

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ
Декан ФИиВТ

УТВЕРЖДАЮ /А.А. Кречетов/
(Ф.И.О. декана (директора института))

25.01.2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б.1.1.14 Алгоритмы и структуры данных

(код и наименование дисциплины по учебному плану)

Направление подготовки
(специальность)

09.03.04 Программная инженерия

Квалификация выпускника

Бакалавр

(бакалавр/магистр/специалист)

Направленность

Разработка программных систем

Курс 2
Семестр 3

Распределение учебного времени

Трудоемкость по учебному плану	180 / 5	часов/зачетных единиц
Лекции	36	часов
Лабораторные работы	36	часов
Практические занятия	-	часов
Иная контактная работа	-	часов
Всего контактной работы (без учета экз.)	72	часов
Контактная работа по экзамену	6	часов
Курсовой проект (работа)	-	семестр
Самостоятельная работа обучающихся (без учета экз.)	72	часов
Самостоятельная работа по подготовке к экзамену	30	часов
Экзамен	3	семестр
Зачет	-	семестр
БРК, ДЗ	-	семестр

(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 09.03.04 Программная инженерия

Программу составили:

доцент с ученой степенью кандидата наук	ИиСП	СОГЛАСОВАНО	В.И. Галочкин
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина
Кафедра информатики и системного программирования

		(наименование кафедры)	
25.01.2023	протокол №	1	
(дата)			
Заведующий кафедрой	СОГЛАСОВАНО	А.В. Бородин	
		(И.О. Фамилия)	

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими)
кафедрой(ами).
СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	А.В. Бородин
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит
выпускающая кафедра

СОГЛАСОВАНО	А.А. Кречетов
	(И.О. Фамилия)

Эксперт(ы): Егошин А.Б., Ген. директор ООО "Цитрус"

Рабочая программа проверена и зарегистрирована в УМЦ 14.02.2023 г.
Специалист учебно-методического центра СОГЛАСОВАНО /И.Р. Валиева/

Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1 Знает основы математики, физики, вычислительной техники и программирования.	знания: Знает алгоритмы и структуры данных для обработки списков, деревьев, графов; методы поиска и сортировки данных умения: навыки:
	ОПК-1.2 Умеет решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования	знания: умения: Умеет решать профессиональные задачи путем разработки эффективных алгоритмов, структур данных и программ на основе общеинженерных знаний навыки:
	ОПК-1.3 Имеет навыки теоретического и экспериментального исследования объектов профессиональной деятельности.	знания: умения: навыки: Имеет навыки по тестированию и отладке программных приложений
2. ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-	ОПК-3.1 Знает принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	знания: Знает средства решения задач профессиональной деятельности на основе поиска информации в интернете и электронных библиотеках умения: навыки:

коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.2 Умеет решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	знания: умения: Умеет решать стандартные профессиональные задачи на основе критического анализа информации в интернете и электронных библиотеках навыки:
	ОПК-3.3 Имеет навыки подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности	знания: умения: навыки: Имеет навыки составления программной документации для докладов и публикаций

Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к обязательной части ОПОП.

Дисциплина является обязательной

Для продолжения формирования заявленных компетенций необходимы знания предшествующих дисциплин: Математика (ОПК-1), Дискретная математика (ОПК-1), Основы программирования (ОПК-3)

Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих дисциплинах: Математическая логика и теория алгоритмов (ОПК-1), Методы и средства защиты компьютерной информации (ОПК-3); практиках: Учебная практика. Научно-исследовательская работа (ОПК-1), Преддипломная практика (ОПК-1); государственной итоговой аттестации в форме: Подготовка к процедуре защиты и защита выпускной квалификационной работы (ОПК-1), Подготовка к процедуре защиты и защита выпускной квалификационной работы (ОПК-3)

Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: дискуссионные, исследовательские, лекционные занятия, практические и лабораторные занятия, процедуры самообучения

На достижение конкретных целей обучения направлены применяемые тактические технологии: классическая лекция

Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
Фундаментальные структуры данных	16	ОПК-1
Лекция. Эволюция представлений о взаимосвязи программы и данных в процедурных и объектно-ориентированных языках программирования.	2	
Лекция. Статические и динамические структуры данных	2	
Лабораторная работа. Фундаментальные структуры данных	4	
Задания для самостоятельной работы, в том числе выполнение 1. Проработка лекций. 2. Подготовка к лабораторным работам. 3. Изучение способов организации статических и динамических структур данных. 4. Освоение работы с файлами последовательного и прямого доступа. 5. Особенности обработки текстовых файлов.	8	
Линейные списки	24	ОПК-1
Лекция. Стеки: организация и применение	2	
Лекция. Очереди и деки: организация и применение	2	
Лекция. Алгоритм Дейкстры преобразования алгебраических выражений в постфиксную форму	2	
Лабораторная работа. Линейные списки	6	
Задания для самостоятельной работы, в том числе выполнение 1. Проработка лекций 2. Подготовка к лабораторным работам 3. Освоение способов организации списков на основе статических и динамических структур данных 4. Изучение контейнерных классов языка C++ для организации и обработки списков. 5. Преобразование в постфиксную форму простых выражений	12	
Деревья	24	ОПК-1, ОПК-3
Лекция. Деревья и способы их организации в памяти	2	
Лекция. Рекурсивные алгоритмы обхода деревьев	2	
Лекция. Обходы деревьев на основе явного стека	2	
Лабораторная работа. Деревья	6	
Задания для самостоятельной работы, в том числе выполнение 1. Проработка лекций 2. Подготовка к лабораторным работам 3. Освоение схем рекурсивного обхода деревьев. 4. Определение веса дерева путем использования рекурсии и стека.	12	
Графы	36	ОПК-1, ОПК-3
Лекция. Представление графов с помощью матрицы смежности, списков смежности и динамических структур	2	

Лекция. Обходы графов в глубину и ширину	2	ОПК-1, ОПК-3
Лекция. Алгоритмы Дейкстры и Флойда поиска кратчайших путей на графе	2	
Лекция. Алгоритмы поиска остовных деревьев Прима и Краскала	4	
Лабораторная работа. Графы	10	
Задания для самостоятельной работы, в том числе выполнение 1. Проработка лекций 2. Подготовка к лабораторным работам 3. Изучить достоинства и недостатки различных представлений графа. 4. Освоение алгоритмов DFS и BFS – обхода графа в глубину и ширину. 5. Изучение алгоритмов Дейкстры и Флойда на примерах тестовых графов. 6. Проведение сравнение вычислительной сложности алгоритмов Прима и Краскала для полных и разреженных графов	16	
Поиск и сортировка данных	44	
Лекция. Последовательный, бинарный и тернарный поиск	2	
Лекция. Бинарные деревья поиска и их балансировка	2	
Лекция. Б-деревья и хеширование	2	
Лекция. Методы внутренней сортировки	2	
Лекция. Методы внешней сортировки	2	
Лабораторная работа. Поиск и сортировка данных	10	
Задания для самостоятельной работы, в том числе выполнение 1. Проработка лекций 2. Подготовка к лабораторным работам 3. Сравнение характеристик методов поиска данных. 4. Освоение методов поиска данных во внешней памяти. 5. Освоение быстрой сортировки из библиотеки STL в языке C++. 6. Оценка целесообразность применения усовершенствованных методов внешней сортировки в зависимости от объема данных 7. Выполнение тестов для самоконтроля	24	
Иная контактная работа:	0	
Подготовка к экзамену	30	
Проведение экзамена	6	

Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

1. Советы по планированию и организации времени, необходимого на изучение дисциплины

Не допускать значительных перерывов в занятиях. Как и при изучении иностранных языков, регулярность занятий способствует более быстрому и глубокому изучению материала. Наиболее опасно запускать предмет, а затем штурмовать его перед очередным контрольным мероприятием.

Готовиться к каждому аудиторному занятию, чтобы не вспоминать (с потерями) пройденный материал.

Весь материал, приведенный на аудиторных занятиях, должен быть детально проработан.

Фиксировать все моменты, по которым не достигнуто полное понимание, чтобы задать необходимые вопросы товарищам и преподавателю.

В каждом алгоритме полезно осознать необходимость всех условий и подбирать примеры, когда при нарушении условий алгоритм неработоспособен.

Прежде чем разрабатывать собственную программу полезно подробно рассмотреть примеры, приводимые на занятиях, а также имеющиеся в библиотеке учебных программ. Имеет смысл изменять варианты представления данных и анализировать, что при этом изменится в коде программы.

Невозможно научиться программировать, списывая у товарищей и даже разбирая приведенные примеры. Гораздо полезнее самому решать задачу, не добившись успеха, чем использовать что-то готовое. Только решение достаточно большого объема задач способно выработать необходимые практические навыки.

Проявлять здоровую бдительность при изучении конспектов лекций, книг, методических материалов. Конспекты могут содержать ошибки, преподаватель также может ошибаться, в книгах нередко встречаются неточности и опечатки.

Безусловно полезны коллективные занятия, но они не должны носить характер перелицовки программ более сильного или трудолюбивого товарища. Известно, что объясняющий и сам, как правило, добивается более глубокого понимания темы.

Просматривать материал вперед, что значительно повышает уровень восприятия.

Изучать материал не только по конспектам лекций, но и по приведенным источникам, анализируя, чем отличаются разные подходы, термины и понятия.

Рекомендуемое время на подготовку к лекционным занятиям – 30 мин., лабораторным работам – 1 час.

2. Описание последовательности действий студента или "сценарий изучения дисциплины"

Организация процесса обучения должна подчиняться требованиям рабочей программы и технологической карты в системе РИТМ. Дисциплина состоит из пяти модулей-разделов. Дадим общие рекомендации по их изучению.

Фундаментальные структуры данных. Прежде всего следует повторить приемы работы с текстовыми файлами. Далее нужно обратить внимание на файлы прямого доступа. Полезно рассмотреть тип данных "множество" и варианты представления множеств в памяти. Динамические структуры данных строятся на указателях. Целесообразно описать особенности использования указателей в таких языках программирования, как Паскаль,

C++, Java. Использование указателей дает многие выгоды, но приводит к дополнительным трудностям для программиста. Это могут пояснить примеры ошибок, связанные с потерянными указателями. Для повторения и понимания организации строчных типов данных имеет смысл запретить использование стандартных функций обработки строк.

Линейные списки. Наиболее употребительны такие списки, как стеки и очереди. Нужно подробно описать типовые процедуры работы со стеками и очередями, как в статическом, так и динамическом представлении, привести ряд примеров их использования в системном программировании и прикладных задачах. В лабораторных работах некоторые студенты используют стандартные классы C++ для работы со стеками и очередями. С одной стороны, это показывает эрудицию студентов, однако не дает полного представления об этих структурах данных. Также необходимо предостеречь от чрезмерного увлечения рекурсией, которая приводит к проблемам при тестировании. Использование явных стеков часто оказывается более простым решением.

Деревья. Необходимо понять различные варианты представления деревьев на основе статических и динамических структур данных. В первую очередь рассматриваются рекурсивные алгоритмы обхода деревьев. Для наглядности стоит подробно рассмотреть схему рекурсивных вызовов при каком-либо варианте обхода. В дальнейшем использование рекурсии существенно облегчается, полным пониманием. Алгоритмы обхода деревьев с помощью явного стека помимо ряда собственных достоинств позволяют повторить материалы предыдущего раздела.

Графы. Сначала рассматривается представление графов на основе матрицы смежности. Ряд графовских алгоритмов описывается с помощью матричных операций. Далее нужно разобраться в примерах задач, когда более рационально использовать динамическое представление графов. Недостаток лекционного времени не позволяет охватить все многообразие алгоритмов на графах. Действительно, в ряде университетов для математиков читается полугодовой курс по теории графов, даже не связанный с программированием прикладных задач. Наиболее важно подробно рассмотреть алгоритмы обхода графа в глубину и ширину, а также алгоритмы поиска кратчайших путей Дейкстры и Флойда. Во всех алгоритмах необходимо приводить оценку трудоемкости и понять важность этой оценки на конкретных примерах. Лабораторные работы содержат большое количество прикладных задач. В некоторых из них (обмены квартир, взаимное списание долгов) используются неявные графы.

Методы поиска и сортировки. Рассмотренные ранее структуры и алгоритмы обработки данных активно используются в задачах поиска. Особенно наглядно это заметно на примерах бинарных деревьев поиска и Б-деревьев. В современных СУБД быстрый поиск данных основан на применении Б-деревьев и хешировании. Нужно классифицировать алгоритмы и структуры данных для поиска в зависимости от их сложности, физического размещения данных, возможности изменения данных в процессе работы программы, трудоемкости алгоритмов. Задачи поиска данных напрямую связаны с задачами сортировки. Необходимо четко разграничить особенности внутренней и внешней сортировки. Студенты неоправданно часто используют пузырьковую сортировку, имеющую репутацию самой медленной. Нужно привести аналитические и экспериментальные сравнительные оценки трудоемкости различных методов сортировки. Особо подробно стоит остановиться на быстрой сортировке, вошедшей в библиотеку шаблонов STL языка C++.

3. Рекомендации по использованию материалов рабочей программы

Для более быстрого и методически правильного освоения дисциплины необходимо начать ее изучение с внимательного рассмотрения рабочей программы. Рабочая программа позволит оценить трудоемкость освоения дисциплины, укажет на контрольные точки, на длительность изучения дисциплины, наличие контрольных мероприятий.

Следует посмотреть рекомендуемую литературу и взять ее в библиотеке, причем потребуется литература как по освоению теоретического материала, так и по выполнению лабораторных работ.

Следует периодически обращаться к контрольным материалам, размещенным в учебно-методическом комплексе дисциплины.

При подготовке к экзамену посмотреть вариант билета, проработать экзаменационные вопросы и просмотреть рекомендуемую литературу.

4. Рекомендации по работе с литературой

В библиотеке имеется достаточное количество экземпляров учебно-методической литературы. Помимо изданных пособий много материалов представлено в электронном виде и имеется на кафедре и у преподавателя.

При работе с литературой рекомендуется пытаться в первую очередь понять смысл вводимых терминов, их связь, практическую значимость, а не заучивать определения, формулы, элементы структур. Последовательность подачи материала соответствует рекомендациям учебной программы;

Для успешного выполнения лабораторных работ имеет смысл в качестве тренировки выполнять на компьютере базовые программы из библиотеки учебных программ, предоставляемой преподавателем, осмысливая получаемые результаты. Указанные программы допускается использовать в качестве фрагментов выполняемых лабораторных работ.

5. Советы по подготовке к экзамену, критерии экзаменационных (зачетных) оценок

Навыки в разработке алгоритмов и программ вырабатываются и оцениваются на лабораторных занятиях, а также при выполнении и разборе контрольных работ. Достиженные результаты непосредственно влияют на рейтинг, с которым студент выходит на экзамен.

Экзаменационный билет состоит из двух теоретических вопросов. По каждому из них задаются дополнительные задания качественного характера. Студент должен показать понимание вопросов и применение теоретических положений на практике. К сожалению, это бывает не всегда. Некоторые студенты, даже обладая хорошей техникой программирования, с трудом усваивают теоретический материал. Это приводит к тому, что

разработанные программы правильно работают на небольших по объему тестах, но "виснут" на реальных данных.

Совершенно недопустимо заучивать термины и определения без уяснения их смысла. Особое внимание стоит обратить на те вопросы, по которым в семестре выявлялись определенные проблемы.

Критерии оценки за экзаменационный ответ:

На тройку необходимо знать основные теоретические положения по каждому вопросу билета.

На четверку необходимо показать уровень тройки и дать полную и правильную интерпретацию применения теории по заданному преподавателем примеру (привести подробные шаги алгоритма, представить программные фрагменты, обосновать получаемые результаты, описать необходимые структуры данных).

На пятерку необходимо показать уровень четверки и выполнить дополнительное задание по связанным с вопросом темам (привести другие подходы, описать применение в других областях, оценить эффективность и указать способы ее повышения и т. п.).

В случае спорной оценки может быть дано дополнительное задание, в том числе не связанное напрямую с вопросами билета.

Для успешной сдачи экзамена необходимо иметь конспект лекций. Учебное пособие следует использовать в качестве консультанта по неясным вопросам. С другой стороны, в учебном пособии могут отсутствовать важные примеры, приводимые на лекциях;

Конечная оценка по дисциплине формируется в соответствии с правилами системы РИТМ.

Все расчеты по формулам проводятся по нормированным баллам: минимум – 40, максимум – 60.

На последней учебной неделе семестра обучающийся, полностью выполнивший программу, т.е. набравший не менее 40 баллов, допускается к итоговому контрольному испытанию (опрос). Итоговое контрольное задание максимально оценивается в 20 баллов.

Обучающиеся, набравшие 14 и более баллов по итогам контрольного испытания, освобождаются от экзамена. Суммарный балл, на основании которого выставляется экзаменационная оценка по четырех балльной шкале, определяется по формуле

$$N^C = N_{\text{обяз}} + N_{\text{доп}} + N_{\text{ск}}$$

где $N_{\text{обяз}}$ – баллы за обязательные виды работ, $N_{\text{доп}}$ – баллы за дополнительные работы, $N_{\text{ск}}$ – количество баллов по итогам семестрового контроля, N^C – суммарный балл.

Дополнительные баллы начисляются: за подготовку мини-доклада – 10 баллов.

Любой обучающийся, участвующий в системе РИТМ, имеет право освобождения от экзамена при условии, что он выдержал итоговый семестровый контроль.

Обучающиеся, освобожденные от экзамена, могут сдавать экзамен с целью повышения суммарного балла. В этом случае им гарантируется оценка, полученная по итогам работы в семестре.

Обучающиеся, набравшие на контрольном испытании менее 14 баллов, остаются участниками системы РИТМ и сдают экзамен.

Балл N_{Σ} за экзамен определяется в интервале от 20 до 40 баллов с учетом качества ответа.

Для всех обучающихся, сдающих экзамен, итоговый балл, при положительной оценке на экзамене, определяется по формуле

$$N^C = (N_{\text{обяз}} + N_{\text{доп}}) + N_{\text{экз.}}$$

Накопленные баллы (расчетные единицы) переводятся в четырех балльную шкалу:

«отлично»	- 90 и более
«хорошо»	-от 75 – до 89
«удовлетворительно»	- от 50 – до 74
«неудовлетворительно»	- менее 50.

Обучающиеся, не набравшие за текущую работу в семестре 40 баллов, считаются выбывшими из РИТМа и обязаны сдавать экзамен после того, как выполнят всю запланированную работу.

Обучающийся, который погашает свои задолженности после окончания сессии, получает минимальный балл за оценку, выставленную на экзамене за вычетом 7 баллов:

«отлично»	- 83
«хорошо»	- 68
«удовлетворительно»	- 43

Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющих в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ		
1.	Галочкин, Владимир Иванович. Структуры и алгоритмы обработки данных [Текст] : учеб. пособие / В. И. Галочкин. Йошкар-Ола: МарГТУ, 2004. - 147 с. ISBN 5-8158-0350-2. Экземпляры: всего 58.	58
2.	Галочкин, Владимир Иванович. Базы данных [Текст] : учеб. пособие / В. И. Галочкин. Йошкар-Ола: МарГТУ, 2009. - 199 с. ISBN 978-5-8158-0688-7. Экземпляры: всего 90.	90
3.	Галочкин, Владимир Иванович. Алгоритмы и программы [Текст] : задачи повышенной сложности : учеб. пособие / В. И. Галочкин; М-во образования и науки Рос. Федерации, ФГБОУ ВПО "Мар. гос. техн. ун-т". Йошкар-Ола: МарГТУ, 2012. - 207 с. ISBN 978-5-8158-0968-0. Экземпляры: всего 88.	88 / https://portal.volgatech.net/books/Galochkin_Algoritmy_i_programmy.pdf
4.	Вирт, Н. Алгоритмы и структура данных [Текст] / Н. Вирт; пер. с англ. Р. П. Подшивалова. Москва: Мир, 1989. - 360 с.:ил. с. ISBN 5-03-001045-9. Экземпляры: всего 3.	3

6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	522 (I)	Кондиционер AEG ACS-09HR (1), ПК RAMEC GALE LCD LG 23"/Intel i5 4590/MSI B85M-E45/2x4DDR3/GT740 2Gb/500Gb/клав,мышь (28), Принтер Canon LBP 2900 лазерный с кабелем (1), Проектор мультимедийный Hitachi CP-EX250 (1), Проектор мультимедийный Hitachi CP-EX251N (1), Комплект учебной мебели (1)	Microsoft Office Standard, Агент Dr.Web, Комплект ПО для решения основных пользовательских задач

Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
- умение применять теоретические знания при решении практических заданий.

Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает затруднения в выполнении практических заданий.	удовлетворительно
Продвинутый уровень	Обучающийся твердо знает программный материал, излагает его грамотно и по существу, не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения	хорошо
Высокий уровень	Обучающийся глубоко и прочно усвоил программный материал, грамотно и логически стройно его излагает, дает исчерпывающие ответы на поставленные вопросы. В ответе тесно увязывается теория с практикой, при этом обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, вопросами и другими видами применения знаний, показывает знакомство с монографической литературой, периодическими изданиями, правильно обосновывает принятые решения, свободно владеет разносторонними навыками, приемами выполнения практических работ	отлично

7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

1. В оперативной памяти имеется двухсвязный стек из элементов, имеющих следующую структуру:

```

Type
ukaz=^elem;
elem=record
    nom: integer;
    next: ukaz; { указатель на следующий элемент }

```

```
pred: ukaz { указатель на предыдущий элемент }  
end;
```

Указатель KON показывает на некоторый элемент. Удалить этот элемент из списка и из памяти.

2. Описать структуру данных для кодировки дерева общего вида с помощью бинарного дерева. Задана некоторая вершина общего дерева. Как на бинарном дереве выделить самого правого сына заданной вершины ?

3. Имеется сеть однокорейных железных дорог, по каждому участку которой поезда движутся только в одном направлении. Выбрать структуру данных для описания сети. Произвести корректировку данных для случая изменения направлений движения всех поездов на противоположные.

4. Массив целых чисел расположен по убыванию. Задается некоторый номер. Требуется методом бинарного поиска вставить данный номер в массив, если его нет, не нарушая порядка сортировки.

5. Дан следующий массив: 17 25 14 34 11 3 47 16 1 31 22. Показать этапы сортировки массива методом простого слияния.

Перечень вопросов для проведения промежуточной аттестации

Пороговый уровень

1. Имеется кольцевая очередь, заданная массивом, и стек, заданный с помощью указателей. Информационная часть элементов обеих структур идентична. Включить очередь в стек в порядке от конца к началу.
2. Имеется стек, заданный массивом, и очередь, заданная с помощью указателей. Информационная часть элементов обеих структур идентична. Исключить из очереди элементы, имеющиеся в стеке.
3. Имеется кольцевая очередь, заданная массивом, и стек, заданный с помощью указателей. Информационная часть элементов обеих структур идентична. Включить последовательно элементы стека от его вершины до конца в начало очереди.
4. Имеется кольцевая очередь, заданная массивом, и стек, заданный с помощью указателей. Информационная часть элементов обеих структур идентична. Исключить из стека элементы, имеющиеся в очереди.
5. Описать на ПАСКАЛЕ либо на C++ с помощью указателей стек, состоящий из элементов типа запись (структуру записи выбрать произвольной). Считая, что стек заполнен, удалить из памяти весь стек.

Продвинутый уровень

6. Имеется сильно ветвящееся дерево, представленное с помощью бинарного. В исходном дереве добавляется самый правый сын некоторой вершины, указатель на которую в бинарном дереве известен. Написать программу для корректировки бинарного дерева.
7. Описать структуру данных для кодировки дерева общего вида. Составить программу определения числа вершин, не являющихся листьями.

8. Имеется сильно ветвящееся дерево, представленное с помощью бинарного. Вершины дерева пронумерованы. Написать программу увеличения на 100 всех номеров вершин бинарного дерева, соответствующих листьям исходного дерева.

9. Задана матрица смежности ориентированного графа из 10 вершин. Написать программу нахождения матрицы $A(10 \times 10)$, в которой элемент $A[i,j]$ равен количеству путей длины не более 2 (по числу звеньев), ведущих из i -ой вершины в j -ю. Считать, что граф уже находится в оперативной памяти.

10. В памяти имеется бинарное дерево поиска. Требуется составить процедуру, которая по заданному имени находит вершину, соответствующую данному имени, и возвращает указатель на эту вершину. Если заданное имя отсутствует в дереве - вернуть NIL.

Высокий уровень

11. Дан следующий массив: 17, 25, 14, 34, 11, 3, 47, 16, 1, 31. Требуется показать в виде деревьев первые 3 шага метода турнирной сортировки. Описать шаги алгоритма.

12. Описать структуру данных для кодировки дерева общего вида с помощью бинарного дерева. Задана некоторая вершина общего дерева. Как на бинарном дереве выделить самого правого сына заданной вершины? Описать алгоритм, разработать программу.

13. Имеется сеть однопутных железных дорог, по каждому участку которой поезда движутся только в одном направлении. Выбрать структуру данных для описания сети. Произвести корректировку данных для случая изменения направлений движения всех поездов на противоположные. Описать алгоритм, разработать программу.

14. Массив целых чисел расположен по убыванию. Задается некоторый номер. Требуется методом бинарного поиска вставить данный номер в массив, если его нет, не нарушая порядка сортировки. Описать алгоритм, разработать программу.

15. Дан следующий массив: 17, 25, 14, 34, 11, 3, 47, 16, 1, 31, 22. Показать этапы сортировки массива методом простого слияния. Разработать программу.