

Министерство просвещения РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Глазовский государственный инженерно-педагогический университет
имени В.Г. Короленко»

Утверждена
на заседании ученого совета университета

«21» апреля 2025 г. протокол № 9
Приказ № 45 от 21 апреля 2025 г.

Ректор Я.А. Чиговская-Назарова

**АДАПТИРОВАННАЯ РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
(для лиц с нарушениями функций опорно-двигательного аппарата)**

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Уровень основной профессиональной образовательной программы	Бакалавриат
Направление подготовки	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Информатика и вычислительная техника
Форма обучения	Очная
Семестр(ы)	5, 6

Глазов 2025

1. Цель и задачи изучения дисциплины

1.1. Цель и задачи изучения дисциплины

Цель изучения дисциплины – формирование компетенций у обучающихся, связанных со способностью разрабатывать алгоритмы и программы, пригодные для практического применения; разрабатывать требования и проектировать программное обеспечение, а также компоненты системных программных продуктов в процессе изучения технологий программирования.

Задачи изучения дисциплины:

- сформировать знания об алгоритмических языках программирования, операционных системах и оболочках, современных средах разработки программного обеспечения; синтаксисе, особенностях программирования и стандартных библиотеках выбранного языка программирования, структуре объектных и исполняемых файлов в операционной системе; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения в рамках определенной технологии программирования;
- сформировать умения составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули; использовать существующие типовые решения и шаблоны проектирования программного обеспечения; использовать коммерческие операционные системы, создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов в рамках определенной технологии программирования;
- сформировать умения составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули; использовать существующие типовые решения и шаблоны проектирования программного обеспечения; использовать коммерческие операционные системы, создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов в рамках определенной технологии программирования; сформировать владение языком программирования; навыками отладки и тестирования работоспособности программы, методами и средствами проектирования программного обеспечения; средствами разработки компонентов системных программных продуктов в рамках определенной технологии программирования.

Программа адаптирована для лиц с нарушениями опорно-двигательного аппарата (ОДА) с учетом их психофизического развития, индивидуальных возможностей и необходимых специальных условий обучения.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными индикаторами достижения компетенций

Код компетенции	ОПК-8
Формулировка компетенции	Способен разрабатывать алгоритмы и программы, пригодные для практического применения;
Индикатор достижения компетенции	ИОПК 8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения ИОПК 8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули ИОПК 8.3 Владеть: языком программирования; навыками

	отладки и тестирования работоспособности программы
Код компетенции	ПК-2
Формулировка компетенции	Способен разрабатывать требования и проектировать программное обеспечение
Индикатор достижения компетенции	ИПК 2.1 Знает: типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения ИПК 2.2 Умеет: использовать существующие типовые решения и шаблоны проектирования программного обеспечения ИПК 2.3 Владеет: методами и средствами проектирования программного обеспечения

Код компетенции	ПК-7
Формулировка компетенции	Способен разрабатывать компоненты системных программных продуктов
Индикатор достижения компетенции	ИПК 7.1 Знает: синтаксис, особенности программирования и стандартные библиотеки выбранного языка программирования, структуру объектных и исполняемых файлов в операционной системе ИПК 7.2 Умеет: использовать коммерческие операционные системы, создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов ИПК 7.3 Владеет: средствами разработки компонентов системных программных продуктов

1.3. Воспитательная работа

Направление воспитательной работы	Типы задач	Формы работы
формирование у обучающихся осознания социальной значимости своей будущей профессии, мотивации к осуществлению профессиональной деятельности	производственно-технологический	включение в социокультурную среду путем формирования у студентов практических умений и навыков в рамках профессиональной деятельности
научно-исследовательская работа обучающихся	производственно-технологический	Исследовательская деятельность студентов (публикация статей, выступление с докладом)

1.4. Место дисциплины в структуре образовательной программы

Дисциплина «Технологии программирования» относится к обязательной части учебного плана.

Для её успешного изучения необходимы знания, умения и навыки, приобретенные в результате освоения дисциплин «Язык программирования C++», «Язык программирования C#», «Язык программирования Python».

1.5. Особенности реализации дисциплины

Дисциплина реализуется на русском языке.

Для лиц с нарушениями функций ОДА используется электронное обучение, дистанционные технологии. Для поддержки курса используется сайт: <http://moodle.ggpi.org>.

2. Объем дисциплины

Вид учебной работы по семестрам	Всего, зачетных единиц	Академ. часы	Из них в форме практической подготовки
Общая трудоемкость дисциплины	5	180	
СЕМЕСТР 5			
Контактная работа с преподавателем:			
Аудиторные занятия (всего)		36	
Занятия лекционного типа		16	
Лабораторные работы		-	
Занятия семинарского типа		-	
Практические занятия		16	
КСР		4	
Самостоятельная работа обучающихся		36	
Вид промежуточной аттестации: Зачет		0	
СЕМЕСТР 6			
Контактная работа с преподавателем:			
Аудиторные занятия (всего)		36	
Занятия лекционного типа		14	
Лабораторные работы		-	
Занятия семинарского типа		-	
Практические занятия		14	
КСР		8	
Самостоятельная работа обучающихся		36	
Вид промежуточной аттестации: Экзамен		36	

3. Содержание дисциплины

3.1. Разделы дисциплины и виды занятий (тематический план занятий)

№ п/п	Разделы и темы дисциплины Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в академических часах)						
		всего	ауд	лекц	прак	Лаб.	КСР	СРС
Семестр 5								
1.	Основные конструкции языка Пролог	8	4	2	2	–	–	4
2.	Арифметика Пролога	8	4	2	2	–	–	4
3.	Рекурсия. Списки	8	4	2	2	–	–	4
4.	Логические задачи	8	4	2	2	–	2	4
5.	Программирование с помощью функций и процедур	8	4	2	2	–	–	4

6.	Программирование в функциональных обозначениях	8	4	2	2	–	–	4
7.	Строго функциональный язык	12	6	2	2	–	–	6
8.	Представление и интерпретация функциональных программ	12	6	2	2		2	6
Всего – по семестру (ам)		72	36	16	16	–	4	36

№ п/п	Разделы и темы дисциплины Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в академических часах)						
		всего	ауд	лекц	прак	Лаб.	КСР	СРС
Семестр 6								
1.	Понятие параллельной вычислительной системы	8	4	2	2	–	–	4
2.	Граф алгоритма и параллельные вычисления. Внутренний параллелизм программ	8	4	2	2	–	–	4
3.	Параллельное программирование в стандарте OpenMP	8	4	2	2	–	–	4
4.	Параллельное программирование в стандарте MPI	12	6	2	2		2	6
5.	Проектирование параллельных алгоритмов	12	6	2	2		2	6
6.	Оценка эффективности параллельных алгоритмов	12	6	2	2	–	2	6
7.	Технологии параллельного программирования	12	6	2	2		2	6
Экзамен		36	–	–	–	–	–	–
Итого – по дисциплине		108	36	14	14	–	8	36

3.2. Занятия лекционного типа

Для лиц с нарушениями функций ОДА лекция сопровождается текстом с увеличенным шрифтом или усиливающей звуковой аппаратурой.

Занятия, при возможности, проводятся в мультимедийной аудитории, где имеется возможность подкрепления основных положений лекционного материала необходимым иллюстративным материалом (письменная презентация ключевых вопросов, являющихся темой обсуждения во время беседы; использование необходимых электронных видеоматериалов для иллюстрирования вопросов и контекста обсуждаемой проблемы, и т.п.). Есть возможность предоставлять необходимый учебный материал электронно для последующей самостоятельной работы с ним.

При объяснении материала мысли излагаются четко и лаконично (в простые предложения), информация подается в виде небольших логически и по смыслу законченных фрагментов.

СЕМЕСТР 5

Лекция 1.

Тема: Основные конструкции языка Пролог

Краткая аннотация к лекции. Общие сведения о структуре языка логического программирования. Алгоритм выполнения программ на Прологе. Основные конструкции языка: факт, правило, вопрос (запрос). Виды запросов.

Лекция 2.

Тема: Арифметика Пролога

Краткая аннотация к лекции. Арифметика в языке логического программирования. Арифметические операции. Предикат отсечения и управление логическим выводом в программах. Реализация условной конструкции.

Лекция 3.

Тема: Рекурсия. Списки

Краткая аннотация к лекции. Понятие рекурсии. Рекурсия как метод организации повторяющихся действий в Прологе. Достоинства и недостатки рекурсии. Создание рекурсивных правил. Примеры решения задач с использованием рекурсии. Метод половинного деления, наибольшее общее кратное. Списки. Встроенные предикаты работы со списками. Примеры задач. Список как рекурсивный объект данных в Прологе. Стандартные задачи обработки списков.

Лекция 4.

Тема: Логические задачи

Краткая аннотация к лекции. Подходы к решению логических задач. Решение задач с помощью таблиц. Решение задач на расположение в ряд. Примеры решения логических задач. Решение задач на круговые расстановки. Решения задач со списками. Примеры решения логических задач.

Лекция 5.

Тема: Программирование с помощью функций и процедур.

Краткая аннотация к лекции. Рекурсивные функции и лямбда-исчисление А. Черча. Математическое представление рекурсивных функций и λ -исчисления. Применение рекурсивных функций и λ -исчисления в функциональном программировании.

Лекция 6.

Тема: Программирование в функциональных обозначениях.

Краткая аннотация к лекции. Программирование в функциональных обозначениях. Функциональные языки

Лекция 7.

Тема: Строго функциональный язык.

Краткая аннотация к лекции. Строго и нестрого функциональные языки. Элементарные понятия: функции, типа, класса, аккумулятора. Приемы программирования. Примеры создания программ, описания различных функций.

Лекция 8.

Тема: Представление и интерпретация функциональных программ

Краткая аннотация к лекции. Представление и интерпретация функциональных программ. Соответствие между функциональными и императивными программами. Применения функционального программирования.

СЕМЕСТР 6

Лекция 1

Тема. Понятие параллельной вычислительной системы

Краткая аннотация к лекции. Большие задачи. Параллельная обработка. Виды параллелизма. Многопроцессорная обработка. Конвейерная обработка. Векторная обработка. Векторно-конвейерная обработка. Архитектура параллельных вычислительных систем. Классификация Флинна. Классификация MIMD систем. Системы с общей и системы с распределенной оперативной памятью. SMP системы, их преимущества и недостатки. Преимущества и проблемы вычислительных кластеров (COW системы).

Лекция 2

Тема. Граф алгоритма и параллельные вычисления. Внутренний параллелизм программ

Краткая аннотация к лекции. Внутренний параллелизм. Параллельная форма алгоритма. Эквивалентные преобразования. Задача – метод – алгоритм – программа. Граф алгоритма. Теорема об ошибках округления. Графы зависимостей. Минимальные графы зависимостей. Линейный класс программ. Фундаментальная теорема. Примеры типовых структур.

Лекция 3

Тема. Параллельное программирование в стандарте OpenMP

Стандарт OpenMP. Структура OpenMP-программы. Директивы, функции библиотеки OpenMP. Переменные окружения OpenMP. Область видимости переменных. Директивы синхронизации. Редукция операций. Параллельные секции.

Лекция 4

Тема. Параллельное программирование в стандарте MPI

Краткая аннотация к лекции. Модель передачи сообщений. Модель выполнения SPMD. Модель выполнения MPMD. Стандарт MPI. Цикл разработки MPI программ. Понятия: процессы, коммутаторы, сообщения. Структура MPI-программы. MPI-функции. Виды взаимодействия процессов: «точка-точка», коллективное взаимодействие. Виды коммуникационных функций «точка-точка». Отправка сообщений при использовании функций «точка-точка». Тупики и их разрешение. Барьерная синхронизация процессов. Стандартные и пользовательские типы данных в MPI. Модель программирования в общей памяти. Модель Fork – Join.

Лекция 5

Тема. Проектирование параллельных алгоритмов

Краткая аннотация к лекции. Моделирование параллельных программ. Модель «процессы-каналы». Методология проектирования. Разбиение по данным. Функциональная декомпозиция. Проектирование связей. Представление алгоритма. Агрегирование. Привязка.

Лекция 6

Тема. Оценка эффективности параллельных алгоритмов

Показатели эффективности параллельного алгоритма: ускорение, эффективность, стоимость. Оценка максимально достижимого параллелизма. Закон Амдала и его следствия. Анализ масштабируемости параллельных вычислений. Примеры. Способы вычислений времени выполнения алгоритма.

Лекция 7

Тема. Технологии параллельного программирования

Критерии выбора технологии. Классификация технологий. Параллельные расширения. Параллельные языки. Интерфейсы параллельного программирования. Специализированные параллельные библиотеки. Инструментальные системы разработки. Средства автоматического распараллеливания. Распараллеливающие компиляторы. Специализированные прикладные пакеты.

3.3. Занятия семинарского типа

Учебным планом не предусмотрены

3.4. Практические занятия

Для лиц с нарушениями функций ОДА материал в электронном виде можно найти по адресу: <http://moodle.ggpi.org>.

Выполнение практических работ проводятся в микрогруппах или парами, в которых присутствует смешанный состав обучающихся: в паре – один обычный обучающийся и один обучающийся с двигательным нарушением; микрогруппа включает одного обучающегося с двигательным нарушением и несколько обычных обучающихся.

В ходе практического занятия используются следующие методы:

- опора на определенные и точные понятия;
- использование для иллюстрации конкретных примеров;
- применение вопросов для мониторинга понимания;
- разделение изучаемого материала на небольшие логические блоки;
- увеличение доли конкретного материала и соблюдение принципа от простого к сложному при объяснении материала.

СЕМЕСТР 5

Практическое занятие 1.

Тема: Основные конструкции языка Пролог

Перечень заданий: Решение задач по теме «Общие сведения о языке логического программирования» с использованием фактов. Решение задач по теме «Общие сведения о языке логического программирования» с использованием правил.

Практическое занятие 2.

Тема: Арифметика Пролога

Перечень заданий: Решение задач по теме «Арифметика. Управление логическим выводом в программах»

Практическое занятие 3.

Тема: Рекурсия

Перечень заданий: Решение задач по теме «Повторение и рекурсия». Решение задач с использованием встроенных предикатов работы со списками.

Решение задач по теме «Применение рекурсии для обработки списков»

Практическое занятие 4.

Тема: Логические задачи

Перечень заданий: Решение задач по теме «Решение логических задач» с использованием таблиц, расположение в ряд. Решение задач по теме «Решение логических задач» с использованием списков

Практическое занятие 5.

Тема: Программирование с помощью функций и процедур

Перечень заданий: Конструирование простейших функций.

Практическое занятие 6.

Тема: Программирование в функциональных обозначениях

Перечень заданий: Конструирование функций в префиксной и инфиксной нотации.

Практическое занятие 7.

Тема: Строго функциональный язык

Перечень заданий: Знакомство с Hugs-98. Конструирование простейших функций. Применение выражений охраны при конструировании функций. Конструирование функций в префиксной и инфиксной нотации. Структуры данных. Конструирование списков.

Практическое занятие 8.

Тема: Представление и интерпретация функциональных программ

Перечень заданий: Применение выражений охраны при конструировании функций. Конструирование функций с применением аккумулятора. Полиморфизм, пользовательские типы и множественный конструктор.

СЕМЕСТР 6

Практическое занятие 1

Тема. Понятие параллельной вычислительной системы

Перечень заданий

Напишите параллельную программу, реализующую скалярное произведение двух векторов.

Вычислительные процессы. Формальная модель. Свойства процесса.

Система процессов. Независимые процессы. Взаимодействие процессов.

Напишите параллельную программу, реализующую поиск максимального значения вектора.

Классы параллельных ВС.

Отображение системы процессов. Синхронизация.

Практическое занятие 2

Тема. Граф алгоритма и параллельные вычисления. Внутренний параллелизм программ

Перечень заданий

Напишите параллельную программу, реализующую транспонирование матрицы $n \times n$.

Реализуйте параллельный алгоритм метода Гаусса решения систем линейных алгебраических уравнений.

На основе программы для решения СЛАУ методом Гаусса напишите параллельную программу для вычисления определителя и постройте зависимость коэффициента ускорения от числа ядер (потоков) на узле.

На основе программы для решения СЛАУ методом Гаусса напишите параллельную программу для вычисления обратной матрицы и постройте зависимость коэффициента ускорения от числа ядер (потоков) на узле.

Практическое занятие 3

Тема. Параллельное программирование в стандарте OpenMP

Перечень заданий

Напишите параллельную программу реализовывающую метод Гаусса (метод элементарных преобразований) для вычисления ранга произвольной матрицы $A - m \times n$.

OpenMP: Общие понятия. Область применения.

Директивы OpenMP: Директивы для определения параллельной области.

OpenMP: Директивы для распределения вычислений внутри параллельной области: Директива for.

Найти численное решение задачи Дирихле для уравнения Пуассона итерационным методом Зейделя.

OpenMP: Директивы для распределения вычислений внутри параллельной области: Директива sections. Директива single.

OpenMP: Директивы для распределения вычислений внутри параллельной области: Объединение директив parallel и for (sections).

Практическое занятие 4

Тема. Параллельное программирование в стандарте MPI

Перечень заданий

Создать 2 вектора с 1000 элементами из случайных чисел. Сложить эти вектора. (MPI)

MPI: Стандарт MPI. Общие понятия. Области связи и коммутаторы.

Общие процедуры MPI: Инициализация, Завершение.

Общие процедуры MPI: Определение общего числа параллельных процессов в группе.

Определение номера процесса в группе.

Создать 2 вектора с 1000 элементами из случайных чисел. Скалярно умножить эти вектора. Сравнить время счета параллельного и последовательного алгоритмов.

MPI: Прием/передача сообщений между отдельными процессами.

MPI: Групповые (коллективные) взаимодействия. Рассылка целого сообщения процессам. Сборка данных от процессов.

Практическое занятие 5

Тема. Проектирование параллельных алгоритмов

Перечень заданий

Дан двумерный массив размерности $n \times m$. Найти максимальное и минимальное значение массива.

Дан двумерный массив размерности (n, m) , где $n > 10$, $m > 5$. Найти сумму диагональных элементов и сумму всех элементов матрицы.

Выполнить транспонирование матрицы $n \times n$.

Дан двумерный массив размерности (n, m) , где $n > 10$, $m > 5$. Отсортировать данный массив.

Практическое занятие 6

Тема. Оценка эффективности параллельных алгоритмов

Перечень заданий

Вычислить одномерные определенные интегралы от трёх разных подынтегральных функций (Например: $\sin x$, $1 + 2e^x$, $x^2 + 4x + 3$).

MPI: Групповые (коллективные) взаимодействия: Рассылка частей сообщения процессам. Сборка частей сообщения с процессов.

Схема параллельного выполнения алгоритма.

Сгенерировать три вектора a , b и c размером 100 из случайных чисел. Вычислить сумму векторов a и b , затем найденную сумму скалярно умножить на вектор c .

MPI: Функции поддержки распределенных операций: Выполнение глобальных операций с возвратом результатов в главный процесс. Выполнение глобальных операций с возвратом результатов во все процессы.

Определение времени выполнения параллельного алгоритма.

Теоретические оценки времени выполнения параллельных алгоритмов.

Практическое занятие 7

Тема. Технологии параллельного программирования

Перечень заданий

Формула скалярного умножения a и b : $a_1 * b_1 + a_2 * b_2 + a_3 * b_3$.

Даны трехмерные вектора a , b и c . Найти скалярное произведение a и b , векторное произведение a и c , смешанное произведение a , b и c .

МРІ: Синхронизация процессов.

Показатели эффективности параллельного алгоритма.

Оценка максимально достижимого параллелизма. Анализ масштабируемости параллельных вычислений.

Формула векторного произведения a и c , вектор c компонентами: $(a_2 * c_3 - a_3 * c_2, a_3 * c_1 - a_1 * c_3, a_1 * c_2 - a_2 * c_1)$.

Формула смешанного произведения a , b и c : сначала векторное произведение a и b , затем результат скалярно умножаем на c .

Процессы и потоки.

Закон Амдала.

3.5. Лабораторные работы

Учебным планом не предусмотрены

3.6. Контроль самостоятельной работы

Для лиц с нарушениями функций ОДА материал в электронном виде можно найти по адресу: <http://moodle.ggpi.org>.

Для лиц с нарушениями функций опорно-двигательного аппарата учебно-методическое обеспечение для контроля самостоятельной работы обучающихся по дисциплине предъявляется (по выбору обучающегося): устно, письменно на бумаге или на компьютере, в форме тестирования, электронных тренажеров и т.п.

Конкретные формы и виды самостоятельной работы обучающихся с нарушениями функций ОДА устанавливаются преподавателем с учетом индивидуальных психофизических особенностей. При необходимости обучающимся предоставляется дополнительное время для консультаций и выполнения заданий.

Самостоятельная работа включает следующие виды деятельности: работа с книгой и другими источниками информации, планы-конспекты; реферативные (воспроизводящие), реконструктивно-вариативные, эвристические, творческие самостоятельные работы; проектные работы; дистанционные технологии.

Уделяется внимание индивидуальной работе. Под индивидуальной работой подразумевается две формы взаимодействия с преподавателем: индивидуальная учебная работа (консультации), т.е. дополнительное разъяснение учебного материала и углубленное изучение материала с теми обучающимися, которые в этом заинтересованы, и индивидуальная воспитательная работа. Индивидуальные консультации по предмету становятся важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся с нарушениями функций ОДА.

СЕМЕСТР 5

Контроль самостоятельной работы 1.

Тема: Логические задачи

Перечень заданий: Решение задач на строковые списки. Решение логических задач

Контроль самостоятельной работы 2.

Тема: Соответствие между функциональными и императивными программами.

Перечень заданий: Решение задач по теме, выполнение контрольной работы

СЕМЕСТР 6

Контроль самостоятельной работы 1

Тема. Параллельное программирование в стандарте MPI

Перечень заданий

MPI: Прием/передача сообщений между отдельными процессами.

MPI: Групповые (коллективные) взаимодействия. Рассылка целого сообщения процессам. Сборка данных от процессов.

MPI: Групповые (коллективные) взаимодействия: Рассылка частей сообщения процессам. Сборка частей сообщения с процессов.

MPI: Функции поддержки распределенных операций: выполнение глобальных операций с возвратом результатов в главный процесс.

Контроль самостоятельной работы 2

Тема. Проектирование параллельных алгоритмов

Перечень заданий

OpenMP: Директивы для распределения вычислений внутри параллельной области: Директива sections. Директива single.

OpenMP: Директивы для распределения вычислений внутри параллельной области: Объединение директив parallel и for (sections).

Классическая задача об обедающих философах.

Задача «читатели и писатели».

Контроль самостоятельной работы 3

Тема. Оценка эффективности параллельных алгоритмов

Перечень заданий

Показатели параллельных ВС и вычислений.

Модель вычислений в виде графа «операции-операнды».

Схема параллельного выполнения алгоритма.

Определение времени выполнения параллельного алгоритма.

Теоретические оценки времени выполнения параллельных алгоритмов.

Показатели эффективности параллельного алгоритма.

Оценка максимально достижимого параллелизма. Анализ масштабируемости параллельных вычислений.

Контроль самостоятельной работы 4

Тема. Технологии параллельного программирования

Перечень заданий

Процессы и потоки.

Закон Амдала.

MPI: Стандарт MPI. Общие понятия. Области связи и коммутаторы.

Общие процедуры MPI: Инициализация, Завершение.

Общие процедуры MPI: Определение общего числа параллельных процессов в группе.

Определение номера процесса в группе.

3.7. Самостоятельная работа студентов

Рекомендуемые формы самостоятельной работы студентов: закрепление материала по конспекту лекции, подготовка к практическим занятиям, подготовка презентаций к докладам, подготовка к различным формам промежуточной и итоговой аттестации.

4. Фонд оценочных средств

Формы текущего контроля, промежуточной аттестации и поститоговый контроль для лиц с нарушениями функций ОДА устанавливаются с учетом их психофизиологических особенностей. При необходимости все виды аттестации проходит в несколько этапов.

Текущий контроль результатов обучения осуществляется преподавателем в процессе проведения практических занятий и лабораторных работ, а также выполнения индивидуальных работ и домашних заданий, или в режиме тренировочного тестирования в целях получения информации о выполнении обучаемым требуемых действий в процессе учебной деятельности; правильности выполнения требуемых действий; соответствии формы действия данному этапу усвоения учебного материала; формировании действия с должной мерой обобщения, освоения и т.д.

Формы и сроки проведения промежуточного контроля определяются преподавателем с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). Для лиц с нарушениями функций опорно-двигательного аппарата формами текущего контроля, промежуточной аттестации и поститогового контроля используются (в зависимости от индивидуальных особенностей и потребностей):

- устный ответ;
- письменный ответ;
- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

При проведении всех форм контроля учитываются психофизическое развитие и ограничения здоровья. Время выполнения заданий для лиц с нарушениями функций ОДА может быть увеличено, но не более чем на 30 минут.

Для лиц с нарушениями опорно-двигательного аппарата материалы ко всем видам аттестации предъявляться (в зависимости от индивидуальных особенностей и потребностей):

- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

Рекомендуемые формы контроля и оценки результатов обучения лиц с нарушением функций ОДА:

- письменная проверка с использованием специальных технических средств (альтернативных средств ввода, управления компьютером и др.): контрольные, графические работы, тестирование, домашние задания, эссе, письменные коллоквиумы, отчеты и др.;
- устная проверка, с использованием специальных технических средств (средств коммуникаций): дискуссии, тренинги, круглые столы, собеседования, устные коллоквиумы и др.;
- с использованием компьютера и специального ПО (альтернативных средств ввода и управления компьютером и др.): работа с электронными образовательными ресурсами, тестирование, рефераты, курсовые проекты, графические работы, дистанционные формы предпочтительнее обучающимся, ограниченным в передвижении и др.

ФОС включает оценочные средства текущего, промежуточного и поститогового контроля (Приложение 1).

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

5.1. Основная литература

1. Алексеев, А. А. Основы параллельного программирования с использованием Visual Studio 2010 : учебное пособие / А. А. Алексеев. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 330 с. — ISBN 978-5-4497-0341-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89456.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
2. Антонов, А. С. Параллельное программирование с использованием технологии MPI : учебное пособие / А. С. Антонов. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 83 с. — ISBN 978-5-4497-0934-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/146378.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
3. Ездаков, А. Л. Функциональное и логическое программирование [Текст]: учеб. пособие / А. Л. Ездаков. - М.: БИНОМ. Лаборатория знаний, 2009. - 119 с. 26 экз.
4. Казанский, А. А. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 192 с. — (Высшее образование). — ISBN 978-5-534-12338-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/451467> (дата обращения: 31.03.2025).
5. Левин, М. П. Параллельное программирование с использованием OpenMP : учебное пособие / М. П. Левин. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 133 с. — ISBN 978-5-4497-3330-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/142293.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
6. Новиков, П. В. Логическое программирование : учебно-методическое пособие к лабораторным работам / П. В. Новиков. — Саратов : Вузовское образование, 2017. — 103 с. — ISBN 978-5-4487-0010-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/66314.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. Пользователей
7. Прыкина, Е. Н. Основы логического программирования в среде Турбо Пролог : учебное пособие по курсу «Экспертные системы» / Е. Н. Прыкина. — Кемерово : Кемеровский государственный институт культуры, 2006. — 68 с. — ISBN 5-8154-0130-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/22048.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
8. Рогозин, О. В. Функциональное и рекурсивно-логическое программирование : учебное пособие / О. В. Рогозин. — Москва : Евразийский открытый институт, 2009. — 139 с. — ISBN 978-5-374-00182-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/11119.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
9. Якимов, С. П. Структурное программирование : учебник для вузов / С. П. Якимов. — Москва : Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-14885-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567948> (дата обращения: 31.03.2025).

5.2. Дополнительная литература

1. Городняя, Л. В. Основы функционального программирования : учебное пособие / Л. В. Городняя. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 246 с. — ISBN 978-5-4497-0932-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/146377.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей

2. Салмина, Н. Ю. Функциональное программирование и интеллектуальные системы : учебное пособие / Н. Ю. Салмина. — Томск : Томский государственный университет систем управления и радиоэлектроники, 2016. — 100 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/72216.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей

1. Обучающиеся с нарушениями функций опорно-двигательного аппарата обеспечены печатными и электронными ресурсами в форме, адаптированной к ограниченным возможностям здоровья и восприятия информации:

- в печатной форме
- в форме электронного документа
- в форме аудиофайла

2. Каждому обучающемуся с нарушениями функций ОДА обеспечен доступ к библиотечным ресурсам и сети Интернет и предоставлен не менее чем одним учебным, методическим и (или) электронным изданием в форме, адаптированной к ограничениям здоровья.

3. Для обучения лиц с нарушениями функций ОДА комплектация библиотечного фонда осуществляется электронными изданиями основной и дополнительной литературы по дисциплинам.

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине

6.1 Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины

1. <https://www.intuit.ru/> - Национальный Открытый Университет «ИНТУИТ»
2. <http://www.gprolog.org/> - Официальный сайт GnuProlog
3. <http://www.haskell.ru/> - официальный сайт Haskell
4. <https://wiki.haskell.org/Ru/> Справочник_по_языку_Haskell - Ru / Справочник по языку Haskell
5. parallel.ru – Информационно-аналитический центр по параллельным вычислениям.
6. www.hpcc.unn.ru/multicore/programm.html – Развитие в Нижегородском университете межфакультетской магистратуры по системному и прикладному программированию для многоядерных компьютерных систем
7. emc.orgfree.com/e14 – Многопроцессорный учебный компьютер «E14»
8. compscicenter.ru/courses/video_cards_computation – Вычисления на видеокартах
9. metanit.com/c/tutorial/11.1.php – Руководство по языку программирования Си. Многопоточность
10. ulearn.me/course/basicprogramming2/4dd557b3-7903-4ff4-932a-a95b4a7fb15f – Основы программирования на примере C#. Часть 2. Многопоточное программирование
11. www.geeksforgeeks.org/multithreading-python-set-1 – Многопоточность в Python

12. www.internet-technologies.ru/articles/threading-upravlenie-parallelnymi-potokami.html – Многопоточность в Python - шпаргалка для начинающих
13. sites.google.com/view/ifizmat/devhpc – Параллельное программирование. Часть 2. Сайт преподавателя
14. sites.google.com/site/ifizmat/prog/hpc – Параллельное программирование. Часть 1. Сайт преподавателя

6.2. Перечень необходимых профессиональных баз данных и информационных справочных систем

Электронная библиотечная система «IPR SMART». Режим доступа: <http://www.iprbookshop.ru>

Электронная библиотечная система «Юрайт». Режим доступа: <https://urait.ru>

Электронно-библиотечная система «Лань» (раздел «Сетевая электронная библиотека педагогических вузов»). Режим доступа: <https://e.lanbook.com>

7. Методические указания и учебно-методическое обеспечение для обучающихся по освоению дисциплины

Дисциплина реализуется в соответствии с указаниями «Методические рекомендации по организации образовательного процесса при освоении дисциплины», размещенными в ЭИОС университета (eios.ggpi.org).

Методические рекомендации для работы с инвалидами и лицами с ОВЗ размещены в ЭИОС университета (eios.ggpi.org).

8. Материально-техническая база, программное обеспечение, необходимое для осуществления образовательного процесса по дисциплине

Учебный корпус 1, аудитории(я) 237, 219.

Полный перечень материально-технической базы и программного обеспечения размещены в ЭИОС университета (eios.ggpi.org).

Образовательная среда организации, организация рабочих мест обучающихся, технические и программные средства общего и специального назначения соответствуют Методическим рекомендациям по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса (утв. Министерством образования и науки РФ 8 апреля 2014 г. N АК-44/05вн), а именно:

- наличие компьютерной техники, адаптированной для инвалидов со специальным программным обеспечением, альтернативных устройств ввода информации и других технических средств приема-передачи учебной информации в доступных формах для студентов с нарушениями опорно-двигательного аппарата;

- для студентов с нарушениями функций опорно-двигательного аппарата используются альтернативные устройств ввода информации (при необходимости);

- используются специальные возможности операционной системы Windows, такие как экранная клавиатура, с помощью которой можно вводить текст, настройка действий Windows при вводе с помощью клавиатуры или мыши.

Для студентов с нарушениями функций опорно-двигательного аппарата предусмотрено расположение рабочих мест в первых рядах у окна и в среднем ряду.

9. Рейтинг-план оценки успеваемости студентов

Дисциплина /семестры	Объем аудиторной работы				Виды текущей аттестационной аудиторной и внеаудиторной работы	Максимальное (норматив) количество баллов	Поощрение	Штрафы	Итоговая форма отчета (мин. балл)
	лек	Сем / пр	лаб	КСР					
Технологии и программирования /5	16	- /16	-	4	1. Контроль посещаемости лекций 2. Работа на практических занятиях 4. Контроль самостоятельной работы <u>Формы контрольных мероприятий</u> 1. контрольная работа 2. тест <u>Компенсационные мероприятия</u> 1. Решение задач по темам практических занятий	16 40 (5*8) 4 5 5 1	+ 1 балл за дополнения; + 3 балла за подготовку дополнительного дидактического материала	- 3 балла за невыполнение в установленные сроки	Допуск к зачету – 50% «автомат» при зачете – 70%
ИТОГО						70 (без компенсации)			

Дисциплина / Семестр	Объем аудиторной работы				Перечень контрольных мероприятий	Максимально е кол-во баллов	Поощрение	Штрафы	Итоговая форма отчета (мин. балл)
	лк	Прак.	Лаб	КСР					
Технологии программирования / 6	14	14	—	8	1. Контроль посещаемости лекций 2. Контроль посещаемости практических занятий и КСР 3. Работа на практическом занятии и занятии КСР <u>Контрольные мероприятия</u> 1. Тестирование 2. Контрольная работа <u>Компенсационные мероприятия</u> Создание программного продукта по теме, определяемой преподавателем	14 22 55 5 5 10	не применяются	не применяются	Допуск к экзамену – 50% «автомат» при экзамене – 90%
ВСЕГО						103			

Лист регистрации изменений и дополнений к РПД
(фиксируются изменения и дополнения перед началом учебного года,
при необходимости внесения изменений на следующий год –
оформляется новый лист изменений)

№ п.п.	Содержание изменения	Дата, номер протокола заседания кафедры. Подпись заведующего кафедрой	Дата, номер протокола заседания совета факультета. Подпись декана факультета
1.			
2.			
3.			
4.			
5.			
6.			

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

1. Фонд оценочных средств для текущего контроля успеваемости, промежуточной аттестации и послитогового контроля по дисциплине

1.1. Настоящий Фонд оценочных средств(ФОС) по дисциплине «Технологии программирования» является неотъемлемым приложением к рабочей программе дисциплины «Технологии программирования» (РПД). На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

1.2. Оценивание всех видов контроля(текущего, промежуточного, послитогового) осуществляется по 5-ти балльной шкале.

1.3. Результаты оценивания текущего контроля учитываются в рейтинге.

2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными индикаторами достижения компетенций

Код компетенции	ОПК-8
Формулировка компетенции	Способен разрабатывать алгоритмы и программы, пригодные для практического применения;
Индикатор достижения компетенции	ИОПК 8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения ИОПК 8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули ИОПК 8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы

Код компетенции	ПК-2
Формулировка компетенции	Способен разрабатывать требования и проектировать программное обеспечение
Индикатор достижения компетенции	ИПК 2.1 Знает: типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения ИПК 2.2 Умеет: использовать существующие типовые решения и шаблоны проектирования программного обеспечения ИПК 2.3 Владеет: методами и средствами проектирования программного обеспечения

Код компетенции	ПК-7
Формулировка компетенции	Способен разрабатывать компоненты системных программных продуктов
Индикатор достижения компетенции	ИПК 7.1 Знает: синтаксис, особенности программирования и стандартные библиотеки выбранного языка программирования, структуру объектных и исполняемых файлов в операционной

	<p>системе</p> <p>ИПК 7.2 Умеет: использовать коммерческие операционные системы, создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов</p> <p>ИПК 7.3 Владеет: средствами разработки компонентов системных программных продуктов</p>
--	---

3. Содержание оценочных средств текущего контроля и критерии их оценивания

3.1. Текущий контроль осуществляется преподавателем дисциплины при проведении занятий в следующих формах: тестирование, контрольная работа.

3.2. Формы текущего контроля и критерии их оценивания.

Форма контроля 1 - Типовые тестовые задания

Типовой тест.

Проверяемые компетенции и индикаторы достижения компетенций: ОПК-1., ИОПК-8.1., ИОПК-8.2., ИОПК-8.3, ИПК-2.1, ИПК-2.2, ИПК-2.3, ИПК-7.1, ИПК-7.2, ИПК-7.3.

Время выполнения заданий: 45 минут

Критерии оценивания:

- верные ответы на 90%-100% вопросов – «отлично»;
- верные ответы на 70% - 89% вопросов – «хорошо»;
- верные ответы на 50% - 69% вопросов – «удовлетворительно»;
- меньше 50% ответов на вопросы – «неудовлетворительно».

1. Имеются следующие факты: мама('Маша','Саша') - читается как «Маша является мамой Саши»; мама('Саша','Даша'), мама('Саша','Лида'). Как выглядит запрос для вывода дочерей Саши?
 - а) мама('Саша',X).
 - б) мама (X, 'Саша').
 - в) мама('Саша',_).
 - г) мама('Саша', 'Даша').
2. Необходимо записать следующее правило на языке Пролог:

"X - собака при условии, что родителем X является Y, Y - собака?"

 - а) собака (X) :- родитель(Y, X); собака(Y).
 - б) собака (X) :- родитель(Y, X), собака(Y);
 - в) собака (X) :- родитель(Y, X), собака(Y).
 - г) собака (X) :- родитель(Y, X) :- собака(Y).
3. Определите назначение программы *prog*:


```
prog(1,1):-!.
prog(N,F):-M is N-1, prog(M,F1), F is F1 * N.
```

 - а) вычисляет степень числа;
 - б) вычисляет произведение цифр числа;
 - в) определяет n-ый член последовательности Фибоначчи;
 - г) вычисляет факториал числа.
4. Правило - это предложение, которое состоит ...
 - а) только из заголовка;
 - б) из заголовка и тела;
 - в) только из тела;
 - г) из нескольких заголовков.
5. Свободная переменная - это переменная, которая ...
 - а) имеет какое-то значение;

- б) не имеет никакого значения;
 - в) имеет "начальное" значение, соответствующее домену (ноль для числового домена; пустая строка – для строкового домена и т.д.);
 - г) вычисляется в процессе работы программы.
6. Какой из алгоритмов, в общем случае, требует больше оперативной памяти?
- а) циклический;
 - б) рекурсивный;
 - в) условный;
 - г) линейный.
7. В результате выполнения предиката `fact(4,X)` переменная X будет означена числом
- а) 8;
 - б) 32;
 - в) 16;
 - г) 24.
8. Операция отделения «головы» обозначается:
- а) `(HeadTail)`;
 - б) `[Head / Tail]`;
 - в) `(Head | Tail)`;
 - г) `[Head | Tail]`.
9. Установите соответствие:
- | | |
|-----------------|--------------------------|
| 1 , | а) отрицание переменных; |
| 2 ; | б) логическое «и»; |
| 3 $\backslash=$ | в) отрицание фактов; |
| 4 $\backslash+$ | г) логическое «или». |
10. Установите соответствие:
- | | |
|-------------|---|
| 1 Правило | а) :- |
| 2 Факт | б) <code>study(petya,english).</code> |
| 3 Если | в) ! |
| 4 Отсечение | г) <code>prog(V,T, S) :- S is V * T.</code> |

Форма контроля 1 - Типовые тестовые задания

Типовой тест.

Проверяемые компетенции и индикаторы достижения компетенций: ОПК-1., ИОПК-8.1., ИОПК-8.2., ИОПК-8.3, ИПК-2.1, ИПК-2.2, ИПК-2.3, ИПК-7.1, ИПК-7.2, ИПК-7.3.

Время выполнения заданий: 45 минут

Критерии оценивания:

- верные ответы на 90%-100% вопросов – «отлично»;
- верные ответы на 70% - 89% вопросов – «хорошо»;
- верные ответы на 50% - 69% вопросов – «удовлетворительно»;
- меньше 50% ответов на вопросы – «неудовлетворительно».

1. Какое из приведённых значений строковая величина?

- а) текст;
- б) '36т';
- в) "36";
- г) 36.

2. Что выполняет пример?

```
length :: [a] -> Integer
length [] = 0
length (x:xs) = 1 + length xs
```

- а) подсчет количества элементов в списке;

- б) сложение элемента к списку;
 - в) приписывает к списку 1;
 - г) удаление последнего элемента списка.
3. Haskell язык:
- а) логического программирования;
 - б) структурного программирования;
 - в) функционального программирования;
 - г) процедурного программирования.
4. Выберите идентификатор конструктора:
- а) name;
 - б) 1NAME;
 - в) Name;
 - г) na1.
5. Какой литерал относится к числовым:
- а) charesc;
 - б) hexadecimal;
 - в) cntrl;
 - г) char.
6. Выберите строковые литералы:
- а) string;
 - б) ascii;
 - в) cntrl;
 - г) char.
7. Поскольку язык функциональный, то основная управляющая структура – это:
- а) матрица;
 - б) таблица;
 - в) функция;
 - г) список.
8. Одна из основных характеристик языка Haskell:
- а) ленивые вычисления;
 - б) резкие вычисления;
 - в) мгновенные вычисления;
 - г) не сопоставление с образцом.

9. Установите соответствие:

- | | | |
|---|-----------------------------|---|
| 1 | <code>nullxs</code> | а) возвращает первый элемент в списке <code>xs</code> (обернутый конструктором <code>Just</code>), который удовлетворяет предикату <code>p</code> или <code>Nothing</code> в противном случае. Для работы необходим подключенный модуль <code>Data.List</code> |
| 2 | <code>list1 ++ list2</code> | б) удаляет первое вхождение заданного элемента <code>x</code> из списка <code>xs</code> Для работы необходим подключенный модуль <code>Data.List</code> |
| 3 | <code>delete x xs</code> | в) конкатенация двух списков <code>list1</code> и <code>list2</code> ; |
| 4 | <code>find p xs</code> | г) проверяет, пуст ли список <code>xs</code> (возвращает <code>True</code> или <code>False</code>) |

10. Установите соответствие:

- | | | |
|---|---|------------------------------------|
| 1 | <code>delete 'a' "banana"</code> | а) функция вычисления длины списка |
| 2 | <code>Prelude>[1,2]++[3,4]
[1,2,3,4]::Integer</code> | б) удаляет первое вхождение «a» из |

- | | |
|---|--|
| | указанного в его аргументе списка. Для работы необходим подключенный модуль <code>Data.List</code> |
| 3 <code>len [] = 0</code>
<code>len s = 1 + len (tail s)</code> | в) функция возведения в квадрат |
| 4 <code>square :: Integer -> Integer</code>
<code>square x = x * x</code> | г) соединение (конкатенация) списков |

Форма контроля 1 - Типовые тестовые задания

Типовой тест.

Проверяемые компетенции и индикаторы достижения компетенций: ОПК-1., ИОПК-8.1., ИОПК-8.2., ИОПК-8.3, ИПК-2.1, ИПК-2.2, ИПК-2.3, ИПК-7.1, ИПК-7.2, ИПК-7.3.

Время выполнения заданий: 25 минут

Критерии оценивания:

- верные ответы на 90% вопросов – «отлично»;
 - верные ответы на 70% вопросов – «хорошо»;
 - верные ответы на 50% вопросов – «удовлетворительно»;
 - меньше 50% ответов на вопросы – «неудовлетворительно».
1. Для создания нового проекта в Visual Studio нужно выполнить команду:
 - а) Проект – Создать проект – Приложение;
 - б) Проект – Файл – Создать;
 - в) File – New – C++ Project;
 - г) Создать – Приложение – Проект.
 2. Расположите процесс создания приложения в верном порядке:
 - а) написание программного кода, описание свойств элементов, доступных только во время работы приложения, описание реакций на событие появления окна, нажатие на кнопку и других;
 - б) формирование окна программы - расположение необходимых элементов, задание размеров, изменение свойств;
 - в) сохранение приложения;
 - г) отладка программы.
 3. Для подключения к приложению стандартных пространств имен используют ключевое слово
 - а) `void`;
 - б) `class`;
 - в) `private`;
 - г) `import`.
 4. Текстовая часть программы пишется в окне
 - а) Панель элементов.
 - б) Обозреватель объектов.
 - в) Инспектор объектов.
 - г) Редактора кода.
 - д) Список ошибок.
 - е) Вывод.

5. Компонент TextField служит для ...

- а) ввода исходных данных;
- б) чтения данных из полей ввода;
- в) записи в файл;
- г) работы с числовой информацией.

6. Установите соответствие между операторами языка C++ и назначением

- | | |
|------|--------------------|
| 1 ++ | а) Исключающее ИЛИ |
| 2 -- | б) Логическое И |
| 3 ^ | в) Декремент |
| 4 && | г) Инкремент |

7. Установите соответствие между значением и типом данных в языке C++

- | | |
|---------|------------------|
| 1 -123 | а) unsigned char |
| 2 1 | б) int |
| 3 true | в) double |
| 4 4.669 | г) bool |

8. Установите соответствие между технологиями параллельного программирования и определениями

- | | |
|----------|--|
| 1 CUDA | а) стандарт параллельных программ для систем с общей памятью на языках C, C++, Fortran |
| 2 OpenCL | б) программный интерфейс обмена сообщениями между процессами, выполняющими одну задачу |
| 3 OpenMP | в) фреймворк для параллельного программирования на графических и центральных процессорах |
| 4 MPI | г) программно-аппаратная архитектура параллельных вычислений на графических процессорах nVidia |

9. Что означают технологии проектирования вычислительных систем по классификации Флинна?

- | | |
|--------|---|
| 1 SIMD | а) много потоков команд, много потоков данных |
| 2 SISD | б) много потоков команд, один поток данных |
| 3 MISD | в) один поток команд, один поток данных |
| 4 MIMD | г) один поток команд, много потоков данных |

Форма контроля 2 –Типовая контрольная работа

Типовая контрольная работа.

Проверяемые компетенции и индикаторы достижения компетенций: ОПК-1., ИОПК-8.1., ИОПК-8.2., ИОПК-8.3, ИПК-2.1, ИПК-2.2, ИПК-2.3, ИПК-7.1, ИПК-7.2, ИПК-7.3.

Время выполнения заданий: 45 минут

Критерии оценивания:

Обучающимся предлагается решить серию из трех задач на компьютере.

Для получения оценки «удовлетворительно» предлагается решить одну задачу.

Для получения оценки «хорошо» предлагается решить две задачи.

Для получения оценки «отлично» предлагается решить все задачи.

1. База данных “Распорядок дня”: занятие(0, 7, сон), занятие(7, 8, завтрак), занятие(8, 13, школа), занятие(13, 14,обед), занятие(14, 19, свобода), занятие(19, 20, ужин), занятие(20, 23, отдых), занятие(23, 24, сон). Сформулировать вопросы на Прологе: Когда бывает обед? Что бывает между 14 и 19 часами? Когда бывает сон? (сколько будет решений?)
2. Вывести все числа от 1 до n .
3. Три подруги вышли в белом, зеленом и синем платьях и туфлях. Известно, что только у Ани цвета платья и туфель совпадали. Ни туфли, ни платье Вали не были белыми. Наташа была в зеленых туфлях. Определить цвета платья и туфель на каждой из подруг.

Форма контроля 2 –Типовая контрольная работа

Типовая контрольная работа.

Проверяемые компетенции и индикаторы достижения компетенций: ОПК-1., ИОПК-8.1., ИОПК-8.2., ИОПК-8.3, ИПК-2.1, ИПК-2.2, ИПК-2.3, ИПК-7.1, ИПК-7.2, ИПК-7.3.

Время выполнения заданий: 45 минут

Критерии оценивания:

Обучающимся предлагается решить серию из трех задач на компьютере.

Для получения оценки «удовлетворительно» предлагается решить одну задачу.

Для получения оценки «хорошо» предлагается решить две задачи.

Для получения оценки «отлично» предлагается решить все задачи.

1. Написать программу определения N -ого числа Фибоначчи.
2. Построить функцию вычленения N -ого элемента из заданного списка.
3. Построить функцию сложения элементов двух списков. Возвращает список, составленный из сумм элементов списков-параметров. Учесть, что переданные списки могут быть разной длины.

Форма контроля 2 – Типовая контрольная работа

Типовая контрольная работа.

Проверяемые компетенции и индикаторы достижения компетенций: ОПК-1., ИОПК-8.1., ИОПК-8.2., ИОПК-8.3, ИПК-2.1, ИПК-2.2, ИПК-2.3, ИПК-7.1, ИПК-7.2, ИПК-7.3.

Время выполнения заданий: 45 минут

Критерии оценивания:

Обучающимся предлагается выполнить серию из трех заданий.

Для получения оценки «удовлетворительно» необходимо выполнить 1 задание.

Для получения оценки «хорошо» необходимо выполнить 2 задания.

Для получения оценки «отлично» необходимо выполнить все задания.

1. Напишите параллельную программу, реализующую скалярное произведение двух векторов.
2. Создать 2 вектора с 1000 элементами из случайных чисел. Сложить эти вектора.
3. Сгенерировать три вектора a , b и c размером 100 из случайных чисел. Вычислить сумму векторов a и b , затем найденную сумму скалярно умножить на вектор c .

3.3 Методические указания по проведению процедуры текущего контроля

1. Текущий контроль проводится на протяжении всего семестра.
2. Сбор, обработка и оценивание результатов текущего контроля проводятся преподавателем, ведущим дисциплину.

3. Предъявление результатов оценивания осуществляется в течение недели после проведения контрольного мероприятия.
4. Результаты текущего контроля учитываются в рейтинге по дисциплине.
5. Все материалы, полученные от обучающихся в ходе текущего контроля (контрольная работа, диктант, тест, организация дискуссии, круглого стола, доклад, реферат, отчет по лабораторной работе, отчет по педагогической практике и т.п.), должны храниться в течение текущего семестра на кафедрах.
6. Считать, что положительные результаты текущего контроля свидетельствуют об успешном процессе формирования указанных компетенций и индикаторов достижения компетенций (этапов формирования компетенций).

4. Содержание оценочных средств промежуточной аттестации и критерии их оценивания

- 4.1. Промежуточная аттестация проводится в виде: зачета (5 сем.) и экзамена (6 сем.).
- 4.2. Содержание оценочного средства. Проверяемые компетенции и индикаторы достижения компетенций: ОПК-8, ИОПК 8.1, ИОПК 8.2, ИОПК 8.3, ПК-2, ИПК 2.1, ИПК 2.2, ИПК 2.3, ПК-7, ИПК 7.1, ИПК 7.2, ИПК 7.3

Примерные вопросы и задания к зачету

1. Представление о логическом программировании: область применения, основные принципы, языки.
2. Представление данных при помощи фактов. Приведите пример.
3. Представление данных при помощи правил. Приведите пример.
4. Запросы программы, их назначение. Приведите пример.
5. Понятие рекурсии. Приведите пример.
6. Рекурсия как метод организации повторяющихся действий в Прологе.
7. Создание рекурсивных правил. Приведите пример.
8. Список как рекурсивный объект данных в Прологе. Приведите пример.
9. Объявление списков. Приведите пример.
10. Объединение списков. Приведите пример.
11. Поиск заданного элемента в списке. Приведите пример.
12. Удаление элемента из списка и вставка элемента в список. Приведите пример.

Теория

1. История функционального программирования. Решаемые задачи. Языки функционального программирования.
2. Свойства функциональных языков.
3. Структуры данных и базисные операции. Атомы и списки. Программная реализация списков и списочных структур. Графическое представление списочной структуры. Примеры.
4. Структуры данных и базисные операции. Типы в функциональных языках. Образцы и клозы. Охрана. Локальные переменные.
5. Структуры данных и базисные операции. Аккумулятор. Принципы построения определений с накапливающим параметром.
6. Основы языка Haskell. Структуры данных и их типы. Соглашения по именованию. Определители списков и математические последовательности. Вызовы функций. Использование λ -исчисления. Инфиксный способ записи функций.
7. Служебные слова и синтаксис Haskell'a. Охрана и локальные переменные. Полиморфизм. Сравнение с другими языками.
8. Модули и монады в Haskell'e. Абстрактные типы данных. Встроенные монады.

9. Операции ввода/вывода в Haskell'е. Базовые операции ввода/вывода. Программирование при помощи действий. Обработка исключений. Файлы, каналы и обработчики.
10. Конструирование функций. Декартово произведение. Размеченное объединение. Примеры определения новых типов:
 - формальное определение типа List (A);
 - формальное определение типа List_str (A);
 - формальное определение деревьев и лесов с помеченными вершинами;
 - формально определение деревьев с помеченными вершинами и дугами;
 - функция insert для вставки элемента в дерево;
 - функция access для поиска элементов в B-дереве.
11. Доказательство свойств функций. Виды областей определения. Примеры.
12. Трансформация программ. Виды преобразований. Второй закон информатики.
13. Частичные вычисления. Проекция Футаморы

Практика

4. Написать программу, реализующую быструю сортировку Хоара.
5. Написать программу определения N-ого числа Фибоначчи.
6. Определить функцию возведения целого числа в квадрат.
7. Построить функцию вычленения N-ого элемента из заданного списка.
8. Построить функцию сложения элементов двух списков. Возвращает список, составленный из сумм элементов списков-параметров. Учесть, что переданные списки могут быть разной длины.
9. Построить функцию перестановки местами соседних чётных и нечётных элементов в заданном списке.
10. Построить функцию, обращающую список (первый элемент списка становится последним, второй — предпоследним, и так далее до последнего элемента).
11. Построить функцию, возвращающую номер первого вхождения заданного атома в список.

Примерные вопросы и задания к экзамену

Теоретические вопросы

1. Вычислительные процессы. Формальная модель. Свойства процесса.
2. Система процессов. Независимые процессы. Взаимодействие процессов.
3. Классы параллельных ВС.
4. Отображение системы процессов. Синхронизация.
5. Показатели параллельных ВС и вычислений.
6. Модель вычислений в виде графа «операции-операнды».
7. Схема параллельного выполнения алгоритма.
8. Определение времени выполнения параллельного алгоритма.
9. Теоретические оценки времени выполнения параллельных алгоритмов.
10. Показатели эффективности параллельного алгоритма.
11. Оценка максимально достижимого параллелизма. Анализ масштабируемости параллельных вычислений.
12. Процессы и потоки.
13. Закон Амдала.
14. MPI: Стандарт MPI. Общие понятия. Области связи и коммутаторы.
15. Общие процедуры MPI: Инициализация, Завершение.
16. Общие процедуры MPI: Определение общего числа параллельных процессов в группе. Определение номера процесса в группе.
17. MPI: Прием/передача сообщений между отдельными процессами.
18. MPI: Групповые (коллективные) взаимодействия. Рассылка целого сообщения

процессам. Сборка данных от процессов.

19. MPI: Групповые (коллективные) взаимодействия: Рассылка частей сообщения процессам. Сборка частей сообщения с процессов.
20. MPI: Функции поддержки распределенных операций: Выполнение глобальных
21. операций с возвратом результатов в главный процесс. Выполнение глобальных операций с возвратом результатов во все процессы.
22. MPI: Синхронизация процессов.
23. OpenMP: Общие понятия. Область применения.
24. Директивы OpenMP: Директивы для определения параллельной области.
25. OpenMP: Директивы для распределения вычислений внутри параллельной области: Директива `for`.
26. OpenMP: Директивы для распределения вычислений внутри параллельной области: Директива `sections`. Директива `single`.
27. OpenMP: Директивы для распределения вычислений внутри параллельной области: Объединение директив `parallel` и `for (sections)`.
28. Классическая задача об обедающих философах.
29. Задача «читатели и писатели».
30. Язык параллельного программирования Okkam: основные конструкции и приемы программирования.
31. Язык параллельного программирования Ada: основные конструкции и приемы программирования.
32. Язык параллельного программирования Linda: основные конструкции и приемы программирования.

Практика (одна из задач)

1. Напишите параллельную программу, реализующую скалярное произведение двух векторов.
2. Напишите параллельную программу, реализующую поиск максимального значения вектора.
3. Напишите параллельную программу, реализующую транспонирование матрицы $n \times n$.
4. Реализуйте параллельный алгоритм метода Гаусса решения систем линейных алгебраических уравнений.
5. На основе программы для решения СЛАУ методом Гаусса напишите параллельную программу для вычисления определителя и постройте зависимость коэффициента ускорения от числа ядер (поток) на узле.
6. На основе программы для решения СЛАУ методом Гаусса напишите параллельную программу для вычисления обратной матрицы и постройте зависимость коэффициента ускорения от числа ядер (поток) на узле.
7. Напишите параллельную программу реализовывающую метод Гаусса (метод элементарных преобразований) для вычисления ранга произвольной матрицы $A - m \times n$.
8. Найти численное решение задачи Дирихле для уравнения Пуассона итерационным методом Зейделя.

MPI

9. Создать 2 вектора с 1000 элементами из случайных чисел. Сложить эти вектора.
10. Создать 2 вектора с 1000 элементами из случайных чисел. Скалярно умножить эти вектора. Сравнить время счета параллельного и последовательного алгоритмов.
11. Дан двумерный массив размерности $n \times m$. Найти максимальное и минимальное значение массива.
12. Дан двумерный массив размерности (n, m) , где $n > 10$, $m > 5$. Найти сумму диагональных элементов и сумму всех элементов матрицы.
13. Выполнить транспонирование матрицы $n \times n$.

14. Дан двумерный массив размерности (n, m) , где $n > 10, m > 5$. Отсортировать данный массив.
15. Вычислить одномерные определенные интегралы от трёх разных подынтегральных функций (Например: $\sin x, 1 + 2e^x, x^2 + 4x + 3$).
16. Сгенерировать три вектора a, b и c размером 100 из случайных чисел. Вычислить сумму векторов a и b , затем найденную сумму скалярно умножить на вектор c .
17. Формула смешанного произведения a, b и c : сначала векторное произведение a и b , затем результат скалярно умножаем на c .
18. Даны трехмерные вектора a, b и c . Найти скалярное произведение a и b , векторное произведение a и c , смешанное произведение a, b и c . Формула скалярного умножения a и b : $a_1 * b_1 + a_2 * b_2 + a_3 * b_3$. Формула векторного произведения a и c , вектор c компонентами: $(a_2 * c_3 - a_3 * c_2, a_3 * c_1 - a_1 * c_3, a_1 * c_2 - a_2 * c_1)$.

4.3. Критерии оценивания

Зачет выставляется по результатам рейтинга. Если обучающийся набрал недостаточное количество баллов, то он сдает зачет.

Шкала оценивания для зачета:

Уровни освоения индикаторов достижения компетенций	Основные признаки выделения уровня	Академическая оценка	% освоения (рейтинговая оценка)
Сформирован	Студент показал достаточно прочные знания основных положений учебной дисциплины, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты.	Зачтено	50-100
Не сформирован	При ответе выявились существенные пробелы в знаниях основных положений учебной дисциплины, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины.	Не зачтено	менее 50

Оценка за экзамен выставляется с учетом рейтинга. Если обучающийся набрал недостаточное количество баллов или хочет повысить оценку, то обучающийся сдает экзамен.

Шкала оценивания для экзамена:

Уровни освоения индикаторов в достижения компетенций	Содержательное описание уровня	Основные признаки выделения уровня	Академическая оценка	% освоения (рейтинговая оценка)
Повышенный (высокий)	Творческая деятельность	Включает нижестоящий уровень. Умение самостоятельно	Отлично	90-100

		принимать решение, решать проблему/задачу теоретического или прикладного характера на основе изученных методов, приемов, технологий.		
Базовый	Продуктивная деятельность	Включает нижестоящий уровень. Способность собирать, систематизировать, анализировать и грамотно использовать информацию из самостоятельно найденных теоретических источников и иллюстрировать ими теоретические положения или обосновывать практику применения	Хорошо	70-89
Удовлетворительный	Репродуктивная деятельность	Изложение в пределах задач курса теоретического и практического материала	Удовлетворительно	50-69
Недостаточный	Отсутствие признаков удовлетворительного уровня		Неудовлетворительно	менее 50

4.4. Методические указания по проведению процедуры промежуточной аттестации

1. Сроки проведения процедуры оценивания: по расписанию экзаменов (зачета - на последнем занятии по предмету). Если обучающийся по результатам рейтинговой системы не набирает нужное количество баллов или желает повысить оценку, то сдает экзамен/ зачет согласно требованиям.
2. Сбор, обработка и оценивание результатов промежуточной аттестации проводится преподавателем, ведущим дисциплину.
3. Предъявление результатов оценивания осуществляется: по окончании ответа студента и фиксируется в зачетной книжке и экзаменационной ведомости.
4. При наличии письменных ответов обучающихся, полученных в ходе экзаменационной сессии, материалы хранятся в течение месяца после завершения сессии на кафедрах.
5. Порядок выполнения и защиты курсовой работы регламентирован «Положением о курсовой работе ФГБОУ ВО «Глазовский государственный инженерно-педагогический университет имени В.Г. Короленко».
6. Считать, что положительные результаты промежуточного контроля свидетельствуют об успешном процессе формирования указанных компетенций и индикаторов достижения компетенций (этапов формирования компетенций).

5. Содержание оценочных средств для проверки сформированности компетенций и индикаторов достижения компетенций (поститоговый контроль) и критерии их оценивания

Задания для проверки компетенции и индикаторов достижения компетенции: ОПК-8, ИОПК 8.1, ИОПК 8.2, ИОПК 8.3, ПК-2, ИПК 2.1, ИПК 2.2, ИПК 2.3, ПК-7, ИПК 7.1, ИПК 7.2, ИПК 7.3

Код компетенции	ОПК-8
Формулировка компетенции	Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

Индикатор достижения компетенции	ИОПК 8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения ИОПК 8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули ИОПК 8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы
--	---

Время выполнения заданий: не более 30 минут

- Имеются следующие факты: мама('Маша','Саша') - читается как «Маша является мамой Саша»; мама('Саша','Даша'), мама('Саша','Лида'). Как выглядит запрос для вывода дочерей Саша?
 - мама('Саша',X).
 - мама (X, 'Саша').
 - мама('Саша',_).
 - мама('Саша', 'Даша').
- Необходимо записать следующее правило на языке Пролог:
"X - собака при условии, что родителем X является Y, Y - собака?"
 - собака (X) :- родитель(Y, X); собака(Y).
 - собака (X) :- родитель(Y, Y), собака(Y);
 - собака (X) :- родитель(Y, X), собака(Y).
 - собака (X) :- родитель(Y, X) :- собака(Y).
- Определите назначение программы *prog*:

```
prog(1,1) :- !.
```

```
prog(N,F) :- M is N-1, prog(M,F1), F is F1 * N.
```

 - вычисляет степень числа;
 - вычисляет произведение цифр числа;
 - определяет n-ый член последовательности Фибоначчи;
 - вычисляет факториал числа.
- Правило - это предложение, которое состоит ...
 - только из заголовка;
 - из заголовка и тела;
 - только из тела;
 - из нескольких заголовков.
- Операция отделения «головы» обозначается:
 - (HeadTail);
 - [Head / Tail];
 - (Head | Tail);
 - [Head | Tail].

Ключ к тесту:

Номер вопроса	1	2	3	4	5
Номер правильного ответа	а	в	г	б	г

Время выполнения заданий: не более 30 минут

- Установите соответствие:

1 ,	а) отрицание переменных;
2 ;	б) логическое «и»;
3 \=	в) отрицание фактов;
4 \+	г) логическое «или».

2. Установите соответствие:

- | | | | |
|---|-----------|----|----------------------------|
| 1 | Правило | а) | :- |
| 2 | Факт | б) | study(petya, english). |
| 3 | Если | в) | ! |
| 4 | Отсечение | г) | prog(V,T, S) :-S is V * T. |

Ключ к тесту:

Номер вопроса	1	2
Номер правильного ответа	1 - б 2 - г 3 - а 4 - в	1 - г 2 - б 3 - а 4 - в

Время выполнения заданий: не более 30 минут

1. Практическое задание. Решить задачу на языке Prolog.

Три подружки вышли в белом, зеленом и синем платьях и туфлях. Известно, что только у Ани цвета платья и туфель совпадали. Ни туфли, ни платье Вали не были белыми. Наташа была в зеленых туфлях. Определить цвета платья и туфель на каждой из подруг.

Ключ к решению практической задачи:

Ответ к задаче: у Ани платье и туфли – белые. Платье Вали - зеленое, туфли Вали – синие. Платье Наташи - синее, туфли Наташи – зеленые.

Код компетенции	ПК-2
Формулировка компетенции	Способен разрабатывать требования и проектировать программное обеспечение
Индикатор достижения компетенции	ИПК 2.1 Знает: типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения ИПК 2.2 Умеет: использовать существующие типовые решения и шаблоны проектирования программного обеспечения ИПК 2.3 Владеет: методами и средствами проектирования программного обеспечения

Время выполнения заданий: не более 30 минут

Практическое задание. Решить задачу на языке Prolog.

База знаний “Воинская служба”: возраст(борис, 18), возраст(андрей, 17), возраст(михаил, 18), возраст(анна, 18), возраст(юлия, 17), мужчина(андрей), мужчина(борис), мужчина(михаил), женщина(анна), женщина(юлия).

Определить правило подлежит призыву (priz), не_подлежит_призыву (no_priz).

Сформулировать вопросы:

- Кто подлежит призыву?
- Подлежит ли призыву Анна?

Ключ к решению практической задачи:

Ответ к задаче:

- priz(X).
- no_priz(X).

Код компетенции	ПК-7
Формулировка	Способен разрабатывать компоненты системных программных

компетенции	продуктов
Индикатор достижения компетенции	ИПК 7.1 Знает: синтаксис, особенности программирования и стандартные библиотеки выбранного языка программирования, структуру объектных и исполняемых файлов в операционной системе ИПК 7.2 Умеет: использовать коммерческие операционные системы, создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов ИПК 7.3 Владеет: средствами разработки компонентов системных программных продуктов

Время выполнения заданий: не более 30 минут

Практическое задание. Исправьте ошибки в программном коде, допущенные при решении задачи: с помощью библиотеки OpenMP присвоить элементам массива псевдослучайные числа из отрезка [A, B].

```
#include <iostream>
#define N 20
#define A 23
#define B 55
using namespace std;
int main() {
    float i;
    int v[N] = {};
    cout << "Test array random" << endl;
#pragma omp for
    for (i = 0; i < v; i++) {
        int r = A + rand()%(B - A + 1);
        v[i] == r;
        cout << "v[" << i << "] = " << v[i] << beginl;
    }
    cout << endl;
    cin.get()
    return 0;
}
```

Ключ к практическому заданию:

```
#include <omp.h>
#include <iostream>
#define N 20
#define A 23
#define B 55
using namespace std;
int main() {
    int v[N] = {};
    cout << "Test array random" << endl;
#pragma omp for

    for (int i = 0; i < N; i++) {
```

```

        int r = A + rand()%(B - A + 1);
v[i] = r;
cout << "v[" << i << "] = " << v[i] << endl;
    }
    cout << endl;
    cin.get();
    return 0;
}

```

Критерии оценивания:

Каждый индикатор достижения компетенции оценивается в 10 баллов:

- Тестовое задание оценивается в 10 баллов (ответ на вопрос теста стоит 0 или 2 балла);
- Задания на соответствие оцениваются в 10 баллов (каждое оценивается 0-5 баллов)
 - 5 баллов – полностью правильно найденные соответствия;
 - 4 балла – три правильных соответствия;
 - 3 балла – два правильных соответствия;
 - 2 балла – одно правильно соответствие;
 - 1 балл – отсутствие правильных соответствий;
 - 0 баллов – не приступал к выполнению задания;
- Каждое практическое задание оценивается в 10 баллов:
 - 10 баллов - студент правильно выполнил предложенные задания на основе изученной теории, методов, приемов, технологий;
 - 8 баллов - студент способен применять полученные теоретические знания в практической деятельности, решать типичные задачи на основе воспроизведения стандартных алгоритмов, при выполнении заданий допускает незначительные ошибки;
 - 6 баллов - при выполнении задания допущены грубые ошибки;
 - 0 баллов - студент не выполнил задание.

Оценка зависит от процента выполнения всех заданий.

Шкала оценивания сформированности компетенции(ий) и индикатора(ов) достижения компетенции(ий)

Уровни освоения индикатора (ов) достижений компетенций	Основные признаки выделения уровня	Академическая оценка	% выполнения всех заданий
Повышенный (высокий)	Включает нижестоящий уровень. Умение самостоятельно принимать решение, решать проблему/задачу теоретического или прикладного характера на основе изученных методов, приемов, технологий.	Отлично	90-100
Базовый	Включает нижестоящий уровень. Способность собирать, систематизировать, анализировать и грамотно использовать информацию из самостоятельно найденных теоретических источников и иллюстрировать ими теоретические положения или обосновывать практику	Хорошо	70-89

	применения		
Удовлетворительны й	Изложение в пределах задач курса теоретического и практического контролируемого материала	Удовлетво рительно	50-69
Недостаточный	Отсутствие признаков удовлетворительного уровня	Неудовлетв орительно	менее 50

Считать, что положительные результаты поститогового контроля свидетельствуют об успешном процессе формирования компетенции(ий) и индикатора(ов) достижения компетенции(ий) (этапа формирования компетенции). Если обучающийся получил оценку «неудовлетворительно», то считать компетенцию не сформированной на данном этапе. При получении оценок «удовлетворительно», «хорошо» или «отлично» считать, что проверяемая компетенция сформирована на достаточном уровне.

Методические указания для проверки остаточных знаний

1. Сроки проведения процедуры оценивания: по графику деканата.
2. Сбор, обработка и оценивание результатов поститогового контроля проводится преподавателем по распоряжению деканата.
3. Предъявление результатов оценивания осуществляется в течение недели после проведения контрольного мероприятия, оформляется в виде отчета и хранится в деканате в течение всего срока обучения обучающегося.