[m] = " << 1[m] << "\n0[m] = " << 0[m] ;
}

```

Задания

1. Запустите программу. Объясните результаты.
2. В чем отличие операций `j--[m]` и `--j[m]`?
3. Будет ли выполняться команда `*(++m)`? Какими свойствами обладает имя массива?

2. Двумерные массивы

```

#include <iostream>
const int MAX_COL = 3;
#define MAX_ROW 4
int main() {
    double matrix[MAX_ROW][MAX_COL] = {11.11, 2.2, 3, 4, 5.5, 6, 7.7, 8, 9.9,
4};
    double sum, average;
    int rows = MAX_ROW, cols = MAX_COL;
    cout << " Матрица :" << endl;
    for( int i = 0; i < rows; i++ )
    {
        for( int j = 0; j < cols; j++ )

```

```

{
cout.width(4);
cout.precision(1);
cout << matrix[i][j] << " ";
}
cout << endl;
}
cout << endl;
// Вычислить среднее значение по каждому столбцу.
for( ... )
    {
sum = 0;
for( ... )
sum += ....;
average = ....;
cout << " Среднее для столбца " << j << " = "
        << average << endl;
}
return 0;
}

```

Задания

1. Дополните программу недостающим текстом так, чтобы она рассчитывала и выводила на экран среднее значение по столбцам.
2. Объясните использование функций **width()** и **precision()**.
3. Дан некоторый двумерный массив `matrix[M][M]`. Какое условие позволит определить элементы, кратные 5 и расположенные выше главной диагонали?
4. (Доп.) Составить алгоритм нахождения числа строк двумерного массива `matrix[M][M]`, количество отрицательных элементов в которых больше заданной величины **a**.
5. (Доп.) Сгенерировать двумерный массив размером 5*3. Поменять столбец с максимальным элементом с последним столбцом массива.

Типовые задачи

1. Запишите результат работы данной программы:

```
#include <iostream>
using namespace std;
void main()
{
    int i = 5, j = 10;
    double x;
    x = 1. / 5 * j++;
    x--;
    cout << " x = " << x << endl << "\n";
    x = --i % 2 + 30./2;
    cout << " x = " << x << "\n" << endl;
}
```

2. Запишите результат работы данной программы:

```
#include <iostream>
using namespace std;
void main()
{
    int i, j, k;
    k = 0;
    j = 1;
    i = j + 1;
    cout << (k ? "Yes!" : "No(") << endl;
    (i == 2 && j == 1) ? cout << "Yes!" : cout << "No(" << endl;
}
```

3. а) Перевести из десятичной системы в двоичную число (101,7).

б) $(101,01)_2 = (?)_{10}$.

в) $(1111)_2 = (?)_8$.

4. Преобразуйте следующие числа в формат стандарта IEEE с одинарной точностью. Результаты представьте в восьми шестнадцатеричных разрядах: 10, 4/17, -6.

5. Создайте программу калькулятора, позволяющего производить арифметические действия и поразрядные операции.

С клавиатуры пользователь вводит два числа и знак операции. Для введенного символа операции программа должна произвести выбор соответствующего арифметического действия. Результат выводится на экран в виде:

введенное число 1 __ знак операции __ введенное число 2 __ знак равенства __ результат

6. Масса одной молекулы воды приблизительно равна 3.0×10^{-23} грамм. Кварта воды весит примерно 950 грамм. Напишите программу, которая приглашает ввести некоторое значение объема воды в квартах и отображает количество молекул воды в этом объеме.

7. Напишите программу, выполняющую сортировку одномерного массива одним из указанных способов: прямой выбор, пузырьковая сортировка, шейкерная сортировка, сортировка простым слиянием, сортировка Шелла, быстрая сортировка.

8. Продемонстрировать работу с динамическими массивами и объяснить, в каких случаях они применяются.

9. Составьте программу вычисления объема правильного многогранника (конкретный вид задан), опираясь на соответствующие

измерения (длину, ширину, глубину, высоту и т.д.), используя только функции и передавая все значения в них только через указатели.

10. Составьте программу вычисления суммы заданного числового ряда.

11. Напишите программу, выполняющую по выбору пользователя одну из указанных операций: сложение, вычитание или умножение двух заданных матриц. Размерности матриц, значения их элементов и знак операции должны вводиться с клавиатуры. Вывод на экран должен содержать введенные матрицы и результат применения указанной пользователем операции.

12. Составьте программу, работающую с динамическими структурами данных (стеком, очередью, списком или деревом).

Требования к представлению и оформлению результатов самостоятельной работы

Самостоятельная работа включает в себя повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам занятий; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением — средой разработки для языка программирования C++.

Результаты самостоятельной работы представляются и оформляются в виде ответов на основные положения теоретического и практического материала дисциплины по темам; работающего исходного кода результата решения практических заданий и задач; собственных действий, осуществляемых в ходе выполнения лабораторных работ.

В случае подготовки слайдов для защиты проекта, они должны быть контрастными (рекомендуется черный цвет шрифта на светлом фоне), кегль текста слайдов – не менее 22pt, заголовков – 32pt. Основная цель использования слайдов - служить вспомогательным инструментом к

подготовленному выступлению, цитирование больших фрагментов текста на слайдах не допускается. Приветствуется использование рисунков, графиков, таблиц, интерактивного материала, однако, следует предусмотреть выбор цвета и толщину линий.

Слайды должны содержать титульный лист, цели и задачи (не более 2-х слайдов с обзором актуальности, новизны, теоретической и практической значимости работы), основные публикации с их кратким обзором (1-2 слайда), формальную постановку задачи и формулировку моделей (1-2 слайда), краткое тезисное (!) изложение ключевых положений работы (разумное количество слайдов с учетом общего времени выступления), заключение (с изложением результатов работы, подведением выводов, обсуждением практического использования работы, возможностей проведения дальнейших исследований и разработок в данной области).

Как правило, 12-15 слайдов оказывается достаточным для полного представления работы.

Критерии оценки выполнения самостоятельной работы

Общие критерии оценки выполнения самостоятельной работы – правильность ответов на вопросы по темам теоретической части дисциплины, верность получаемых ответов в ходе решения практических заданий и задач, достижение правильного результата при осуществлении собственных действий по лабораторным работам.

Оценивание знаний в форме собеседования проводится по критериям:

- логичность изложения, знание и понимание основных теоретических и практических положений и концепций по теме;
- владение теоретическими и практическими приёмами написания, отладки и тестирования программ на языке программирования Python.

Оценивание знаний в форме защиты проекта проводится по критериям:

- завершенность и полнота выполненных заданий в рамках данного проекта;
- владение методами и приемами решения конкретных задач и самостоятельность использования специализированного программного обеспечения;
- качество оформления отчёта в соответствии с правилами и стандартами оформления документации разработанных программ.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине «Теоретические основы компьютерных наук»
Направление подготовки 01.03.04 Прикладная математика
профиль «Математические методы в экономике»
Форма подготовки очная

Владивосток
2017

**Паспорт
фонда оценочных средств
по дисциплине «Теоретические основы компьютерных наук»**

Код и формулировка компетенции	Этапы формирования компетенции	
ОК-5 - способностью использовать современные методы и технологии (в том числе информационные) в профессиональной деятельности	Знает	современные языки программирования (в частности, языки С и С++)
	Умеет	применять знания современных языков программирования для разработки программных продуктов
	Владеет	навыками разработки программ на современных языках программирования
ОПК-2 - способность использовать современные математические методы и современные прикладные программные средства и осваивать современные технологии программирования	Знает	современные технологии программирования на языке С++
	Умеет	разрабатывать программные продукты с помощью современных технологий программирования
	Владеет	эффективными методами разработки программных продуктов с помощью современных технологий программирования
ПК-11 – готовность применять знания и навыки управления информацией	Знает	основные принципы управления информацией
	Умеет	применять знания основных принципов управления информацией
	Владеет	навыками управления информацией

№ п/п	Контролируемые разделы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
10	Аппаратные средства информатики	ОК-5	Знает	Собеседование (УО-1)	Экзамен, вопросы 1-7
			Умеет	Проект (ПР-9)	Проект, 1-4
			Владеет	Проект (ПР-9)	Проект, 1-4
		ОПК-2	Знает	Собеседование (УО-1)	Экзамен, вопросы 1-7
			Умеет	Проект (ПР-9)	Проект, 1-4
			Владеет	Проект (ПР-9)	Проект, 1-4
		ПК-11	Знает	Собеседование (УО-1)	Экзамен, вопросы 1-7

			Умеет	Проект (ПР-9)	Проект, 1-4		
			Владеет	Проект (ПР-9)	Проект, 1-4		
		ПК-11	Знает	Собеседование (УО-1)	Экзамен, вопросы 1-7		
			Умеет	Проект (ПР-9)	Проект, 1-4		
			Владеет	Проект (ПР-9)	Проект, 1-4		
2	Основы языков программирования	ОК-5	Знает	Собеседование (УО-1)	Экзамен, вопросы 8-19		
			Умеет	Проект (ПР-9)	Проект, 1-4		
			Владеет	Проект (ПР-9)	Проект, 1-4		
		ОПК-2	Знает	Собеседование (УО-1)	Экзамен, вопросы 8-19		
			Умеет	Проект (ПР-9)	Проект, 1-4		
			Владеет	Проект (ПР-9)	Проект, 1-4		
		ПК-11	Знает	Собеседование (УО-1)	Экзамен, вопросы 8-19		
			Умеет	Проект (ПР-9)	Проект, 1-4		
			Владеет	Проект (ПР-9)	Проект, 1-4		
		ПК-11	Знает	Собеседование (УО-1)	Экзамен, вопросы 8-19		
			Умеет	Проект (ПР-9)	Проект, 1-4		
			Владеет	Проект (ПР-9)	Проект, 1-4		
		3	Адресная арифметика и массивы	ОК-5	Знает	Собеседование (УО-1)	Экзамен, вопросы 20-24
					Умеет	Проект (ПР-9)	Проект, 5
					Владеет	Проект (ПР-9)	Проект, 5
ОПК-2	Знает			Собеседование (УО-1)	Экзамен, вопросы 20-24		
	Умеет			Проект (ПР-9)	Проект, 5		
	Владеет			Проект (ПР-9)	Проект, 5		
ПК-11	Знает			Собеседование (УО-1)	Экзамен, вопросы 20-24		
	Умеет			Проект (ПР-9)	Проект, 5		
	Владеет			Проект (ПР-9)	Проект, 5		
ПК-11	Знает			Собеседование (УО-1)	Экзамен, вопросы 20-24		
	Умеет			Проект (ПР-9)	Проект, 5		
	Владеет			Проект (ПР-9)	Проект, 5		

Модуль 2

1	Технология разработки	ОК-5	Знает	Собеседование (УО-1)	Экзамен, вопросы 1-16
---	-----------------------	------	-------	----------------------	-----------------------

	программ		Умеет	Проект (ПР-9)	Проект, 1-2		
			Владеет	Проект (ПР-9)	Проект, 1-2		
		ОПК-2	Знает	Собеседование (УО-1)	Экзамен, вопросы 1-16		
			Умеет	Проект (ПР-9)	Проект, 1-2		
			Владеет	Проект (ПР-9)	Проект, 1-2		
		ПК-11	Знает	Собеседование (УО-1)	Экзамен, вопросы 1-16		
			Умеет	Проект (ПР-9)	Проект, 1-2		
			Владеет	Проект (ПР-9)	Проект, 1-2		
		ПК-11	Знает	Собеседование (УО-1)	Экзамен, вопросы 1-16		
			Умеет	Проект (ПР-9)	Проект, 1-2		
			Владеет	Проект (ПР-9)	Проект, 1-2		
		2	Элементы языка C/C++ и алгоритмы сортировки	ОК-5	Знает	Собеседование (УО-1)	Экзамен, вопросы 17-47
					Умеет	Проект (ПР-9)	Проект, 3
					Владеет	Проект (ПР-9)	Проект, 3
				ОПК-2	Знает	Собеседование (УО-1)	Экзамен, вопросы 17-47
Умеет	Проект (ПР-9)				Проект, 3		
Владеет	Проект (ПР-9)				Проект, 3		
ПК-11	Знает			Собеседование (УО-1)	Экзамен, вопросы 17-47		
	Умеет			Проект (ПР-9)	Проект, 3		
	Владеет			Проект (ПР-9)	Проект, 3		
ПК-11	Знает			Собеседование (УО-1)	Экзамен, вопросы 17-47		
	Умеет			Проект (ПР-9)	Проект, 3		
	Владеет			Проект (ПР-9)	Проект, 3		
3	Динамические структуры данных			ОК-5	Знает	Собеседование (УО-1)	Экзамен, вопросы 48-71
					Умеет	Проект (ПР-9)	Проект, 4
					Владеет	Проект (ПР-9)	Проект, 4
		ОПК-2	Знает	Собеседование (УО-1)	Экзамен, вопросы 48-71		
			Умеет	Проект (ПР-9)	Проект, 4		
			Владеет	Проект (ПР-9)	Проект, 4		
		ПК-11	Знает	Собеседование (УО-1)	Экзамен, вопросы 48-71		
			Умеет	Проект (ПР-9)	Проект, 4		
			Владеет	Проект (ПР-9)	Проект, 4		
		ПК-11	Знает	Собеседование (УО-1)	Экзамен, вопросы 48-71		
			Умеет	Проект (ПР-9)	Проект, 4		
			Владеет	Проект (ПР-9)	Проект, 4		

Зачётно-экзаменационные материалы

Вопросы для подготовки к экзамену по дисциплине «Теоретические основы компьютерных наук»

Модуль 1

1. Понятие компьютера. Представление информации в компьютере. Устройство и принципы работы компьютера. Различные типы компьютеров. IBM PC-совместимые компьютеры.
2. Основные и дополнительные блоки ПК. Устройства, подключаемые к компьютеру. Указательные устройства. Дисководы. Принтеры, сканеры, средства мультимедиа.
3. Микропроцессор и сопроцессор. Оперативная память. Кэш-память.
4. Материнская плата, контроллеры, шины.
5. Мониторы и видеоконтроллеры. Видеоинформация.
6. Жесткие диски и другие устройства хранения данных.
7. Устройства передачи данных. Локальные сети. Интернет.
8. Алфавит и лексемы языков программирования. Идентификаторы и служебные слова.
9. Символьный тип данных. Таблица ASCII. Массив символов.
10. Строковые константы. Особенности и примеры их применения.
11. Числовые целые константы. Особенности и примеры их применения.
12. Числовые вещественные константы. Особенности и примеры их применения.
13. Операция присваивания. Пример линейной программы.
14. Синтаксис и семантика операторов выражения, возврата, ветвления и выбора. Пример программы ветвления.
15. Синтаксис и семантика операторов цикла, прерывания и продолжения. Пример циклической программы.

- 16.Классификация типов данных. Основные и производные типы данных.
Правила преобразования арифметических типов: по умолчанию и явное.
- 17.Арифметические операции и их приоритеты.
- 18.Логические операции. Синтаксис и семантика. Условная операция.
- 19.Виды выражений. Приоритеты операций.
- 20.Одномерные массивы. Индексация выражения.
- 21.Многомерные массивы. Индексация выражения.
- 22.Массивы с элементами типа char и строковые массивы.
- 23.Указатели: понятие, синтаксис и семантика объявления. Операция выделения значения. Адрес-константа. Операции адресации, адресная арифметика.
- 24.Взаимодействие между массивами и указателями.

Вопросы для подготовки к экзамену
по дисциплине «Теоретические основы компьютерных наук»

Модуль 2

I. Элементы технологии программирования

1. Формальная постановка задачи. Анализ предметной области (бизнес-моделирование).
2. Алгоритм. Блок-схема. Примеры.
3. Программа. Программное обеспечение (ПО). Классификация ПО.
4. Технология программирования. Технологическая операция.
5. Основные этапы развития технологии программирования.
6. Проблемы разработки сложных программных систем.
7. Блочнo-иерархический подход к созданию сложных систем.
8. Декомпозиция предметной области. Методы декомпозиции.
9. Жизненный цикл и этапы разработки программного обеспечения.

10. Анализ требований. Спецификации. Проектирование.
11. Эволюция моделей жизненного цикла ПО. Каскадная модель и модель с промежуточным контролем.
12. Спиральная модель жизненного цикла ПО.
13. Тестирование. Цель тестирования. Зависимость вероятности правильного исправления и его стоимости от этапа разработки.
14. Стратегии тестирования. Структурный подход.
15. Ручной контроль ПО.
16. Функциональное тестирование. Эквивалентное разбиение. Анализ граничных значений.

II. Элементы языка C/C++

17. Алфавит и лексемы языка C++. Идентификаторы и служебные слова.
18. Символьные и строковые константы. Разница между ними. ESC-последовательности.
19. Расширенная таблица ASCII. Пример программы.
20. Целочисленные типы данных. Особенности и примеры их применения. Представление в памяти.
21. Типы данных с плавающей точкой. Особенности и примеры их применения. Представление в памяти.
22. Оператор присваивания. Пример линейной программы.
23. Синтаксис и семантика операторов выражения, возврата, ветвления и выбора. Пример программы ветвления.
24. Синтаксис и семантика операторов цикла, прерывания и продолжения. Пример циклической программы.
25. Классификация типов данных. Основные и производные типы данных. Правила преобразования арифметических типов: по умолчанию и явное.
26. Арифметические операции и их приоритеты.
27. Логические операции. Синтаксис и семантика. Условная операция.
28. Поразрядные целочисленные операции.
29. Виды выражений. Приоритеты операций.

30. Символьный тип данных. Операции над данными символьного типа.
31. Одномерные массивы. Индексация выражения. Массив как параметр функций.
32. Многомерные массивы. Индексация выражения. Массив как параметр функций.
33. Массивы с элементами типа `char` и тип `string`. Разница между ними.
34. Указатели: понятие, синтаксис и семантика объявления. Операция разыменования. Типы указателей.
35. Адресная арифметика.
36. Взаимодействие между массивами и указателями. Массивы указателей. Примеры.
37. Динамические массивы. Операции выделения и освобождения памяти.
38. Тип данных «структура». Объявление, синтаксис и особенности использования.
39. Массивы структур. Пример.
40. Тип данных «ссылка». Объявление, синтаксис и особенности использования.
41. Функции. Объявление, определение, сигнатура, тело функции. Параметры функции. Примеры.
42. Функции. Способы и особенности передачи значений параметров. Примеры.
43. Указатель на функцию. Примеры использования.
44. Оператор `typedef`. Особенности применения.
45. Глобальные и локальные переменные. Область видимости.
46. Чем плох оператор `goto`? Мнение Н. Вирта.
47. История создания языка C++ и его связь с языком C. Почему язык так назван?

III. Алгоритмы сортировки данных

48. Сортировка «простой выбор». Примеры.
49. Пузырьковая сортировка. Примеры.
50. Шейкерная сортировка. Примеры.
51. Сортировка Шелла. Примеры.

52. Быстрая сортировка. Примеры.

53. Метод суммирующего множителя.

54. Полный анализ быстрой сортировки с помощью метода суммирующего множителя.

IV. Динамические структуры данных

55. Список. Основные операции над элементами списка. Реализация на С. Примеры.

56. Очередь. Основные операции над элементами очереди. Реализация на С. Примеры.

57. Стек. Основные операции над элементами стека. Реализация на С. Примеры.

58. Хеширование. Хеш-функции. Коллизии. Метод деления. Метод умножения.

59. Хеш-функции для строк. Пример кода.

60. Хеш-таблицы. Метод цепочек.

61. Хеш-таблицы. Метод открытой адресации.

62. Деревья. Определение. Примеры. Число вершин и рёбер.

63. Двоичные, n-арные деревья. Реализация.

64. Обход двоичного дерева в глубину. Прямой, обратный, поперечный.

65. Обход двоичного дерева в ширину.

66. Двоичное дерево поиска. Операции: поиск по ключу, вставка, удаление узла, поиск минимального ключа.

V. Методы построения, анализа алгоритмов, теория трансляции программ

67. Компилируемые и интерпретируемые языки программирования. Разница. Примеры.

68. Основные фазы компиляции. Лексический анализатор. Синтаксический анализатор.

69. Жадные алгоритмы.

70. Понятие рекурсии. Примеры использования.

71. O-символика для оценки быстродействия. Примеры.

Комплекты оценочных средств для текущей аттестации

Вопросы для собеседования

по дисциплине «Теоретические основы компьютерных наук»

I. Теория информации

1. Понятие информации.
2. Информационные революции в истории развития человечества.
3. Информационный потенциал общества.
4. Информационные ресурсы и услуги.
5. Классификация информации.
6. Системы кодирования информации.
7. Меры информации.

II. Аппаратные средства информатики

1. Понятие компьютера. Представление информации в компьютере. Устройство и принципы работы компьютера. Различные типы компьютеров. IBM PC-совместимые компьютеры.
2. Основные и дополнительные блоки ПК. Устройства, подключаемые к компьютеру. Указательные устройства. Дисководы. Принтеры, сканеры, средства мультимедиа.
3. Микропроцессор и сопроцессор. Оперативная память. Кэш-память.
4. Материнская плата, контроллеры, шины.
5. Мониторы и видеоконтроллеры. Видеоинформация.
6. Жесткие диски и другие устройства хранения данных.
7. Устройства передачи данных. Локальные сети. Интернет.

III. Элементы языка C/C++

1. Алфавит и лексемы языка C++. Идентификаторы и служебные слова.
2. Символьные и строковые константы. Разница между ними. ESC-последовательности.
3. Расширенная таблица ASCII. Пример программы.

4. Целочисленные типы данных. Особенности и примеры их применения. Представление в памяти.
5. Типы данных с плавающей точкой. Особенности и примеры их применения. Представление в памяти.
6. Оператор присваивания. Пример линейной программы.
7. Синтаксис и семантика операторов выражения, возврата, ветвления и выбора. Пример программы ветвления.
8. Синтаксис и семантика операторов цикла, прерывания и продолжения. Пример циклической программы.
9. Классификация типов данных. Основные и производные типы данных. Правила преобразования арифметических типов: по умолчанию и явное.
10. Арифметические операции и их приоритеты.
11. Логические операции. Синтаксис и семантика. Условная операция.
12. Поразрядные целочисленные операции.
13. Виды выражений. Приоритеты операций.
14. Символьный тип данных. Операции над данными символьного типа.
15. Одномерные массивы. Индексация выражения. Массив как параметр функций.
16. Многомерные массивы. Индексация выражения. Массив как параметр функций.
17. Массивы с элементами типа `char` и тип `string`. Разница между ними.
18. Указатели: понятие, синтаксис и семантика объявления. Операция разыменования. Типы указателей.
19. Адресная арифметика.
20. Взаимодействие между массивами и указателями. Массивы указателей. Примеры.
21. Динамические массивы. Операции выделения и освобождения памяти.
22. Тип данных «структура». Объявление, синтаксис и особенности использования.
23. Массивы структур. Пример.

24. Тип данных «ссылка». Объявление, синтаксис и особенности использования.
25. Функции. Объявление, определение, сигнатура, тело функции. Параметры функции. Примеры.
26. Функции. Способы и особенности передачи значений параметров. Примеры.
27. Указатель на функцию. Примеры использования.
28. Оператор typedef. Особенности применения.
29. Глобальные и локальные переменные. Область видимости.
30. Чем плох оператор goto? Мнение Н. Вирта.
31. История создания языка C++ и его связь с языком C. Почему язык так назван?

IV. Элементы технологии программирования

1. Формальная постановка задачи. Анализ предметной области (бизнес-моделирование).
2. Алгоритм. Блок-схема. Примеры.
3. Программа. Программное обеспечение (ПО). Классификация ПО.
4. Технология программирования. Технологическая операция.
5. Основные этапы развития технологии программирования.
6. Проблемы разработки сложных программных систем.
7. Блочный-иерархический подход к созданию сложных систем.
8. Декомпозиция предметной области. Методы декомпозиции.
9. Жизненный цикл и этапы разработки программного обеспечения.
10. Анализ требований. Спецификации. Проектирование.
11. Эволюция моделей жизненного цикла ПО. Каскадная модель и модель с промежуточным контролем.
12. Спиральная модель жизненного цикла ПО.
13. Тестирование. Цель тестирования. Зависимость вероятности правильного исправления и его стоимости от этапа разработки.

14. Стратегии тестирования. Структурный подход.
15. Ручной контроль ПО.
16. Функциональное тестирование. Эквивалентное разбиение. Анализ граничных значений.

V. Алгоритмы сортировки данных

17. Сортировка «простой выбор». Примеры.
18. Пузырьковая сортировка. Примеры.
19. Шейкерная сортировка. Примеры.
20. Сортировка Шелла. Примеры.
21. Быстрая сортировка. Примеры.
22. Метод суммирующего множителя.
23. Полный анализ быстрой сортировки с помощью метода суммирующего множителя.

VI. Динамические структуры данных

24. Список. Основные операции над элементами списка. Реализация на С. Примеры.
25. Очередь. Основные операции над элементами очереди. Реализация на С. Примеры.
26. Стек. Основные операции над элементами стека. Реализация на С. Примеры.
27. Хеширование. Хеш-функции. Коллизии. Метод деления. Метод умножения.
28. Хеш-функции для строк. Пример кода.
29. Хеш-таблицы. Метод цепочек.
30. Хеш-таблицы. Метод открытой адресации.
31. Деревья. Определение. Примеры. Число вершин и рёбер.
32. Двоичные, n-арные деревья. Реализация.
33. Обход двоичного дерева в глубину. Прямой, обратный, поперечный.
34. Обход двоичного дерева в ширину.

35. Двоичное дерево поиска. Операции: поиск по ключу, вставка, удаление узла, поиск минимального ключа.

VII. Методы построения, анализа алгоритмов, теория трансляции программ

36. Компилируемые и интерпретируемые языки программирования. Разница. Примеры.

37. Основные фазы компиляции. Лексический анализатор. Синтаксический анализатор.

38. Жадные алгоритмы.

39. Понятие рекурсии. Примеры использования.

40. O-символика для оценки быстродействия. Примеры.

Критерии оценки:

✓ 100-86 баллов - если ответ показывает глубокое и систематическое знание всего программного материала и структуры конкретного вопроса, а также основного содержания и новаций лекционного курса по сравнению с учебной литературой. Студент демонстрирует отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей научной области. Знание основной литературы и знакомство с дополнительно рекомендованной литературой. Логически корректное и убедительное изложение ответа.

✓ 85-76 - баллов - знание узловых проблем программы и основного содержания лекционного курса; умение пользоваться концептуально-понятийным аппаратом в процессе анализа основных проблем в рамках данной темы; знание важнейших работ из списка рекомендованной литературы. В целом логически корректное, но не всегда точное и аргументированное изложение ответа.

✓ 75-61 - балл – фрагментарные, поверхностные знания важнейших разделов программы и содержания лекционного курса; затруднения с использованием научно-понятийного аппарата и терминологии учебной дисциплины; неполное знакомство с рекомендованной литературой; частичные затруднения с выполнением предусмотренных программой заданий; стремление логически определенно и последовательно изложить ответ.

✓ 60-50 баллов – незнание, либо отрывочное представление о данной проблеме в рамках учебно-программного материала; неумение использовать понятийный аппарат; отсутствие логической связи в ответе.

Итоговая шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

Темы проектов

по дисциплине «Теоретические основы компьютерных наук»

Модуль 1

1. Алгоритм. Блок-схемы.
2. Системы счисления.
3. Разработка программного обеспечения. Простые программы. Ввод-вывод информации в программе.
4. Организация памяти. Адресная арифметика.
5. Работа с указателями и массивами в Си++.

Модуль 2

1. Проектирование программного обеспечения.
2. Отладка и тестирование программного обеспечения.
3. Виды сортировок массива.
4. Динамические структуры данных. Реализация на С++.

Критерии оценки:

✓ 100-86 баллов - если ответ показывает глубокое и систематическое знание всего программного материала и структуры конкретного вопроса, а также основного содержания и новаций лекционного курса по сравнению с учебной литературой. Студент демонстрирует отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей научной области. Знание основной литературы и знакомство с дополнительно рекомендованной литературой. Логически корректное и убедительное изложение ответа.

✓ 85-76 - баллов - знание узловых проблем программы и основного содержания лекционного курса; умение пользоваться концептуально-понятийным аппаратом в процессе анализа основных проблем в рамках данной темы; знание важнейших работ из списка рекомендованной

литературы. В целом логически корректное, но не всегда точное и аргументированное изложение ответа.

✓ 75-61 балл – фрагментарные, поверхностные знания важнейших разделов программы и содержания лекционного курса; затруднения с использованием научно-понятийного аппарата и терминологии учебной дисциплины; неполное знакомство с рекомендованной литературой; частичные затруднения с выполнением предусмотренных программой заданий; стремление логически определенно и последовательно изложить ответ.

✓ 60-50 баллов – незнание, либо отрывочное представление о данной проблеме в рамках учебно-программного материала; неумение использовать понятийный аппарат; отсутствие логической связи в ответе.

Итоговая шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

Описание показателей и критериев оценивания компетенций, шкал оценивания

Критерии оценки собеседования

✓ 100-86 баллов - если ответ показывает глубокое и систематическое знание всего программного материала и структуры конкретного вопроса, а также основного содержания и новаций курса по сравнению с учебной литературой. Студент демонстрирует отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей научной области. Знание основной литературы и знакомство с дополнительно рекомендованной литературой. Логически корректное и убедительное изложение ответа.

✓ 85-76 - баллов - знание узловых проблем программы и основного содержания курса; умение пользоваться концептуально-понятийным аппаратом в процессе анализа основных проблем в рамках данной темы; знание важнейших работ из списка рекомендованной литературы. В целом логически корректное, но не всегда точное и аргументированное изложение ответа.

✓ 75-61 - балл – фрагментарные, поверхностные знания важнейших разделов программы и содержания курса; затруднения с использованием научно-понятийного аппарата и терминологии учебной дисциплины; неполное знакомство с рекомендованной литературой; частичные затруднения с выполнением предусмотренных программой заданий; стремление логически определенно и последовательно изложить ответ.

✓ 60-50 баллов – незнание, либо отрывочное представление о данной проблеме в рамках учебно-программного материала; неумение использовать понятийный аппарат; отсутствие логической связи в ответе.

Итоговая шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Текущая аттестация студентов. Текущая аттестация студентов по дисциплине «Теоретические основы компьютерных наук» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация по дисциплине «Теоретические основы компьютерных наук» проводится в форме собеседования и выполнения проектных работ, контрольных работ, лабораторных работ и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- степень усвоения теоретических знаний - оценивается в форме собеседования;
- уровень овладения практическими умениями и навыками – оценивается в форме выполнения лабораторных работ, контрольных работ, работы над проектами.

Промежуточная аттестация студентов. Промежуточная аттестация студентов по дисциплине «Теоретические основы компьютерных наук» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По дисциплине предусмотрен зачёт в 1 семестре и экзамен во 2 семестре, которые проводятся в устной форме.

Критерии выставления оценки студенту на зачёте/ экзамене по дисциплине «Теоретические основы компьютерных наук»

Баллы (рейтинговой оценки)	Оценка зачета/ экзамена (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и

		<p>приёмами выполнения лабораторных работ; знает современные технологии программирования на языке C++; умеет разрабатывать программные продукты с помощью современных технологий программирования; владеет эффективными методами разработки программных продуктов с помощью современных технологий программирования; знает технологии отладки и тестирования прикладного программного обеспечения; знает способы тестирования программных средств, в том числе, разработанных самостоятельно; умеет находить и исправлять ошибки в тестируемых программных продуктах; владеет навыками настройки вычислительной техники для работы в интегрированной среде разработки, навыками тестирования и проверки программных продуктов; знает современные языки программирования (в частности, язык C++); умеет применять знания современных языков программирования для разработки программных продуктов; владеет навыками разработки программ на современных языках программирования; имеет готовность применять знания и навыки управления информацией</p>
76-85	«зачтено»/ «хорошо»	<p>Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения; знает современные технологии программирования на языке C++; умеет разрабатывать программные продукты с помощью современных технологий программирования; владеет эффективными методами разработки программных продуктов с помощью современных технологий программирования; умеет находить и исправлять ошибки в тестируемых программных продуктах; владеет навыками настройки вычислительной техники для работы в интегрированной среде разработки; знает современные языки программирования (в частности, язык программирования C++); умеет применять знания современных</p>

		языков программирования для разработки программных продуктов; владеет навыками разработки программ на современных языках программирования.
61-75	<i>зачтено»/ «удовлетворительно»</i>	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении лабораторных работ; частично владеет навыками разработки программ на современных языках программирования.
0-60	<i>«не зачтено»/ «неудовлетворительно»</i>	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет лабораторные работы, практически не владеет навыками разработки программ на современных языках программирования. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.