

Балаковский инженерно-технологический институт – филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий
Кафедра «Информационные системы и технологии»

РАБОЧАЯ ПРОГРАММА
по дисциплине «Языки программирования»

Направление подготовки
«27.03.04 Управление в технических системах»

Основная профессиональная образовательная программа
«Управление и информатика в технических системах»

Квалификация выпускника
Бакалавр

Форма обучения
Заочная

Цель освоения дисциплины

Цель освоения дисциплины: формирование, развитие и совершенствование у студентов общепрофессиональных и профессиональных компетенций в области программирования на языках высокого и низкого уровня в соответствии с трудовыми функциями профессиональных стандартов: (Профессиональный стандарт «20.002. Работник по эксплуатации оборудования автоматизированных систем управления технологическими процессами гидроэлектростанции/гидроаккумулирующей электростанции»).

Задачи освоения дисциплины: изучение парадигм программирования, технологий разработки программ на языках высокого и низкого уровня, получение навыков применения теоретических знаний при разработке программных решений для устройств управления и автоматики.

Место учебной дисциплины в структуре ООП ВО

Требуемым условием для освоения дисциплины «Языки программирования» являются знания, умения и практические навыки, полученные при изучении курсов математики и информатики. Знания, полученные при освоении дисциплины «Языки программирования», необходимы при изучении дисциплин: Программирование микроконтроллеров, Информационные технологии в проектировании сложных систем, Робототехнические системы и комплексы, Экспертные системы, Интеллектуальные системы автоматического управления, при выполнении выпускной квалификационной работы, а также при прохождении производственной (технологической и преддипломной) практики.

В результате освоения дисциплины студент может частично продемонстрировать трудовую функцию в области эксплуатации АСУТП, связанную с выполнением работ по разработке и внедрению программного обеспечения оборудования автоматизированных систем управления технологическим процессом электростанции (А/01.6. Техническое сопровождение оперативной эксплуатации оборудования АСУТП ГЭС/ГАЭС).

Компетенции обучающегося, формируемые в результате освоения дисциплины

В процессе освоения данной дисциплины у студента формируются следующие компетенции: универсальные

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции
УКЦ-2	Способен искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при работе с полученными из различных источников данными с целью эффективного использования полученной информации для решения задач	З-УКЦ-2 Знать: методики сбора и обработки информации с использованием цифровых средств, а также актуальные российские и зарубежные источники информации в сфере профессиональной деятельности, принципы, методы и средства решения стандартных задач профессиональной деятельности с использованием цифровых средств и с учётом основных требований информационной безопасности У-УКЦ-2 Уметь: применять методики поиска, сбора и обработки информации; с использованием цифровых средств, осуществлять критический анализ и синтез информации, полученной из разных источников, и решать стандартные задачи профессиональной деятельности с использованием цифровых средств и с учётом основных требований информационной безопасности В-УКЦ-2 Владеть: методами поиска, сбора и обработки, критического анализа и синтеза информации с использованием цифровых средств для решения поставленных задач, навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с использованием цифровых средств и с учётом требований информационной безопасности

общепрофессиональные

ОПК-6	Способен разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства кон-	З-ОПК-6 Знать: основы алгоритмизации и программирования У-ОПК-6 Уметь: использовать информационно-коммуникационные технологии для практиче-
-------	--	--

	троля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности	ского применения в профессиональной деятельности В-ОПК-6 Владеть: языками программирования как низкого, так и высокого уровня
ОПК-11	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	З-ОПК-11 Знать: принципы работы современных информационных технологий и способы их использования для решения задач профессиональной деятельности У-ОПК-11 Уметь: выбирать современные информационные технологии и использовать их для решения задач профессиональной деятельности В-ОПК-11 Владеть: навыками работы с современными информационными технологиями и способами их использования для решения задач профессиональной деятельности

Профессиональные

Задачи профессиональной деятельности (ЗПД)	Объект или область знания	Код и наименование компетенции	Индикаторы достижения компетенции
Программное обеспечение объектов профессиональной деятельности	Системы и средства автоматизации, управления, контроля, технического диагностирования и информационного обеспечения	ПК-8 Способен проводить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления	З-ПК-8 Знать: основные языки программирования, программные средства автоматизации и систем управления базами данных. У- ПК-8 Уметь: проводить настройку системного, прикладного и инструментального программного обеспечения В- ПК-8 Владеть: методами и алгоритмами инструментального и программного обеспечения систем автоматизации и управления

Задачи воспитания, реализуемые в рамках освоения дисциплины

В процессе освоения данной дисциплины реализуются следующие задачи воспитания:

Направление/цели	Создание условий, обеспечивающих	Использование воспитательного потенциала учебных дисциплин	Вовлечение в разноплановую внеучебную деятельность
Профессиональное и трудовое воспитание	- формирование культуры исследовательской и инженерной деятельности (В16)	Использование воспитательного потенциала дисциплин профессионального модуля для формирования инженерного мышления и инженерной культуры за счёт практических студенческих исследований современных производственных систем; проектной деятельности студентов по разработке и оптимизации технологических систем, связанной с решением реальных производственных задач; прохождения через разнообразные игровые, творческие, фестивальные формы, требующие анализа сложного объекта, постановки относительно него преобразовательных задач для их оптимального решения.	1. Организация научно-практических конференций и встреч с ведущими специалистами предприятий города и ветеранами атомной отрасли. 2. Организация и проведение предметных олимпиад и участие в конкурсах профессионального мастерства. 3. Участие в ежегодных акциях студенческих строительных отрядов

Структура и содержание учебной дисциплины

Дисциплина изучается студентами в 3-4-ом семестрах. Общая трудоемкость дисциплины составляет 8 зачетных единиц, 288 ак. часов.

Календарный план

№ Раздела	№ Темы	Наименование раздела (темы) дисциплины	Виды учебной деятельности (час.)					Аттестация раздела (форма*)	Максимальный балл за раздел**
			Всего	Лекции	Лабораторные	Практические	СРС/КРС		
3 семестр									
1	Раздел 1							Т	30
	1	Понятия технологии и языка программирования. Виды языков программирования, жизненный цикл программы.	17	1			16		
	2	Программирование алгоритмов различной структуры	19	1		2	16		
	3	Работа с массивами и указателями	19/2	1		2/2	16		
	4	Функции	19	1		2	16		
2	Раздел 2							Т	30
	5	Строки	16				16		
	6	Статические структуры и файлы	19	1		2	16		
	7	Директивы препроцессора	16				16		
	8	Линейные динамические структуры данных.	19	1		2	16		
	Итого за 1 семестр		144/2	6		10/2	128		
	Зачет с оценкой							30	40
4 семестр									
3	Раздел 3							Т	30
	1.	Нелинейные динамические структуры. Деревья.	19/2	1		2/2	16		
	2	Графы, способы представления графов	19	1		2	16		
	3	Хеширование и хэш-таблица	19	1		2	16		
	4	Вычислительная сложность алгоритмов. Асимптотический анализ сложности алгоритмов.	17	1			16		
5	Раздел 4							Т	30
	4	Алгоритмы сортировки	19	1		2	16		
	6	Рекурсивные алгоритмы	16				16		
	7	Алгоритмы поиска	19	1		2	16		
	8	Алгоритмы сжатия и кодирования информации	16				16		
	Итого за 2 семестр		144/4	6/2		10/2	128		
	Экзамен							Э	40
Итого			288/6	12/2	-	20/4	256		

Сокращенное наименование форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
Т	Тест
30	Зачет с оценкой
Э	Экзамен

Содержание лекционного курса

Темы лекции. Вопросы, отрабатываемые на лекции	Всего часов	Учебно-методическое обеспечение
1	2	3
3 семестр		
<p>Лекция 1. Понятия технологии и языка программирования. Виды языков программирования, жизненный цикл программы.</p> <p>1.Этапы разработки программного обеспечения, жизненный цикл ПО. Архитектура современных программных приложений.</p> <p>2. Современные технологии разработки ПО. Виды технологий программирования.</p> <p>3.Языки программирования, эволюция языков программирования. Компилируемые и интерпретируемые языки программирования. Стандарты языков программирования.</p> <p>4.Среды разработки программных продуктов. Системы компиляции, этапы сборки исполняемого файла, трансляция программы.</p> <p>5.Алгоритмические языки, характеристики алгоритмического языка. Алфавит языка программирования, ключевые слова и идентификаторы, структура программы, типы данных, выражения, инструкции, разделители, унарные и бинарные операторы. Организация ввода и вывода данных.</p>	1	2 – 5, 14
<p>Лекция 2. Программирование алгоритмов различной структуры</p> <p>1.Программирование алгоритмов линейной структуры на языке высокого уровня, форматный ввод и вывод данных.</p> <p>2.Программирование разветвляющихся алгоритмов, условные операторы, полные и неполные условные операторы, вложенные условные операторы, тернарный оператор, оператор выбора.</p> <p>3. Циклические структуры, программирование циклических алгоритмов, операторы повтора с предусловием, с постусловием и заданным числом итераций, бесконечные циклы, вложенные циклы.</p>	1	1, 3 – 5, 12
<p>Лекция 3. Работа с массивами и указателями</p> <p>1. Массивы, инициализация массивов, способы заполнения массивов, обход массива.</p> <p>2. Массивы переменной длины, размерность массива,</p> <p>3. Обработка массивов: вычисление суммы элементов массива, поиск элемента в массиве, сортировка массивов.</p> <p>4. Массивы строк, ввод и вывод строк, преобразование строк к числовому типу.</p> <p>5. Понятие и назначение указателя, операции получения адреса объекта и разыменования указателя, операции над указателями., адресация и индексация.</p> <p>6. Алгоритмы обработки числовых массивов с использованием указателей, массивы указателей, указатели на массивы, массивы с разной длиной строк, динамическое изменение размера массива</p>	1	1, 3 - 5, 10, 12, 14
<p>Лекция 4. Функции</p> <p>1.Функции, библиотечные функции и функции, созданные пользователем, синтаксическое конструкция пользовательской функции, вызов функции, выход из функции, возврат значения функции.</p> <p>1. Параметры функции, формальные и фактические параметры, функции с неопределенным числом параметров, передача параметров в функцию, указатели на функцию.</p> <p>2. Рекурсия.</p>	1	1, 3 - 5, 10, 12, 14
Раздел 2		
<p>Лекция 5. Статические структуры и файлы</p> <p>1. Структуры, инициализация структур, доступ к членам структуры, массивы структур, вложенные структуры, передача структуры в качестве па-</p>	1	4, 5, 14

раметра. 2. Объединения, перечисления. Примеры использования объединений и перечислений в программах. 3. Файлы в программировании, открытие и закрытие потоков, текстовые и бинарные файлы, режимы открытия и закрытия текстовых и бинарных файлов. 4. Чтение и запись в файл, форматизируемый ввод-вывод, позиционирование в потоке.		
Лекция 11. Линейные динамические структуры данных. 1. Линейные динамические структуры: односвязные и двусвязные списки, стек, очередь. 1. Добавление и удаление элементов в линейную динамическую структуру. 2. Поиск элемента в линейной динамической структуре.	1	1, 4, 5, 8
4 семестр		
3 раздел		
Лекция 1. Нелинейные динамические структуры. Деревья 1. Нелинейные динамические структуры: деревья, способы реализации деревьев. 2. Добавление элементов в дерево, поиск элемента в дереве, удаление элемента из дерева, методы и алгоритмы обхода деревьев. 3. Виды деревьев.	1	1, 4, 5, 7
Лекция 2. Графы 1. Графы, отношения между элементами в графе. 2. Способы представления графов, 3. Алгоритмы поиска в графе, циклы в графе	1	1, 4, 5, 9
Лекция 3 Хеширование и хэш-таблица 1. Хэш-таблица, хэш-функция. 2. Коллизии 3. Поиск с применением хэш-таблицы	1	8, 9, 11
Лекция 4. Вычислительная сложность алгоритмов. Асимптотический анализ сложности алгоритмов. 1. Эффективность и критерии эффективности алгоритмов и программ. 2. Методы оценки вычислительной сложности алгоритмов 3. Асимптотический анализ сложности алгоритмов. Применение асимптотических оценок для оценки сложности.	1	7, 8, 11
4 раздел		
Лекция 5. Алгоритмы сортировки 1. Простые алгоритмы внутренних сортировок 2. Улучшенные алгоритмы сортировки 3. Алгоритмы ускоренных сортировок 4. Критерии сложности алгоритмов сортировки.	1	6, 8, 11
Лекция 7. Алгоритмы поиска 2. Поиск в массивах, критерии оценки алгоритмов поиска, методы поиска. 3. Линейный поиск, алгоритмы поиска посредством сравнения ключей. 4. Алгоритмы поиска образца в тексте, цифровой поиск.	1	6, 8, 11
Итого	12	

Перечень практических занятий

Тема практического занятия. Вопросы, обрабатываемые на практическом занятии	Всего часов	Учебно-методическое обеспечение
1	2	3
1 семестр		
Изучение интегрированной среды разработки ПО 1. Разработка алгоритма вычисления математического выражения. Ис-	2	1, 3 – 5, 12

<p>пользование в программе математических операторов и методов. Организация ввода/вывода в программе</p> <p>2. Разработка алгоритма и программы для решения задачи с использованием условного оператора. Использование оператора выбора в программировании</p> <p>3. Программирование циклических алгоритмов</p>		
<p>Программирование с использованием массивов и указателей</p> <p>1. Объявление и инициализация элементов одномерных и двумерных массивов</p> <p>2. Вычисление суммы и произведения элементов массива</p> <p>3. Поиск элемента в массиве по заданному условию</p> <p>4. Инверсия и сортировка элементов массива</p> <p>5. Создание одномерных массивов переменной длины с помощью указателей.</p> <p>6. Создание двумерных массивов переменной длины с помощью указателей.</p>	2/2	1, 3 - 5, 10, 12, 14
<p>Программирование функций</p> <p>1. Объявление функций в программе, вызов функций.</p> <p>2. Передача параметров в функцию, передача функции в качестве параметра.</p>	2	1, 3 - 5, 10, 12, 14
<p>Программирование с использованием статических структур и бинарных файлов</p> <p>1. Структуры, инициализация структур, доступ к членам структуры, массивы структур.</p> <p>2. Создание программ, использующих бинарные файлы.</p> <p>3. Чтение и запись текстовых файлов в языке программирования Си.</p>	2	4, 5, 14
<p>Однонаправленные динамические списки</p> <p>1. Программирование односвязных списков, добавление элементов в список и удаление элементов из списка. Поиск элемента в списке. Программирование двунаправленных списков, добавление элементов в список и удаление элементов из списка. Поиск узла с заданным значением.</p> <p>2. Программирование стеков, добавление элементов в стек и удаление элементов из стека.</p> <p>3. Программирование очередей, добавление элементов в очередь и удаление элементов из очереди.</p>	2	1, 4, 5, 8
Итого за 1 семестр		
	10	
<p>Бинарное дерево поиска</p> <p>Разработка приложения для создания двоичного дерева поиска.</p>	2	1, 4, 5, 7
<p>Основные алгоритмы работы с графами</p> <p>Разработка программы создания графа и реализация различных процедур для работы с графами</p>	3	1, 4, 5, 9
<p>Хэширование</p> <p>Хэширование как прямой доступ к данным, использование хэш-таблицы для организации прямого доступа к элементам динамического множества данных.</p>	2	8, 9, 11
<p>Эмпирический анализ простых алгоритмов сортировки</p> <p>Оценка эффективности простого алгоритма сортировки</p> <p>Асимптотический анализ сложности алгоритма сортировки.</p> <p>Алгоритмы сортировки прямого и естественного слияния</p> <p>Применение битовых операций, сортировка числового множества битовым массивом, сортировка числового файла битовым массивом</p>	2	7, 8, 11
<p>Алгоритмы поиска в таблице</p> <p>Поиск записей с заданным ключом в двоичном файле с применением различных алгоритмов,</p> <p>Разработка приложений, реализующих алгоритм поиска образца в тексте</p>	2	6, 8, 11
Итого за 2 семестр		
	10	

Задания для самостоятельной работы студентов

Вопросы для самостоятельного изучения (задания)	Всего часов	Учебно-методическое обеспечение
1	2	3
1 семестр		
<p><i>Тема 1. Понятия технологии и языка программирования. Виды языков программирования, жизненный цикл программы.</i></p> <p>1. История развития языков и технологий программирования. 2. Области применения языков программирования. 3. Стандарты жизненного цикла программ 4. Программные платформы</p>	16	2 – 5, 14
<p><i>Тема 2. Программирование алгоритмов различной структуры</i></p> <p>1. Решение задач с использованием сложных условных выражений и оператора выбора 2. Спецификаторы форматного ввода/вывода</p>	16	1, 3 – 5, 12
<p><i>Тема 3. Работа с массивами и указателями</i></p> <p>1. Многомерные массивы и их реализация в программах. 2. Арифметика указателей</p>	16	1, 3 - 5, 10, 12, 14
<p><i>Тема 4. Функции</i></p> <p>1. Функции с переменным количеством параметров.</p>	16	1, 3 - 5, 10, 12, 14
<p><i>Тема 5. Строки</i></p> <p>1.Символьные массивы - строки. Создание строк, строка как набор символов, обращение к символу строки. 2.Массивы строк, ввод и вывод строк, преобразование строк к числовому типу. 3.Работа со строками: копирование строк, конкатенация строк, сравнение строк, разбиение строк на лексемы, поиск подстроки в строке. 4.Функции поиска элементов в строке.</p>	16	1, 3 - 5, 10, 12, 14
<p><i>Тема 6. Статические структуры и файлы</i></p> <p>1. Использование структур для работы с битовыми полями 2. Форматированный ввод и вывод в строки. 3. Позиционирование в потоке, функция управления позицией указателя в потоке, параметры функции позиционирования в потоке.</p>	16	4, 5, 14
<p><i>Тема 10. Директивы препроцессора</i></p> <p>1. Препроцессор языка C, директивы препроцессора: создание макроопределений, директива включения, директивы условной компиляции, директива определяемая реализацией. Препроцессорные операции</p>	16	4, 5
<p><i>Тема 11 Динамические структуры данных</i></p> <p>1. Распределение рабочего пространства оперативной памяти. 2. Реализация алгоритма вычисления значений арифметических выражений, представленных в префиксной форме с помощью очереди. 3. Очереди с приоритетом</p>	16	1, 4, 5, 8
Итого за 1 семестр		128
<p><i>Тема 1. Нелинейные динамические структуры. Деревья.</i></p> <p>Парсер - дерево синтаксического анализатора, парсинг арифметических выражений, построение дерева синтаксического анализа арифметических выражений.</p>	16	1, 4, 5, 7
<p><i>Тема 2. Графы, способы представления графов</i></p> <p>Циклы в графе, гамильтонов и эйлеров циклы. Обход графа для поиска гамильтонова цикла – задача коммивояжера. Реализация алгоритма поиска эйлерова цикла.</p>	16	1, 4, 5, 9
<p><i>Тема 3. Хеширование и хэш-таблица</i></p> <p>Методы разработки хэш-функций: метод деления и метод умножения. Примеры хэш-функций.</p>	16	8, 9, 11

Тема 4. Вычислительная сложность алгоритмов. Асимптотический анализ сложности алгоритмов. Определение эффективности алгоритма с использованием оценки теоретической сложности Определение эффективности алгоритма с использованием практической оценки сложности Правила вычисления асимптотических оценок	16	7, 8, 11
Тема 5. Алгоритмы сортировки Алгоритмы сортировки за линейное время: сортировка методом подсчета, «карманная» сортировка, алгоритм поразрядной сортировки.	16	6, 8, 11
Тема 6 Рекурсивные алгоритмы Теория рекурсивных алгоритмов. Анализ рекурсивных алгоритмов. Разработка и тестирование рекурсивных функций. Рекурсивные алгоритмы обработки данных. Оценка сложности рекурсивного алгоритма: методы решения рекуррентных уравнений: основной метод, метод подстановки, метод деревьев рекурсии.	16	6, 8, 11
Тема 7 Алгоритмы поиска Интерполяционный поиск: метод и алгоритм интерполяционного поиска, анализ интерполяционного поиска	16	6, 8, 11
Тема 8 Алгоритмы сжатия и кодирования информации Классы алгоритмов сжатия. Алгоритмы группового сжатия Исследование алгоритмов сжатия, разработка программ сжатия и восстановления текста. Алгоритмы семейства LZ: суть алгоритма, составляющие алгоритма, принцип скользящего окна, сжатие битовой последовательности с помощью алгоритма Лемпеля-Зива	16	7
Итого за 2 семестр	128	
Итого:	256	

Оценка результатов самостоятельной работы организуется как единство двух форм: самоконтроль и контроль со стороны преподавателя.

С целью самоконтроля каждая тема (раздел) в учебно-методических материалах завершается набором контрольных вопросов и содержит индивидуальные задания.

Курсовая работа

Курсовая работа предусмотрена во 2 семестре. Выполняется по индивидуальному заданию.

Курсовая работа включает в себя задания по темам, изученным в 1 семестре и в 1 разделе 2 семестра. Работа выполняется на языке программирования С.

План выполнения курсовой работы

Срок выполнения по неделям	Этапы выполнения курсовой работы
1-2	Выдача задания, Анализ литературных источников
3	Анализ предметной области
4-5	Подготовка спецификации требований к разрабатываемой программе
6-7	Разработка интерфейса программного решения
8-9	Подготовка источника данных: разработка процедур загрузки данных из файла в динамическую структуру и сохранения данных в файле.
10-11	Разработка и тестирование программного обеспечения: разработка функций добавления, редактирования, удаления данных, поиска, сортировки.
12	Выводы по работе, заключение
13-14	Оформление пояснительной записки
15	Подготовка доклада и презентации к защите курсовой работы
16	Защита курсовой работы

Задание для курсовой работы

1. Изучить предметную область, сформировать и проанализировать требования к разрабатываемому программному решению. Описать назначение программного продукта и выполняемые функции. Определить набор входных данных и требования и ограничения на использование входных данных. Определить формат вывода результатов работы программы в соответствии с потребностями пользователей.

2. Разработать алгоритм работы программного продукта с учетом требований к организации интерфейса взаимодействия пользователя и программного решения.

3. Разработать и описать процедуры программного продукта, реализующие функционал приложения (реализация создания, добавления, удаления, редактирования нового элемента, поиск по параметрам, сортировка по заданному полю). Организовать работу программы, реализовав вызов функций в соответствии с навигационной схемой интерфейса пользователя.

4. Реализовать хранения данных, аккумулируемых в результате работы приложения, в файле.

5. Протестировать работу приложения на всех возможных вариантах входных значений.

6. Оформить результаты разработки в пояснительной записке к курсовой работе.

Примерный перечень тем для курсовой работы

1. Разработка электронного библиотечного каталога.
2. Разработка программного решения регистратора медицинского учреждения.
3. Разработка программного решения для хранения и учета данных о студентах в ВУЗе

Курсовая работа оценивается отдельно, исходя из 100 баллов.

Шкалы оценивания курсового проекта

Баллы (рейтинговой оценки)	Оценка КП (стандарт- ная)	Требования к знаниям
90-100	«отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил основные этапы разработки программ, выполнил презентацию, в которой показал глубокие и прочные знания основных разделов проекта, пояснительную записку курсового проекта выполнил в соответствии с ЕСКД, исчерпывающе, последовательно, четко и логически стройно излагает порядок разработки программного обеспечения, программа разработана без синтаксических и логических ошибок, при проведении тестирования на множестве входных значений не возникает исключительных ситуаций, студент умеет применять теоретических знаний при разработке программных решений, тесно увязывать теорию с практикой, свободно справляется с поставленной задачей, вопросами при защите курсового проекта, причем не затрудняется с ответом при видеоизменении заданий, правильно обосновывает принятое решение
70-89	«хорошо»	Оценка «хорошо» выставляется студенту, если он разработал программное решение в соответствии с требованиями, отсутствуют синтаксические и логические ошибки в программном коде, в результате выполнения программы при различных входных значениях не возникает сбоев, студент выполнил презентацию, в которой показал знания материала, грамотно и по существу излагает его, пояснительную записку курсового проекта и графической части выполнил в соответствии с ЕСКД, не допускал существенных неточностей в ответе на вопросы, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения
60-69	«удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он разработал программный продукт в соответствии с теоретическими знаниями, выполнил презентацию, в которой показал удовлетворительные знания основных этапов; может объяснить языковые конструкции, используемые при разработке программного продукта, получа-

		ются верные результаты при работе программы. При этом не усвоил всех деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении материала.
менее 60	<i>«неудовлетворительно»</i>	Оценка «неудовлетворительно» выставляется студенту, который представил неверные теоретические выкладки, в программном коде допущены синтаксические и логические ошибки, программа не отлажена; студент не может объяснить назначение языковых конструкций, используемых при написании программного решения. Пояснительная записка и графическая часть выполнены с нарушениями требований ЕСКД.

Итоговая оценка за курсовую работу представляет собой сумму баллов, заработанных студентом при выполнении заданий и по итогам защиты курсовой работы, выставляется в соответствии с Положением о кредитно-модульной системе в соответствии со следующей шкалой:

Оценка по 5-балльной шкале	Сумма баллов за курсовую работу	Оценка ECTS
5 – «отлично»	90-100	A
4 – «хорошо»	85-89	B
	75-84	C
	70-74	D
3 – «удовлетворительно»	65-69	E
	60-64	F
2 – «неудовлетворительно»	Менее 60	F

Образовательные технологии

При изучении дисциплины «Языки программирования» используются интерактивные формы проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

При реализации учебного материала курса используются следующие образовательные технологии: использование электронных образовательных ресурсов, обеспечивающих обучение в информационной образовательной среде; лекционные занятия проводятся с использованием мультимедийного лекционного материала, практические занятия проводятся с применением ПК в интегрированной среде разработки программного обеспечения. Самостоятельная работа студентов проводится под руководством преподавателей, с оказанием консультаций и помощи при подготовке к контрольным работам, выполнении домашних заданий.

Фонд оценочных средств

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

№ п/п	Наименование контролируемых разделов (темы)	Код и наименование индикатора достижения компетенций	Наименование оценочного средства
3 семестр			
Входной контроль			
1	Входной контроль		Вопросы и задания входного контроля (письменно)
Аттестация разделов, текущий контроль успеваемости			
2	Раздел 1	3-УКЦ-2, У-УКЦ-2, В-УКЦ-2, 3-ОПК-6, У-УПК-6, В-ОПК-6, 3-ПК-8, В-ПК-8, 3-ОПК-11, У-ОПК-11	Тестирование (письменно) Практические задания (письменно)
3	Раздел 2	3-УКЦ-1, У-УКЦ-2, В-УКЦ-2, 3-ОПК-6, У-УПК-6, В-ОПК-6, 3-ПК-	Тестирование (письменно) Практические задания

		8, У-ОПК-11, В-ОПК-11	(письменно)
Промежуточная аттестация			
4	Зачет с оценкой	3-ОПК-6, В-ОПК-6, 3-ПК-8, У-ОПК-11	Вопросы к зачету (устно)
4 семестр			
5	Раздел 3	У-УКЦ-2, В-УКЦ-2, 3-ОПК-6, У-УПК-6, В-ОПК-6, 3-ПК-8, У-ОПК-11, В-ОПК-11	Тестирование (письменно) Практические задания (письменно) Курсовая работа (письменно)
6	Раздел 4	3-УКЦ-1, 3-ОПК-6, У-УПК-6, В-ОПК-6, 3-ПК-8, В-ПК-8, У-ОПК-11, В-ОПК-11	Тестирование (письменно) Практические задания (письменно)
Промежуточная аттестация			
7	Экзамен	3-ОПК-6, У-ОПК-6, В-ОПК-6, 3-ОПК-11, У-ОПК-11, В-ОПК-11, 3-ПК-8, У-ПК-8, В-ПК-8, 3-УКЦ-2, У-УКЦ-2, В-УКЦ-2	Вопросы к экзамену (устно)

Оценочные средства для входного контроля представляют собой вопросы и практические задания, которые задаются студентам в письменной форме.

В качестве оценочного средства текущего контроля используются выполнение индивидуальных заданий на практических занятиях и устный опрос по результатам их выполнения, рефераты по вопросам, рекомендованным для самостоятельного изучения.

В качестве оценочного средства аттестации раздела используются итоговая контрольная работа или тест.

Для промежуточной аттестации предусмотрены теоретические вопросы.

По итогам обучения в первом семестре выставляется зачёт с оценкой, во втором семестре - экзамен.

Примерные вопросы и задания входного контроля

Входной контроль предназначен для выявления пробелов в знаниях студентов и готовности их к получению новых знаний. Оценочные средства для входного контроля представляют собой вопросы и практические задания, которые задаются студентам в письменной форме.

1. В программе «:=» обозначает оператор присваивания, знаки «+», «-», «*» и «/» — соответственно операции сложения, вычитания, умножения и деления. Правила выполнения операций и порядок действий соответствуют правилам арифметики. Определите значение переменной b после выполнения алгоритма:

$$\begin{aligned} a &:= 2 \\ b &:= 4 \\ a &:= 2*a + 3*b \\ b &:= a/2*b \end{aligned}$$

В ответе укажите одно целое число — значение переменной b .

2. Найдите значение выражения: $((0 \& 0) \cup 0) \& (1 \cup 1)$
3. Зашифрованы русские слова (последовательности букв), вместо каждой буквы записан её код:

А	Д	К	Н	О	С
01	100	101	10	111	000

Расшифруйте закодированное слово 10111000:

4. Для какого из приведённых значений числа X ложно высказывание:
НЕ ($X < 6$) ИЛИ ($X < 5$)?

- a) 7
 - b) 6
 - c) 5
 - d) 4
5. Переведите двоичное число 1101101 в десятичную систему счисления.
 6. Переведите число 135 из десятичной системы счисления в двоичную систему счисления. Сколько единиц содержит полученное число? В ответе укажите одно число — количество единиц.
 7. Доступ к файлу **example.doc**, находящемуся на сервере **task.org**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.
 a) .txt b) :// c) http d) task e) / f) .org g) example
 8. Файл размером 64 Кбайт передаётся через некоторое соединение со скоростью 1024 бит в секунду. Определите размер файла (в Кбайт), который можно передать за то же время через другое соединение со скоростью 256 бит в секунду. В ответе укажите размер файла в Кбайт.
 9. Статья, набранная на компьютере, содержит 48 страниц, на каждой странице 40 строк, в каждой строке 64 символа. Определите размер статьи в кодировке КОИ-8, в которой каждый символ кодируется 8 битами
 10. Для хранения растрового изображения размером 32 x 32 пикселя отвели 512 байт памяти. Каково максимально возможное число цветов в палитре изображения?

Примерный тест (по вариантам) для аттестации раздела 1.

1. Какое из ключевых слов используется для описания вещественного типа?
 - a) double
 - b) int
 - c) char
 - d) bool
2. В записи инициализации двумерного массива: `int x[2][3] = {{0,4,5},{2,2,8}}` число 5 будет находится по адресу:
 - a) [0,0]
 - b) [1,2]
 - c) [0,2]
 - d) [2,3]

3. Чему будет равна переменная `s` в результате выполнения следующего программного кода?

```

7 int _tmain(int argc, _TCHAR* argv[])
8 {
9     int s = 0;
10    for(int i = 1; i <= 3; i++)
11    {
12        s = s + 2 * i;
13    }
14    printf("%d", s);
15    return 0;
16 }

```

- a) 2
 - b) 6
 - c) 12
 - d) 18
4. С помощью какого оператора можно сравнить два числа в программе?
 - a) =
 - b) ==
 - c) :=
 - d) ++
 5. Чему будет равно значение переменной `y` после выполнения последовательности команд:


```
int x=7; int y=x--;
```

- a) 5
- b) 6
- c) 7
- d) 0

6. Какие служебные символы используются для обозначения начала и конца блока кода?

- a) <>
- b) { }
- c) ()
- d) //

7. С помощью следующего программного кода:

```
7 int _tmain(int argc, _TCHAR* argv[])
8 {
9     int mas[3][3] = {{2, 7, 3}, {45, 6, 12}, {43, 23, 5}};
10    for(int i = 0; i < 3; i++)
11    {
12        for(int k = 0; k < 3; k++)
13        {
14            if(i==k)
15            {
16                printf("%d ",mas[i][k]);
17            }
18        }
19    }
20    return 0;
21 }
```

- a) выполняется инверсия одномерного массива
- b) выводятся на экран элементы главной диагонали матрицы
- c) выполняется сортировка каждой строки матрицы
- d) выводится на экран индексы матрицы

8. Какой из идентификаторов записан неверно?

- a) zzz
- b) polnaja_summa
- c) A&X
- d) as55

9. Формальные параметры - это:

- a) собственно параметры функции, объявленные в заголовке функции
- b) значения, которые передаются функции
- c) аргументы функции
- d) переменные, объявленные в функции main

10. В результате выполнения программного кода переменная a будет равна:

```
8 int _tmain(int argc, _TCHAR* argv[])
9 {
10    int a = 5;
11    printf("Начальное значение a=%d\n", a);
12    Inc(&a);
13    printf("После передачи по ссылке a=%d\n", a);
14    return 0;
15 }
16 void Inc(int* x)
17 {
18     *x = *x + 1;
19     printf("Increment: %d\n", *x);
20 }
```

- a) 5
- b) 6
- c) 2
- d) 0

Студент на контрольном тестировании выполняет задания письменно. За каждое правильное решение начисляется баллы. Максимально за контрольный тест - 10 баллов.

Баллы для аттестации раздела включают в себя баллы текущего контроля и контрольного теста.

Шкала оценивания раздела 1

Текущий контроль успеваемости	Аттестация раздела	Максимальный / минимальный балл для аттестации раздела
Задание 1 - 2 балла Задание 2 – 2 балла Задание 3 – 2 балла Задание 4 – 6 баллов Задание 5 – 4 балла Задание 6 – 4 балла	Контрольный тест 1 - 10 б.	30 баллов / 18 баллов

Примерный тест (по вариантам) для аттестации раздела 2.

1. В результате выполнения программы в окно консоли будет выведена строка:

```

8 int _tmain(int argc, _TCHAR* argv[])
9 {
10     char s1[] = "1 2 3 4 5";
11     for(int j = 0; j <= strlen(s1)-1; j++)
12     {
13         if(s1[j] == ' ')
14         {
15             *(s1 + j) = '+';
16         }
17     }
18     printf("%s", s1);
19     return 0;
20 }
```

- a) 12345
 - b) 1+2+3+4+5
 - c) 1+2+3+4+5+
 - d) +1+2+3+4+
2. Динамическую структуру, работающую по алгоритму FIFO (first in, first out - "первый вошел - первый вышел") называют:
- a) двусвязанным списком
 - b) стеком
 - c) хэш-таблицей
 - d) Очередью
3. Сколько указателей используется в односвязных списках?
- a) 1
 - b) 2
 - c) 3
 - d) сколько угодно
4. Линейный список, в котором доступен только последний элемент, называется
- a) стеком
 - b) очередью
 - c) массивом

5. Для работы с файлами и устройствами ввода-вывода необходимо подключить заголовочный файл:
 - a) `stdlib.h`
 - b) `stdio.h`
 - c) `locale.h`
 - d) `math.`

6. Какой режим работы с файлом определяется спецификатором "rb"
 - a) бинарный файл открывается для записи
 - b) бинарный файл открывается для чтения
 - c) текстовый файл открывается для записи. Если файл ранее существовал, то он пересоздается и записывается заново
 - d) текстовый файл создается для записи. Если файл ранее существовал, то при первой записи после открытия он пересоздается и записывается заново. А при последующих записях после открытия данные добавляются в него без перезаписи.

7. Слово `void` при описании функции указывает на то, что:
 - a) функция объявлена как статическая
 - b) функция ничего не возвращает
 - c) функция возвращает числовое значение
 - d) функция обязана вернуть значение типа `void`

8. При работе с переменной типа указатель на структуру используется оператор?
 - a) `.`
 - b) `->`
 - c) `::`
 - d) `!`

9. Для определения объединений применяется ключевое слово:
 - a) `union`
 - b) `Struct`
 - c) `enum`
 - d) `Typedef`

10. Какая директива препроцессора задаёт макроопределение (макрос) или символическую константу:
 - a) `define`
 - b) `undef`
 - c) `include`
 - d) `Ifndef`

Студент на контрольном тестировании выполняет задания письменно. За каждое правильное решение начисляется баллы. Максимально за контрольный тест - 10 баллов.

Баллы для аттестации раздела включают в себя баллы текущего контроля и контрольного теста.

Шкала оценивания раздела 2

Текущий контроль успеваемости	Аттестация раздела	Максимальный / минимальный балл для аттестации раздела
Задание 1 - 6 баллов	Контрольный тест 2 - 10 б.	30 баллов / 18 баллов
Задание 2 – 4 балла		
Задание 3 – 4 балла		
Задание 4 – 6 баллов		

Вопросы для подготовки к зачёту с оценкой

1. Этапы разработки программного обеспечения, жизненный цикл ПО. Архитектура современных программных приложений.
2. Языки программирования, эволюция языков программирования. Компилируемые и ин-

терперируемые языки программирования. Стандарты языков программирования.

3. Среда разработки программных продуктов. Системы компиляции, этапы сборки исполняемого файла, трансляция программы.
4. Основы алгоритмизации, виды и свойства алгоритмов, способы записи алгоритмов.
5. Алгоритмические языки, характеристики алгоритмического языка. Алфавит языка программирования, ключевые слова и идентификаторы, структура программы.
6. Типы данных, выражения, инструкции, разделители, унарные и бинарные операторы.
7. Организация ввода и вывода данных, форматный ввод и вывод данных.
8. Программирование разветвляющихся алгоритмов, условные операторы, полные и неполные условные операторы, вложенные условные операторы, тернарный оператор, оператор выбора.
9. Программирование циклических алгоритмов, операторы повтора с предусловием, с постусловием и заданным числом итераций, бесконечные циклы, вложенные циклы.
10. Массивы, инициализация массивов, способы заполнения массивов, обход массива, массивы переменной длины, размерность массива, обработка массивов: вычисление суммы элементов массива, поиск элемента в массиве, сортировка массивов.
11. Символьные массивы - строки. Создание строк, строка как набор символов, обращение к символу строки. Массивы строк, ввод и вывод строк, преобразование строк к числовому типу. Работа со строками: копирование строк, конкатенация строк, сравнение строк, разбиение строк на лексемы, поиск подстроки в строке.
12. Понятие и назначение указателя, операции получения адреса объекта и разыменования указателя, операции над указателями., адресация и индексация.
13. Алгоритмы обработки числовых массивов с использованием указателей, массивы указателей, указатели на массивы, массивы с разной длиной строк, динамическое изменение размера массива.
14. Функции, библиотечные функции и функции, созданные пользователем, синтаксическое построение пользовательской функции, вызов функции, выход из функции, возврат значения функции.
15. Параметры функции, формальные и фактические параметры, функции с неопределенным числом параметров, передача параметров в функцию, указатели на функцию, рекурсия.
16. Структуры, инициализация структур, доступ к членам структуры, массивы структур, вложенные структуры, передача структуры в качестве параметра.
17. Объединения, перечисления. Примеры использования объединений и перечислений в программах.
18. Файлы в программировании, открытие и закрытие потоков, текстовые и бинарные файлы, режимы открытия и закрытия текстовых и бинарных файлов.
19. Чтение из файла и запись в файл, форматируемый ввод-вывод, позиционирование в потоке.
20. Линейные динамические структуры: односвязные и двусвязные списки, стек, очередь. Добавление и удаление элементов в линейную динамическую структуру. Поиск элемента в линейной динамической структуре.
21. Препроцессор языка C, директивы препроцессора: создание макроопределений, директива включения, директивы условной компиляции, директива определяемая реализацией.

Шкала оценивания на зачете с оценкой

Зачет проводится в устной форме, путем ответа на 2 вопроса из вышеприведенного перечня вопросов. При этом оценивается правильность и полнота ответа. Максимальный балл за зачет – 40 баллов (20 баллов за каждый вопрос).

Оценка знаний на зачете и начисление баллов производится в соответствии со следующей таблицей:

Оценка по 5-балльной шкале	Сумма баллов	Требования к знаниям на экзамене
«отлично»	36 - 40	выставляется студенту, если он полно, грамотно и без ошибок ответил на все вопросы, в том числе и дополнительные.
«хорошо»	30 - 35	выставляется студенту, если он без существенных ошибок ответил на все вопросы, однако допускал отдельные неточности или не демонстрировал достаточно глубокого знания материала

«удовлетворительно»	24 - 29	выставляется студенту, если он в ответах на вопросы продемонстрировал только знание основного материала, допускал существенные неточности в ответах, недостаточно технически грамотно формулировал ответы
«неудовлетворительно»	менее 23	выставляется студенту, если допускал неправильные ответы на поставленные вопросы или не смог ответить на часть вопросов, не смог подтвердить знание значительной части материала.

Итоговая оценка представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля и выставляется в соответствии с Положением о кредитно-модульной системе в соответствии со следующей шкалой:

Оценка по 5-балльной шкале	Сумма баллов за разделы и экзамен	Оценка ECTS
5 – «отлично»	90-100	A
4 – «хорошо»	85-89	B
	75-84	C
	70-74	D
3 – «удовлетворительно»	65-69	D
	60-64	E
2 – «неудовлетворительно»	Менее 60	F

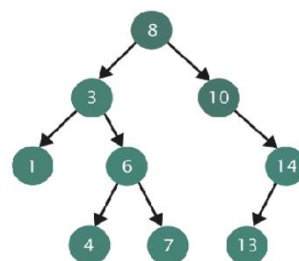
Расшифровка уровня знаний, соответствующего полученным баллам, дается в таблице указанной ниже

Оценка по 5-балльной шкале – оценка по ECTS	Сумма баллов за разделы	Требования к знаниям
«отлично» – A	90 ÷ 100	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.
«хорошо» – C, B	70 ÷ 89	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
«удовлетворительно» – E, D	60 ÷ 69	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.
«неудовлетворительно» – F	менее 60	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Примерные вопросы для подготовки к аттестации 1 раздела (2 семестр):

Студент на коллоквиуме письменно отвечает на 10 вопросов из списка вопросов, приведенных выше. За каждый правильный ответ начисляется баллы. Максимально за коллоквиум - 10 баллов.

1. Элемент дерева, на который нет ссылок, называется:
 - a) корнем
 - b) промежуточным элементом
 - c) листом



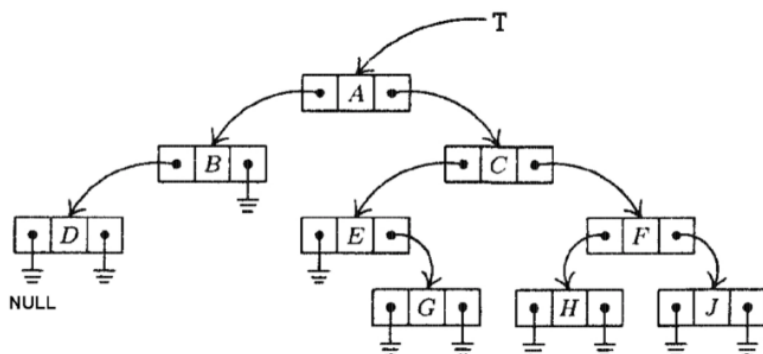
2. На рисунке изображено:

- a) Неупорядоченное неориентированное двоичное дерево
- b) Упорядоченное неориентированное бинарное дерево
- c) Неупорядоченное ориентированное бинарное дерево
- d) Упорядоченное ориентированное двоичное дерево

3. Элемент дерева, который не ссылается на другие, называется:

- a) корнем
- b) листом
- c) узлом
- d) промежуточным

4. При обходе дерева в прямом порядке, будет напечатано:



- a) A B D C E G F H J
- b) A B C D E F G H J
- c) D B A E G C H F J
- d) D B G E H J F C A

5. Элемент дерева, который имеет предка и потомков, называется

- a) корнем
- b) листом
- c) вершиной
- d) промежуточным узлом

6. Дерево является структурой, называемой ...

- a) сетевой
- b) циклической
- c) иерархической
- d) замкнутой

7. Графом G называется пара множеств (V, E) , где V – конечное множество элементов, называемых вершинами графа, а E – конечное множество упорядоченных пар $e=(u, v)$, называемых ...

- a) дугами
- b) отрезками
- c) линиями
- d) узлами

8. Граф составляет пара множеств, элементы которых называются ...

- a) Дугами и вершинами
- b) Ребрами и точками
- c) Гранями и вершинами

9. Обращение в массиве к элементу по индексу можно оценить как:

- a) Алгоритм, в котором порядок не зависит от n , и который выполняется за конечное время
- b) Алгоритм с кубическим временем выполнения высокой сложности
- c) Алгоритм квадратичной сложности, используемый для небольших наборов данных
- d) Алгоритм логарифмической сложности

10. Количество выполняемых операций алгоритма при возрастании объема входных данных называется:

- a) Скоростью роста
- b) Интерполяцией
- c) Пунктом роста
- d) Классом сложности

Шкала оценивания раздела 1

Текущий контроль успеваемости	Аттестация раздела	Максимальный / минимальный балл для аттестации раздела
Задание 1 – 5 баллов Задание 2 – 6 баллов Задание 3 – 3 балла Задание 4 – 6 баллов	Контрольный тест 1 - 10 б.	30 баллов / 18 баллов

Примерный тест для аттестации раздела 2 (2 семестр).

1. Оптимизация последующего поиска элементов в упорядоченном множестве – цель ...
 - a) Сортировки
 - b) Инициализации
 - c) Визуализации
2. Внутренняя сортировка обычно обрабатывает данные, хранящиеся ...
 - a) На дисках
 - b) В оперативной памяти
 - c) На бумажных носителях
3. Трудоемкость сортировки составляет $n \cdot n$ операций сравнений и обменов в случае ...
 - a) Пирамидальной сортировки
 - b) Сортировки простыми включениями
 - c) Сортировки естественными слияниями
4. «Начиная с конца массива, сравниваем значения соседних элементов. Если номер позиции меньшего из этих элементов больше номера позиции большего элемента, то меняем их местами». Приведенный фрагмент описывает основную идею
 - a) Пузырьковой
 - b) Простыми включениями
 - c) Простым выбором
5. В случае внешней сортировки серией называется последовательность элементов, для которых выполняется условие ...
 - a) Упорядоченности
 - b) Взвешенности
 - c) определенности
6. Методы хэширования основаны на ...
 - a) Ассоциативной адресации
 - b) Абсолютной адресации

- c) Относительной адресации
7. Рекурсия с динамической базой предполагает ...
- a) Улучшение прозрачности алгоритма при его записи
- b) Увеличение скорости работы алгоритма
- c) Расширение диапазона значений используемых величин
8. Поиск на графе может быть ...
- a) Глубоким и широким
- b) Информативным и неинформативным
- c) Минимальным и максимальным
9. Критерий сжатия информации, который определяется как отношение длины исходного представления к длине сжатого представления данных, называется
- a) Качество сжатия
- b) Скорость кодирования
- c) Объем сжатия памяти
10. В случае поиска некоторой числовой идентификатор, однозначно определяющий запись в определенном множестве, называется ...
- e) Именем
- f) Признаком
- g) Ключом

Шкала оценивания раздела 2

Текущий контроль успеваемости	Аттестация раздела	Максимальный / минимальный балл для аттестации раздела
Задание 1 - 6 баллов Задание 2 – 6 баллов Задание 3 – 6 баллов Задание 4 - 2 балла	Контрольный тест - 10 б.	30 баллов / 18 баллов

Примерные вопросы для подготовки к экзамену

1. Нелинейные динамические структуры: деревья, способы реализации деревьев.
2. Добавление элементов в дерево, поиск элемента в дереве, удаление элемента из дерева, методы и алгоритмы обхода деревьев.
3. Виды деревьев.
4. Графы, отношения между элементами в графе.
5. Способы представления графов.
6. Алгоритмы поиска в графе.
7. Циклы в графе.
8. Хеширование и хэш-таблица
9. Хэш-таблица, хэш-функция.
10. Коллизии
11. Поиск с применением хэш-таблицы
12. Вычислительная сложность алгоритмов. Эффективность и критерии эффективности алгоритмов и программ.
13. Асимптотический анализ сложности алгоритмов.
14. Методы оценки вычислительной сложности алгоритмов
15. Асимптотический анализ сложности алгоритмов. Применение асимптотических оценок для оценки сложности.
16. Простые алгоритмы внутренних сортировок
17. Улучшенные алгоритмы сортировки
18. Алгоритмы ускоренных сортировок
19. Критерии сложности алгоритмов сортировки.

20. Теория рекурсивных алгоритмов
21. Анализ рекурсивных алгоритмов
22. Алгоритмы поиска. Поиск в массивах, критерии оценки алгоритмов поиска, методы поиска.
23. Линейный поиск, алгоритмы поиска посредством сравнения ключей.
24. Алгоритмы поиска образца в тексте, цифровой поиск
25. Алгоритмы сжатия и кодирования информации. Классы алгоритмов сжатия
26. Алгоритмы группового сжатия.

Шкала оценивания на экзамене

Экзамен проводится в устной форме, путем ответа на 2 вопроса из вышеприведенного перечня вопросов. При этом оценивается правильность и полнота ответа. Максимальный балл за зачет – 40 баллов (20 баллов за каждый вопрос).

Оценка знаний на зачете и начисление баллов производится в соответствии со следующей таблицей:

Оценка по 5-балльной шкале	Сумма баллов	Требования к знаниям на экзамене
«отлично»	36 - 40	выставляется студенту, если он полно, грамотно и без ошибок ответил на все вопросы, в том числе и дополнительные.
«хорошо»	30 - 35	выставляется студенту, если он без существенных ошибок ответил на все вопросы, однако допускал отдельные неточности или не демонстрировал достаточно глубокого знания материала
«удовлетворительно»	24 - 29	выставляется студенту, если он в ответах на вопросы продемонстрировал только знание основного материала, допускал существенные неточности в ответах, недостаточно технически грамотно формулировал ответы
«неудовлетворительно»	менее 23	выставляется студенту, если допускал неправильные ответы на поставленные вопросы или не смог ответить на часть вопросов, не смог подтвердить знание значительной части материала.

Итоговая оценка представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля и выставляется в соответствии с Положением о кредитно-модульной системе в соответствии со следующей шкалой:

Оценка по 5-балльной шкале	Сумма баллов за разделы и экзамен	Оценка ECTS
5 – «отлично»	90-100	A
4 – «хорошо»	85-89	B
	75-84	C
	70-74	D
3 – «удовлетворительно»	65-69	E
	60-64	
2 – «неудовлетворительно»	Менее 60	F

Расшифровка уровня знаний, соответствующего полученным баллам, дается в таблице указанной ниже

Оценка по 5-балльной шкале – оценка по ECTS	Сумма баллов за разделы	Требования к знаниям
«отлично» – A	90 ÷ 100	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.

«хорошо» – С, В	70 ÷ 89	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
«удовлетворительно» – Е, D	60 ÷ 69	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.
«неудовлетворительно» – F	менее 60	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Учебно-методическое и информационное обеспечение учебной дисциплины

Основная литература

1. Андрианова, А. А. Алгоритмизация и программирование. Практикум : учебное пособие / А. А. Андрианова, Л. Н. Исмагилов, Т. М. Мухтарова. — Санкт-Петербург : Лань, 2019. — 240 с. — ISBN 978-5-8114-3336-0. — Текст: электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/113933>
2. Калентьев, А. А. Новые технологии в программировании : учебное пособие / А. А. Калентьев. — Москва : ТУСУР, 2014. — 176 с. <https://e.lanbook.com/reader/book/110361/#1>
3. Орлов С. Теория и практика языков программирования. Учебник для вузов. Стандарт 3-го поколения / С. Орлов. - Санкт-Петербург : Питер, 2014. - 688 с. - URL: <https://ibooks.ru/bookshelf/26402/reading>
4. Подбельский, В. В. Курс программирования на языке Си : учебник / В. В. Подбельский, С. С. Фомин. — Москва : ДМК Пресс, 2012. — 384 с. — URL: <https://e.lanbook.com/book/4148>
5. Рацев, С. М. Программирование на языке Си : учебное пособие для вузов / С. М. Рацев. — Санкт-Петербург : Лань, 2022. — 332 с.— URL: <https://e.lanbook.com/book/193320>
6. Рысин, М. Л. Введение в структуры и алгоритмы обработки данных : учебное пособие / М. Л. Рысин, М. В. Сартаков, М. Б. Туманова. — Москва : РТУ МИРЭА, 2022 — Часть 1 : Сложность алгоритмов. Сортировки. Линейные структуры данных. Поиск в таблице — 2022. — 110 с. — ISBN 978-5-7339-1612-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/256592>
7. Рысин, М. Л. Введение в структуры и алгоритмы обработки данных : учебное пособие / М. Л. Рысин, М. В. Сартаков, М. Б. Туманова. — Москва : РТУ МИРЭА, 2022 — Часть 2 : Поиск в тексте. Нелинейные структуры данных. Кодирование информации. Алгоритмические стратегии — 2022. — 111 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/310826>
8. Скворцова, Л. А. Структуры и алгоритмы обработки данных : учебно-методическое пособие / Л. А. Скворцова, К. В. Гусев, С. М. Трушин. — Москва : РТУ МИРЭА, 2021. — 235 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/218699>
9. Структуры и алгоритмы обработки данных : учебно-методическое пособие / Л. А. Скворцова, К. В. Гусев, А. С. Филатов, С. Р. Ермаков. — Москва : РТУ МИРЭА, 2022 — Часть 2 : Неэлементарные структуры данных — 2022. — 360 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/311015>

Дополнительная литература:

10. Гуркова, М. А. Программирование на языке Си: Практикум : учебное пособие / М. А. Гуркова, Э. Р. Резникова. — Москва : РУТ (МИИТ), 2020. — 70 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/>
11. Кораблин, Ю. П. Структуры и алгоритмы обработки данных : учебно-методическое пособие / Ю. П. Кораблин, В. П. Сыромятников, Л. А. Скворцова. — Москва : РТУ МИРЭА, 2020. — 219 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/163860>

12. Программирование. Сборник задач: учебное пособие / О. Г. Архипов, В. С. Батасова, П. В. Гречкина [и др.] ; под редакцией М. М. Марана. — Санкт-Петербург : Лань, 2019. — 140 с. — ISBN 978-5-8114-3857-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/121485>
13. Структуры и алгоритмы компьютерной обработки данных : учебно-методическое пособие / Ю. М. Мартынюк, В. С. Ванькова, С. В. Даниленко, Б. П. Ваньков. — Тула : ТГПУ, 2021. — 71 с. — ISBN 978-5-6047370-4-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/253676>
14. Тракимус, Ю. В. Основы программирования : учебное пособие / Ю. В. Тракимус, В. П. Хиценко. — Новосибирск : НГТУ, 2020. — 66 с. — ISBN 978-5-7782-4089-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/152224>

Программное обеспечение и Интернет-ресурсы

Процесс реализации образовательной программы обеспечен необходимым комплектом лицензионного программного обеспечения.

Ресурсы информационно-телекоммуникационной сети «Интернет»:

1. Электронно-библиотечная система «Лань» - <http://e.lanbook.com>.

Материально-техническое обеспечение учебной дисциплины

Лекционные занятия по дисциплине проводятся в аудитории с мультимедийным оборудованием. Для проведения практических занятий и выполнения самостоятельной работы используются учебные компьютерные классы с выходом в Интернет и лицензионным программным обеспечением.

Учебно-методические рекомендации для студентов

1. Указания для прослушивания лекций

Перед началом занятий внимательно ознакомиться с учебным планом проведения лекций и списком рекомендованной литературы.

Перед посещением очередной лекции освежить в памяти основные концепции пройденного ранее материала. Подготовить при необходимости вопросы преподавателю.

На лекции основное внимание следует уделять не справочной информации по языку программирования, а примерам использования инструментария языка.

В процессе изучения лекционного курса необходимо по возможности часто возвращаться к основным понятиям и методам решения задач (здесь возможен выборочный контроль знаний студентов).

Желательно использовать конспекты лекций, в которых используется принятая преподавателем система обозначений.

Для более подробного изучения курса следует работать с рекомендованными литературными источниками и вновь появляющимися источниками.

2. Указания для участия в практических занятиях

Перед посещением уяснить тему практического занятия и самостоятельно изучить теоретические вопросы, на основании лекционных материалов предшествующих дисциплин, рекомендованного списка литературы, а также справочной документации по синтаксису языка программирования.

При работе с персональным компьютером соблюдать правила техники безопасности, выполнять работу в соответствии с заданием.

В конце занятия при необходимости выяснить у преподавателя неясные вопросы.

3. Самостоятельная работа студентов обычно складывается из нескольких составляющих:

- работа с текстами: учебниками, справочниками, дополнительной литературой, в том числе материалами интернета, а также проработка вопросов для самостоятельного изучения;
- подготовка к практическим занятиям;
- написание рефератов;
- выполнение курсовой работы;
- подготовка к зачету и экзамену непосредственно перед ним.

Таким образом, самостоятельная работа студентов является необходимым компонентом получения полноценного высшего образования.

Методические рекомендации для преподавателей

1. Указания для проведения лекций

На первой вводной лекции сделать общий обзор содержания курса и отметить новые методы и подходы к решению задач, рассматриваемых в курсе, довести до студентов требования кафедры, ответить на вопросы.

При подготовке к лекционным занятиям необходимо продумать план его проведения, содержание вступительной, основной и заключительной части лекции, ознакомиться с новинками учебной и методической литературы по теме лекционного занятия. Уточнить план проведения практического занятия по теме лекции. Перед изложением текущего лекционного материала напомнить об основных итогах, достигнутых на предыдущих лекциях. С этой целью задать несколько вопросов аудитории и осуществить выборочный контроль знания студентов.

В ходе лекционного занятия преподаватель должен назвать тему, учебные вопросы, ознакомить студентов с перечнем основной и дополнительной литературы по теме занятия.

Следует теоретические сведения иллюстрировать примерами. Задавать по ходу изложения лекционного материала риторические вопросы и самому давать на них ответ. Это способствует активизации мыслительной деятельности студентов, повышению их внимания и интереса к материалу лекции, ее содержанию. Преподаватель должен руководить работой студентов по конспектированию лекционного материала, подчеркивать необходимость отражения в конспектах основных положений изучаемой темы, особо выделяя, категориальный аппарат. В заключительной части лекции необходимо сформулировать общие выводы по теме, раскрывающие содержание всех вопросов, поставленных в лекции. Объявить план очередного практического занятия, дать краткие рекомендации по подготовке студентов к практике. Определить место и время консультации студентам, пожелавшим выступить на практическом занятии с докладами и рефератами.

На последней лекции уделить время для обзора наиболее важных положений, рассмотренных в курсе.

2. Указания для проведения практических занятий

Довести до студентов перечень практических работ, объем и график выполнения курсовой работы, ответить на вопросы.

В ходе практического занятия преподаватель должен назвать тему, учебные вопросы, ознакомить студентов с перечнем основной и дополнительной литературы по теме занятия, алгоритмом его выполнения.

Преподаватель должен руководить работой студентов по выполнению практической работы. В процессе составления программ вести дискуссию со студентами о правильности применения теоретических знаний.

Отмечать студентов, наиболее активно участвующих в решении задач на ПК. При отчете по результатам выполнения практической работы провести устный опрос, позволяющий оценить уровень выполнения работы.

2. Указания по контролю самостоятельной работы студентов

Одним из видов самостоятельной работы студентов по данной дисциплине является выполнение курсовой работы. Для контроля успеваемости отслеживать график выполнения КР, представленный выше, проводить процентовки.

При подготовке к промежуточной аттестации по дисциплине контролировать данный процесс путем устного опроса.

Программа составлена в соответствии с требованиями ОС НИЯУ МИФИ и учебным планом основной образовательной программы по направлению подготовки 27.03.04 Управление в технических системах.

Рабочую программу составил : преподаватель Несытых И.В.

Рецензент: доцент Мефедова Ю.А.

Программа одобрена на заседании УМКН 27.03.04 Управление в технических системах.

Председатель учебно-методической комиссии Мефедова Ю.А.