



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

«СОГЛАСОВАНО»

Руководитель ОП


Подпись

Дремлюга Р.И.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Языки, алгоритмы и методы программирования

Направление подготовки - 09.04.01 Информатика и вычислительная техника
(Искусственный интеллект и большие данные (совместно с ПАО Сбербанк))

Форма подготовки очная

курс 1 семестр 1,2
лекции 36 час.
практические занятия 72 час.
лабораторные работы 00 час.
в том числе с использованием МАО практические занятия 72 час
всего часов аудиторной нагрузки 108 час.
самостоятельная работа 180 час.
в том числе на подготовку к экзамену 00 час.
контрольные работы (количество) не предусмотрены
курсовая работа / курсовой проект не предусмотрены
зачет с оценкой 1,2 семестр
экзамен не предусмотрен

Рабочая программа дисциплины разработана в соответствии с федеральным государственным образовательным стандартом (ФГОС) высшего образования (ВО) – магистратура по направлению подготовки 09.04.01 Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации от 19.09.2017 г. № 918 (с изменениями и дополнениями).

Рабочая программа обсуждена на заседании Академии цифровой трансформации протокол № 1 от 25 января 2022 г.

И.о. директора Академии цифровой трансформации

Еременко А.С.

Составитель (ли): к.т.н. Еременко А.С., Загумённых А.А.

Владивосток
2022

Оборотная сторона титульного листа РПД

- I. **Рабочая программа пересмотрена на заседании департамента:**
Протокол от «_____» _____ 20__ г. № _____
Директор департамента _____
(подпись) (И.О.Фамилия)
- II. **Рабочая программа пересмотрена на заседании департамента:**
Протокол от «_____» _____ 20__ г. № _____
Директор департамента _____
(подпись) (И.О. Фамилия)

I. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель: углубленное изучение классов современных языков и систем программирования, с методами создания приложений с использованием этих языков и систем.

Задачи:

- углубленное изучение современных классов языков программирования, изучение их особенностей и различий;
- углубленное изучение современных систем программирования;
- получение навыков создания программных средств с использованием современных языков и систем программирования.

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют формирование следующих общепрофессиональных компетенций:

Общепрофессиональные компетенции выпускников и индикаторы их достижения:

Код и наименование компетенции выпускника	Код и наименование индикатора достижения компетенции
ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.1. Применяет знания современных интеллектуальных технологий, инструментальных сред, программно-технических платформ для решения профессиональных задач
	ОПК-2.2. Разрабатывает оригинальные алгоритмы и программные средства для решения профессиональных задач
ОПК-5 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	ОПК-5.1 Применяет знания современного программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач
	ОПК-5.2. Осуществляет разработку программного и аппаратного обеспечения информационных и автоматизированных систем
	ОПК-5.3. Выполняет модернизацию программного и аппаратного обеспечения информационных и автоматизированных систем
ОПК-6 Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования	ОПК-6.2. Анализирует техническое задание, разрабатывает и оптимизирует программный код для решения задач обработки информации и автоматизированного проектирования
	ОПК-6.3. Составляет техническую документацию по использованию и настройке компонентов программно-аппаратного комплекса
ОПК-8 Способен осуществлять эффективное управление разработкой программных средств и проектов	ОПК-8.1. Выбирает методы и средства разработки программного обеспечения, оценивает сложность проектов, планирует ресурсы, контролирует сроки выполнения и оценивает качество полученного результата

Код и наименование компетенции выпускника	Код и наименование индикатора достижения компетенции
	ОПК-8.2. Выполняет разработку технического задания, составляет планы, распределяет задачи, тестирует и оценивает качество программных средств

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения)
ОПК-2.1. Применяет знания современных интеллектуальных технологий, инструментальных сред, программно-технических платформ для решения профессиональных задач	Знает пакеты прикладных программ, относящиеся к профессиональной сфере
	Умеет разрабатывать оригинальные программные средства, в том числе с использованием современных информационно-коммуникационных и интеллектуальных компьютерных технологий
	Владеет методами проектирования и программирования интеллектуальных технологий
ОПК-2.2. Разрабатывает оригинальные алгоритмы и программные средства для решения профессиональных задач	Знает пакеты прикладных программ, относящиеся к профессиональной сфере
	Умеет разрабатывать оригинальные программные средства, в том числе с использованием современных информационно-коммуникационных и интеллектуальных компьютерных технологий
	Владеет навыками создания математических моделей, алгоритмов, методов, инструментальных средств по тематике проводимых проектов
ОПК-5.1 Применяет знания современного программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач	Знает современные классы программного и аппаратного обеспечения
	Умеет выбирать требуемые классы при выполнении исследований
	Владеет методами обоснования выбора
ОПК-5.2. Осуществляет разработку программного и аппаратного обеспечения информационных и автоматизированных систем	Знает методы разработки программного обеспечения для решения профессиональных задач
	Умеет проектировать программное обеспечение для решения профессиональных задач
	Владеет методами создания кода программного обеспечения в соответствии с проектом
ОПК-5.3. Выполняет модернизацию программного и аппаратного обеспечения информационных и автоматизированных систем	Знает способы модернизации программного обеспечения
	Умеет модернизировать программное обеспечение в соответствии с требованиями заказчика
	Владеет методами модернизации программного обеспечения
ОПК-6.2. Анализирует техническое задание, разрабатывает и оптимизирует программный код для решения задач обработки информации и автоматизированного проектирования	Знает принципы разработки технической документации
	Умеет анализировать техническое задание
	Владеет навыками оптимизации программного кода для решения задач обработки информации и автоматизированного проектирования
ОПК-6.3. Составляет техническую документацию по использованию и настройке компонентов программно-аппаратного комплекса	Знает принципы разработки технической документации
	Умеет разрабатывать нормативно-технические документы (стандарты и регламенты) по разработке программных средств и проектов

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения)
	Владеет методами составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса
ОПК-8.1. Выбирает методы и средства разработки программного обеспечения, оценивает сложность проектов, планирует ресурсы, контролирует сроки выполнения и оценивает качество полученного результата	Знает методы и средства разработки программного обеспечения
	Умеет оценивать сложность проектов
	Владеет навыками планирования ресурсов
ОПК-8.2. Выполняет разработку технического задания, составляет планы, распределяет задачи, тестирует и оценивает качество программных средств	Знает принципы разработки технической документации
	Умеет анализировать техническое задание
	Владеет навыками составления планов, распределения задач, тестирования и оценки качества программных средств

II. ТРУДОЁМКОСТЬ ДИСЦИПЛИНЫ И ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ

Общая трудоемкость дисциплины составляет 8 зачётных единиц (288 академических часов).

(1 зачетная единица соответствует 36 академическим часам)

Видами учебных занятий и работы обучающегося по дисциплине могут являться:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекции
Пр	Практические занятия
СР	Самостоятельная работа обучающегося в период теоретического обучения
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с преподавателем в период промежуточной аттестации

Структура дисциплины:

Форма обучения – очная

	Наименование раздела дисциплины	с е м е с т р	Количество часов по видам учебных занятий и работы обучающегося					Контроль	Формы промежуточной аттестации
			Лек	Лаб	Пр	ОК	СР		
1	МОДУЛЬ 1. Принципы программирования на языках высокого уровня	1	6		10				УО-1, УО-4, ПР-6
2	МОДУЛЬ 2. Обзор языков программирования высокого уровня	1	6		26		180		
3	МОДУЛЬ 3. Основы программирования на разных языках	2	6		18				
4	МОДУЛЬ 4. Алгоритмы программирования	2	6		18				
	Итого:		36		72		180		

III. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Трудоемкость теоретической части курса 36 час.

1 и 2 семестр (36 час.)

МОДУЛЬ 1. Принципы программирования на языках высокого уровня

Тема 1.1. Парадигмы программирования. Модели разработки программ: водопадная, итерационная. Методы разработки ПО. Способы исполнения программ.

Структурное программирование, объектно-ориентированное программирование, императивное программирование, декларативное программирование.

Сложность как основополагающая проблема программирования, а абстракция-конкретизация и прогнозирование-контроль – как основные ортогональные методы борьбы со сложностью.

Стили программирования: операционный, ситуационный, функциональный, реляционный, параллельный, объектно-ориентированный.

Тема 1.2. Аспекты программирования на языках высокого уровня

Базовые концепции и принципы с различных позиций (технологической, авторской, математической, семиотической и реализаторской).

Управление памятью, ручное и автоматическое управление памятью. Куча. Счётчик ссылок, решение проблемы подсчёта ссылок. Сборка мусора.

Компиляция, интерпретация, байт-код.

Теория тестирования методом белого ящика, тестирование методом чёрного ящика, unit-тест, стресс-тестирование.

Фундаментальные концепции и принципы, воплощенные в современных и перспективных языках программирования

Способы классификации ЯП. Статическая типизация, динамическая типизация, слабая типизация, сильная типизация.

МОДУЛЬ 2. Обзор языков программирования высокого уровня

Тема 2.1 История возникновения языков программирования высокого уровня

Докомпьютерные языки программирования. Появление компьютерных языков программирования. Машинный код, автокод, язык ассемблера.

Появление языков программирования высокого уровня. Язык FORTRAN, современное развитие языка. Язык COBOL, влияние на синтаксис SQL.

Формализация описания языков программирования. Появление структурного программирования, язык Algol, его основные идеи и версии.

Тема 2.2 Классификация языков программирования высокого уровня

Возникновение языков программирования. Уровень языков программирования. Императивные и процедурные языки. Автокоды, язык Ассемблера, FORTRAN, Lisp, Algol 60, COBOL.

Тема 2.3 Поколения языков программирования высокого уровня Второе поколение языков программирования. Структурное программирование. FORTRAN II, Algol 68, PL/1. Интерпретируемые и компилируемые ЯП. Basic.

Третье поколение ЯП. Слабая, сильная и строгая типизация. Поддержка модульности. Pascal, Modula, Modula-2, Oberon, C.

Перспективы развития в области развития языков программирования.

МОДУЛЬ 3. Основы программирования на разных языках

Тема 3.1. ЯП Python

Модель выполнения. Модель памяти. Стандартные структуры данных ЯП Python: списки, кортежи, словари, множества.

Регулярные выражения. Их использование в ЯП Python. ДКА. Реализация регулярных выражений с использованием НДКА.

Конструкции языка. Элементы ФП в ЯП Питон: функции высших порядков, замыкания. Итераторы. Генераторы.

Список в ЯП Python. Основные виды типов данных в ЯП Python, list comprehension, множественное присваивание, слайсинг, отрицательные индексы списков в ЯП Python, ассоциативный массив.

Тема 3.2 Регулярные выражения

Регулярные выражения: определение, проверка принадлежности языку РВ при помощи НДКА. Виртуальное адресное пространство. Лексикографический порядок сравнения строк, сырая строка. ДКА, НДКА. Квантификаторы, множества символов, обратные ссылки.

Тема 3.3 ЯП Perl.

Назначение языка, его классификация. Основные типы данных ЯП Perl (числа, строки, массивы, ссылки). Понятие контекста. Переменные по умолчанию. Управляющие конструкции (if, for, foreach, while, sub).

Функции ЯП Perl. Передача параметров в функции. Управляющие конструкции. Работа со строками.

МОДУЛЬ 4. Алгоритмы программирования

Тема 4.1 Программа как единство алгоритма и структуры данных.

Данные базового типа (целые, вещественные, символьные, логические). Определение типов и выражения с данными базового типа в языке программирования. Механизмы приведения типов, система конструируемых типов языка. Статическая типизация, динамическая типизация, слабая типизация, сильная типизация.

Тема 4.2 Структура хранения данных.

Стандартные структуры данных ЯП Python: списки, кортежи, словари, множества. Механизмы хранения и организации данных. Хранение данных и алгоритмы их обработки. Управление памятью, ручное и автоматическое управление памятью. Куча. Счётчик ссылок, решение проблемы подсчёта ссылок. Сборка мусора.

Тема 4.3 Абстрактные типы данных.

Абстрактные типы данных как метод ведения проектирования нетривиальных алгоритмов на более высоком уровне. Переход от описания алгоритма с использованием прикладных или математических понятий к описанию в конкретной системе вычислений. Виртуальное адресное пространство.

Тема 4.4 Алгоритмы обработки данных

Проблематика CS, трудные задачи и эффективные алгоритмы, эквивалентность языков, машина Тьюринга. Описание и анализ основных алгоритмов обработки данных: сортировка данных, поиск образа в строке,

алгоритмы обработки графов. Хеширование. Алгоритм двоичного поиска. Алгоритм Кнута, Мориса, Пратта.

Тема 4.5 Динамическое программирование. Жадные алгоритмы

Кубическая динамика: транзитивное замыкание, порядок перемножения матриц, разрезание многоугольника. Жадные алгоритмы, нематроидные. Жадные алгоритмы, приближенные.

Тема 4.6 Графы.

Поиск в глубину: топсорт, проверка двудольности, нахождение компонент связности, сильной связности, поиск циклов. Поиск в ширину, небольшие его обобщения. Алгоритмы; Форд-Беллман. Дейкстра.

IV. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Практические занятия (72 часа)

Практическое задание №1 Реализация классических алгоритмов на различных языках высокого уровня, использование разных стилей программирования (операционный, ситуационный, функциональный, реляционный, параллельный, объектно-ориентированный)

Практическое задание №2 Особенности и базовые концепции языков Симула-67, Смолток, Рефал, Ада, Модула-2, Оберон и пр. Работа со строками на различных этих языках высокого уровня.

Практическое задание №3 Реализация наследуемости от модульности до объектной ориентации. Разработка алгоритмов средней сложности на различных языках высокого уровня.

Практическое задание №4 Концепция параллелизма в различных языках высокого уровня.

Практическое задание №5 Реализация структур данных. Стеки (ближайший меньший, обратная польская запись), очереди, связанные списки. Динамические массивы, амортизированная сложность (анализ сложности ближайшего меньшего). Дек. Наивный ассоциативный массив, разные его варианты.

Практическое задание №6 Алгоритмы сортировки. Задача сортировки. Квадратичные алгоритмы. Условие наличия линейного порядка на элементах. Сортировка слиянием, сортировка списков. Разделяй и властвуй, быстрая сортировка, порядковые статистики. Линейные сортировки: подсчетом, карманная, поразрядная. Алгоритмы Кнута, Мориса, Пратта.

Практическое задание №7 Алгоритмы поиска. Бинарный поиск. Тернарный поиск.

Практическое задание №8 Хеширование. Идея хеширования, представление ключей целыми числами. Проблема коллизий, способы разрешения:

таблица, открытая адресация, преодоление кластеризации в открытой адресации. Виды хеш-функций: деление, умножение, универсальное хеширование. Хеш кукушки и Робин-Гуда.

Практическое задание №9 Деревья. Деревья поиска. Балансировка, AVL-деревья. Другие деревья: Красно-черные, Б-деревья. Рандомизированные деревья. Скин-листы.

Практическое задание №10 Переборные алгоритмы. Обход дерева решений. Перебор стандартных конфигураций: Перестановок, Размещений и т.д. Перебор разбиений суммы. Коды Грея. NP задачи: рюкзак, Гамильтонов путь. Отсечения.

Практическое задание №11 Динамическое программирование. LCS, LOS.

V. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине в семестре

№ п/п, название	Дата/сроки выполнения	Вид СРС	Примерные нормы времени на выполнение	Форма контроля
1. Принципы и методы программирования, использование разных стилей программирования (операционный, ситуационный, функциональный, реляционный, параллельный, объектно-ориентированный)	Третья неделя семестра	ИДЗ	2 недели	Коллоквиум
2. Классификация и описание языков программирования,	Четвертая неделя семестра	ИДЗ	1 неделя	Коллоквиум
3. Особенности и базовые концепции языков Симула-67, Смолток, Рефал, Ада, Модула-2, Оберон и пр. Работа со строками на различных языках высокого уровня.	Пятая неделя семестра	ИДЗ	1 неделя	Коллоквиум
4. Реализация классических алгоритмов на различных языках высокого уровня.	Шестая неделя семестра	ИДЗ	1 неделя	Проверка программы
5. Реализация алгоритмов работы со строками на различных языках высокого уровня	Седьмая неделя семестра	ИДЗ	3 неделя	Проверка программы
6. Реализация наследуемости от модульности до объектной ориентации. Разработка алгоритмов средней сложности на различных языках высокого уровня.	Десятая неделя семестра	ИДЗ	2 недели	Проверка программы
7. Концепция параллелизма в различных языках высокого уровня Концепция параллелизма в различных языках высокого уровня	Пятнадцатая неделя семестра	ИДЗ	2 недели	Проверка программы
8. Подготовка к зачету	Последняя неделя	ИДЗ	1 неделя	Итоговое тестирование

Критерии оценивания

В течение семестра студентам последовательно выдается набор из 4-х лабораторных работ, каждая из которых имеет вес от 15% до 20%. Посещаемость занятий также учитывается и имеет вес 10%. Для получения зачета с оценкой «отлично» необходимо иметь итоговый балл не ниже 80%, зачета с оценкой «хорошо» – необходимо иметь итоговый балл не ниже 65%, зачета с оценкой «удовлетворительно» – необходимо иметь итоговый балл не ниже 50%,

Характеристика заданий самостоятельной работы

Самостоятельная работа направлена на развитие интеллектуальных умений, комплекса универсальных (общекультурных) и профессиональных компетенций, повышение творческого потенциала магистрантов и заключается в:

- поиске, анализе, структурировании и презентации информации;
- разработке учебного программного продукта;
- исследовательской работе и участии в научных студенческих семинарах и олимпиадах;
- анализе научных публикаций по заранее определенной преподавателем теме.

Содержание самостоятельной работы студентов по дисциплине

1. Изучение аспектов теории, не уложившихся в лекции.
2. Разработка алгоритмов и программ при выполнении лабораторных работ.
3. Подготовка к зачету.

Требования к представлению и оформлению результатов самостоятельной работы

1. Устный ответ по указанной теме.
2. Исходный код программ заданий по соответствующим темам.

КРИТЕРИИ ОЦЕНКИ ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

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

Критерии оценки:

1. Обзор литературных источников (10 баллов)
2. Качество устного изложения содержания темы (10 баллов)
3. Выполнение каждого лабораторного задания (10 баллов)

Учебно-методические материалы для подготовки к практическим занятиям

Примеры заданий:

Практическое задание №1 Реализация классических алгоритмов на различных языках высокого уровня, использование разных стилей программирования (операционный, ситуационный, функциональный, реляционный, параллельный, объектно-ориентированный)

Реализация каждого из заданий выполняется одновременно на двух языках: Python и Perl. Задания подразумевают использование базовых типов данных, массивов, управляющих конструкций, множественного присваивания.

Для каждого из заданий определяется использование определенного стиля программирования (операционный, ситуационный, функциональный, реляционный, параллельный, объектно-ориентированный).

Практическое задание №2 Реализация алгоритмов работы со строками на различных языках высокого уровня. Особенности и базовые концепции различных языков.

Реализация каждого из заданий выполняется на одном из языков: Симула-67, Смолток, Рефал, Ада, Модула-2, Оберон и пр. Работа со строками на различных этих языках высокого уровня.. В заданиях требуется использование регулярных выражений с использованием НДКА, выбор структур данных, квантификаторов, множества символов, средств работы со строками.

Практическое задание №3 Реализация наследуемости от модульности до объектной ориентации. Разработка алгоритмов средней сложности на различных языках высокого уровня.

Реализация каждого из заданий предполагает использование наследуемости. Задания требуют работу с моделями динамического распределения памяти. Темы Лабораторной проработки: списки, кортежи, отрицательные индексы списков, list comprehension, слайсинг, ассоциативный массив словари, множества.

Практическое задание №4 Объектно-ориентированное программирование в различных языках высокого уровня. Концепция параллелизма в различных языках высокого уровня

Задания на темы: описание классов, объекты классов, функции, механизмы передачи параметров.

Рекомендации по самостоятельной работе студентов

Планирование и организация времени, отведенного на выполнение заданий самостоятельной работы.

Изучив график выполнения самостоятельных работ, следует правильно её организовать. Рекомендуется изучить структуру каждого задания, обратить внимание на график выполнения работ, отчетность по каждому заданию

предоставляется в последнюю неделю согласно графику. Обратите внимание, что итоги самостоятельной работы влияют на окончательную оценку по итогам освоения учебной дисциплины.

Рекомендации по работе с литературой

В процессе подготовки к практическим занятиям, студентам необходимо обратить особое внимание на поиск и на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме занятия, что позволяет студентам проявить свою индивидуальность в рамках выполнения индивидуального проекта, выявить широкий спектр мнений по изучаемой проблеме.

Существуют два метода работы над источниками:

– сплошное чтение обязательно при изучении учебника, глав монографии или статьи, то есть того, что имеет учебное значение. Как правило, здесь требуется повторное чтение, для того чтобы понять написанное. Старайтесь при сплошном чтении не пропускать комментарии, сноски, справочные материалы, так как они предназначены для пояснений и помощи. Анализируйте рисунки (карты, диаграммы, графики), старайтесь понять, какие тенденции и закономерности они отражают;

– метод выборочного чтения дополняет сплошное чтение; он применяется для поисков дополнительных, уточняющих необходимых сведений в словарях, энциклопедиях, иных справочных изданиях. Этот метод крайне важен для повторения изученного и его закрепления, особенно при подготовке к зачету.

Для того чтобы каждый метод принес наибольший эффект, необходимо фиксировать все важные моменты, связанные с интересующей Вас темой.

Тезисы – это основные положения научного труда, статьи или другого произведения, а возможно, и устного выступления; они несут в себе большой объем информации, нежели план. Простые тезисы лаконичны по форме; сложные – помимо главной авторской мысли содержат краткое ее обоснование и доказательства, придающие тезисам более весомый и убедительный характер.

Конспект – это способ самостоятельно изложить содержание книги или статьи в логической последовательности. Конспектируя какой-либо источник, надо

стремиться к тому, чтобы немногими словами сказать о многом. В тексте конспекта желательно поместить не только выводы или положения, но и их аргументированные доказательства (факты, цифры, цитаты).

V. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы / темы дисциплины	Код и наименование индикатора достижения	Результаты обучения	Оценочные средства	
				текущий контроль	промежуточная аттестация
1	МОДУЛЬ 1. Принципы программирования на языках высокого уровня	ОПК - 2 ОПК – 5 ОПК – 6 ОПК - 8	Знает	УО-1, УО-4, ПР-6	Вопросы к зачету
			Умеет		
			Владеет		
2	МОДУЛЬ 2. Обзор языков программирования высокого уровня	ОПК - 2 ОПК – 5 ОПК – 6 ОПК - 8	Знает	УО-1, УО-4, ПР-6	Вопросы к зачету
			Умеет		
			Владеет		
3	МОДУЛЬ 3. Основы программирования на разных языках	ОПК - 2 ОПК – 5 ОПК – 6 ОПК - 8	Знает	УО-1, УО-4, ПР-6	Вопросы к зачету
			Умеет		
			Владеет		
4	МОДУЛЬ 4. Алгоритмы программирования	ОПК - 2 ОПК – 5 ОПК – 6 ОПК - 8	Знает	УО-1, УО-4, ПР-6	Вопросы к зачету
			Умеет		
			Владеет		

VI. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Маккинни, У. Python и анализ данных / У. Маккинни ; перевод с английского А. А. Слинкина. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2020. — 540 с. — ISBN 978-5-97060-590-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131721>
2. Головин И. Г., Волкова И. А. Языки и методы программирования: учебник для вузов/ И. Г Головин. Москва: Академия, 2012 – 304 с.

3. Молдованова, О. В. Языки программирования и методы трансляции [Электронный ресурс]: учебное пособие / О. В. Молдованова. — Электрон. текстовые данные. — Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2012. — 134 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/54809.html>
4. Кауфман, В.Ш. Языки программирования. Концепции и принципы [Электронный ресурс] / В.Ш. Кауфман. — Электрон. дан. — Москва : ДМК Пресс, 2010. — 464 с. — Режим доступа: <https://e.lanbook.com/book/1270>
5. Серебряков, В.А. Теория и реализация языков программирования [Электронный ресурс] : учебное пособие / В.А. Серебряков. — Электрон. дан. — Москва : Физматлит, 2012. — 236 с. — Режим доступа: <https://e.lanbook.com/book/5294>. — Загл. с экрана.
6. Языки программирования: Учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. - 3-е изд., перераб. и доп. - М.: Форум: ИНФРА-М, 2015. - 400 с.: 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-91134-744-4 - Режим доступа: <http://znanium.com/catalog/product/493421>
7. Алгоритмизация и языки программирования [Электронный ресурс] : учебно-методический комплекс / сост. О. С. Ахметова, Б. К. Тульбасова, А. Ж. Абишева. — Электрон. текстовые данные. — Алматы: Нур-Принт, Казахский национальный педагогический университет имени Абая, 2012. — 165 с. — 9965-894-95-7. — Режим доступа: <http://www.iprbookshop.ru/67008.html> ISO/IEC 9075-2:2011 Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation). — ISO/IEC, 2011 — 1483 с.
8. Васильев, А. Н. Python на примерах : практический курс по программированию / А. Н. Васильев. — 2-е изд. — Санкт-Петербург : Наука и Техника, 2017. — 432 с. — ISBN 978-5-94387-741-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/73043.html>
9. Златопольский, Д. М. Основы программирования на языке Python / Д. М. Златопольский. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2018. — 396 с. — ISBN 978-5-97060-641-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131683>

Дополнительная литература
(печатные и электронные издания)

1. Языки программирования: Учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. - 3-е изд., перераб. и доп. - М.: Форум: ИНФРА-М, 2015. - 400 с.: 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-91134-744-4 - Режим доступа: <http://znanium.com/catalog/product/493421>
2. Алгоритмизация и языки программирования [Электронный ресурс] : учебно-методический комплекс / сост. О. С. Ахметова, Б. К. Тульбасова, А. Ж. Абишева. — Электрон. текстовые данные. — Алматы: Нур-Принт, Казахский национальный педагогический университет имени Абая, 2012. — 165 с. — 9965-894-95-7. — Режим доступа: <http://www.iprbookshop.ru/67008.html>
ISO/IEC 9075-2:2011 Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation). — ISO/IEC, 2011 — 1483 с.
3. Васильев, А. Н. Python на примерах : практический курс по программированию / А. Н. Васильев. — 2-е изд. — Санкт-Петербург : Наука и Техника, 2017. — 432 с. — ISBN 978-5-94387-741-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/73043.html>
4. Златопольский, Д. М. Основы программирования на языке Python / Д. М. Златопольский. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2018. — 396 с. — ISBN 978-5-97060-641-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131683>
5. Албахари Дж., Албахари Б. С# 5.0. Справочник. Полное описание языка. — М.: Издательский дом «Вильямс», 2013. —1008 с.
6. Александров А., Бабенко Л., Кауфман В. Стандартизация языков программирования. —Киев : Тэхника , 1989. —160 с. — Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:746212&theme=FEFU>
7. Аммерааль Л. STL для программистов на C++. [Электронный ресурс] : . — Электрон. дан. — М. : ДМК Пресс, 2006. — 240 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1218 — Загл. с экрана.
8. Баженова И. Языки программирования: учебник для вузов. — М.: Академия, 2012. — 358 с. — Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:668317&theme=FEFU>
9. Бакетт К. Dart в действии. — М. : ДМК Пресс, 2013. — 528 с.
10. Бейтс М. CoffeeScript. Второе дыхание JavaScript. — М.: ДМК Пресс, 2014. — 312 с.

11. Блэнди, Д. Программирование на языке Rust. Быстрое и безопасное системное программирование / Д. Блэнди, Д. Орендорф ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2018. — 550 с. — ISBN 978-5-97060-236-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/112925>
12. Дмитриев, В. Г. Скриптовый язык программирования PERL : учебное пособие / В. Г. Дмитриев, Т. Х. Агишев. — Уфа : БГПУ имени М. Акмуллы, 2017. — 83 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/105308>
13. Душкин, Р.В. Функциональное программирование на языке Haskell [Электронный ресурс]: учебник. — Электрон. дан. — М.: ДМК Пресс, 2008. — 608 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1247 — Загл. с экрана.
14. Иерузалымски, Р. Программирование на языке Lua / Р. Иерузалымски. — 3-е изд. — Москва : ДМК Пресс, 2014. — 382 с. — ISBN 978-5-94074-767-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/90107>
15. Керниган Б., Ритчи Д. Язык программирования С. — М.: Издательский дом «Вильямс», 2015. — 304 с.
16. Керниган, Б. В. Язык программирования С : учебник / Б. В. Керниган, Д. М. Ричи. — 2-е изд. — Москва : ИНТУИТ, 2016. — 313 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100543>
17. Леоненков А.В. Языки программирования [Электронный ресурс]: [учебный курс]. — Электрон. текстовые данные. — М.: БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий (ИНТУИТ), 2006. — 320 с. — Режим доступа: <http://www.iprbookshop.ru/22416> . — ЭБС «IPRbooks», по паролю
18. Пратт Т. Языки программирования: разработка и реализация. — М.: Мир, 1979. — 574 с. Хигман Б. Сравнительное изучение языков программирования. — М.: Мир, 1974. — 204 с.
19. Рамальо Л. Python. К вершинам мастерства. — М. : ДМК Пресс, 2016. — 768 с.
20. Роберт У. Себеста. Основные концепции языков программирования. — М.: Издательский дом «Вильямс», 2001. — 672 с.
21. Самарев, Р. С. Основы языка программирования Ruby : методические указания / Р. С. Самарев. — Москва : МГТУ им. Н.Э. Баумана, 2015. — 98 с. — ISBN 978-5-7038-4219-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/103550>

22. Саммерфильд М. Программирование на языке Go. — М. : ДМК Пресс, 2013. — 580 с.
23. Страуструп Б. Программирование. Принципы и практика с использованием C++. — М.: Издательский дом «Вильямс», 2016. — 1328 с.
24. Сузи, Р. А. Язык программирования Python : учебное пособие / Р. А. Сузи. — 2-е изд. — Москва : ИНТУИТ, 2016. — 350 с. — ISBN 5-9556-0058-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100546>
25. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. — М.: Издательский дом «Вильямс», 2015. — 1312.
26. Уорбэртон Р. Лямбда-выражения в Java 8. Функциональное программирование. — М.: ДМК Пресс, 2014. — 192 с.
27. Усов В. Swift. Основы разработки приложений под iOS. — СПб.: Питер, 2016. — 304 с.
28. Фаулер М. Предметно-ориентированные языки программирования. — М.: Издательский дом «Вильямс», 2011. — 576 с.
29. Фитцджеральд М. Изучаем Ruby. — СПб.: БХВ-Петербург, 2008. — 336 с.
30. Фултон Х. Программирование на языке Ruby / Фултон Х.. — Саратов : Профобразование, 2019. — 685 с. — ISBN 978-5-4488-0118-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/89871.html>
31. Хорев П.Б. Технологии объектно-ориентированного программирования : учебное пособие / П. Б. Хорев. — М. : Академия , 2008. — 447 с.
32. Хорстманн К., Корнелл Г. Java. Библиотека профессионала. Том 2. Расширенные средства программирования. — М.: Издательский дом «Вильямс», 2016. — 1008 с.
33. Шилдт Г. Java 8. Полное руководство. — М.: Издательский дом «Вильямс», 2015. — 1376 с.
34. Шилдт Г. Java. Полное руководство. — М.: Издательский дом «Вильямс», 2012. — 1104 с.
35. Шиндин, А. В. Язык программирования математических вычислений JULIA. Базовое руководство : учебно-методическое пособие / А. В. Шиндин. — Нижний Новгород : ННГУ им. Н. И. Лобачевского, 2016. — 24 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/153036>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. http://www.tiobe.com/tiobe_index?page=index — TIOBE Index
2. <http://www.ruby-lang.org/en/> — Ruby Home Page
3. <https://www.python.org/> — Python Home Page
4. <http://www.schemers.org/> — Schemer Home Page
5. <https://swiftlang.eu/> — Swift Home Page
6. <http://www.perl.com/> — Perl Home Page
7. <http://www.levenez.com/lang/> — Computer Languages History
8. <http://www.oracle.com/technetwork/java/index.html> — Java by Sun Microsystems
9. <http://javascript.ru/> — Язык JavaScript
10. <http://www.ecma-international.org/publications/standards/Ecma-262.htm> — Standard ECMA-262 ECMAScript® 2015 Language Specification

Профессиональные базы данных и информационные справочные системы

1. База данных Scopus <http://www.scopus.com/home.ur0l>
2. База данных Web of Science <http://apps.webofknowledge.com/>
3. Научная электронная библиотека <https://www.elibrary.ru/>
4. Общероссийский математический портал Math-Net.Ru <http://www.mathnet.ru>
5. Электронная библиотека диссертаций Российской государственной библиотеки <http://diss.rsl.ru/>
6. Электронная библиотека Европейского математического общества <https://www.emis.de/>
7. Электронные базы данных EBSCO <http://search.ebscohost.com/>

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.
2. Электронно-библиотечная система издательства «Лань».
3. Электронная библиотека "Консультант студента".
4. Электронно-библиотечная система IPRbooks.
5. Информационная система "ЕДИНОЕ ОКНО доступа к образовательным ресурсам".
6. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.
7. Электронный журнал успеваемости система AWorks//
8. Система автоматического тестирования программ CATS

9. Сообщество пользователей SQL [<http://sql.ru>].
10. Утилита для администрирования данных FlameRobin [<http://flamerobin.org>]

VII. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Планирование и организация времени, отведенного на изучение дисциплины. Приступить к освоению дисциплины следует незамедлительно в самом начале учебного семестра. Рекомендуется изучить структуру и основные положения Рабочей программы дисциплины. Обратите внимание, что кроме практических занятий планируется самостоятельная работа, итоги которой влияют на окончательную оценку по итогам освоения учебной дисциплины. Все задания (аудиторные и самостоятельные) необходимо выполнять и предоставлять на оценку в соответствии с графиком.

В процессе изучения материалов учебного курса предлагаются следующие формы работ: чтение лекций, лабораторные занятия, задания для самостоятельной работы.

Практические занятия акцентированы на наиболее принципиальных и проблемных вопросах курса и призваны стимулировать выработку практических умений.

Особо значимой для профессиональной подготовки студентов является *самостоятельная работа* по курсу. В ходе этой работы студенты отбирают необходимый материал по изучаемому вопросу и анализируют его. Студентам необходимо ознакомиться с основными источниками, без которых невозможно полноценное понимание проблематики курса.

Освоение курса способствует развитию навыков обоснованных и самостоятельных оценок фактов и концепций. Поэтому во всех формах контроля знаний, особенно при сдаче зачета, внимание обращается на понимание проблематики курса, на умение практически применять знания и делать выводы.

Работа с литературой. Рекомендуется использовать различные возможности работы с литературой: фонды научной библиотеки ДВФУ и электронные библиотеки (<http://www.dvfu.ru/library/>), а также доступные для использования другие научно-библиотечные системы.

Подготовка к зачету. К сдаче зачета допускаются обучающиеся, *выполнившие все задания (практические, самостоятельные)*, предусмотренные учебной программой дисциплины, посетившие не менее 85% аудиторных занятий.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень материально-технического и программного обеспечения дисциплины приведен в таблице.

Материально-техническое и программное обеспечение дисциплины

Для проведения учебных занятий по дисциплине, а также для организации самостоятельной работы студентам доступно следующее лабораторное оборудование и специализированные кабинеты, соответствующие действующим санитарным и противопожарным нормам, а также требованиям техники безопасности при проведении учебных и научно-производственных работ.

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
690922, Приморский край, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корпус 20(D), ауд. D738, D654/D752, D412/D542, D818, D741, D945, D547, D548, D732;	Мультимедийная лекционная аудитория (мультимедийный проектор, настенный экран, документ-камера	1) Windows Server CAL 2019. Договор № ЭА-261-18. Лицензия 30.06.2020. 2) SharePoint Server Standard CAL 2019. Договор № ЭА-261-18. Лицензия 30.06.2020
690922, Приморский край, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корпус 20(D), (D733, D734, D546, D549),	доска, 15 персональных компьютеров), оснащенный компьютерами с операционной системой Microsoft Windows 10 (минимальные требования: процессор Intel Core i7 5700 3,2 ГГц, 16 ГБ оперативной памяти, видеокарта Nvidia GeForce GTX970, жёсткий диск 500ГБ, SSD 128ГБ), презентационными (мультимедийный проектор, настенный экран, документ-камера) системами, с подключением к общекорпоративной компьютерной сети ДВФУ и сети Интернет.	1С Предприятия8 (8.2), 7-Zip, ABBYY Lingvo12, Alice 3, Anaconda3, Autodesk, CodeBlocks, CorelDRAW X7, Dia, Directum4.8, DosBox-0.74, Farmanager, Firebird 2.5, FlameRobin, Foxit Reader, Free Pascal, Geany, Ghostscript, Git, Greenfoot, gsview, Inscapе0.91, Java, Java development Kit, Kaspersky, Lazarus, LibreOffice4.4, MatLab R2017b, Maxima 5.37.2, Microsoft Expression, Microsoft Office 2013, Microsoft Silverlight, Microsoft Silverlight 5SDK-русский, MicrosoftSistem Center, Microsoft Visial Studio 2012, MikTeX2.9, MySQL, NetBeans, Notepad++, Oracle VM VirtualBox, PascalABC.NET, PostgreSQL 9.4, PTC Mathcad, Putty, PyQt GPL v5.4.1 for Pythonv 3.4, Pyton2.7(3.4,3.6), QGIS Brighton, RStudio, SAM CoDeC Pack, SharePoint, Strawberry Perl, Tecnomatix, TeXnicCenter, TortoiseSVN, Unity2017.3.1f1, Veusz, Vim8.1, Visual Paradigm CE, Visual Studio2013, Windows Kits, Windows Phone SDK8.1, Xilinx Design Tools Acrobat ReaderDC, AdobeBridge CS3,

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья в ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащенными туалетными комнатами, табличками информационно-навигационной поддержки.

VIII. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

Для дисциплины «Языки, алгоритмы и методы программирования» используются следующие оценочные средства:

Устный опрос:

1. Собеседование (УО-1)
2. Презентация / сообщение(УО-3)

Письменные работы:

1. Реферат (ПР-4)
2. Практическая работа (ПР-6)
3. Проект (ПР-9)

Устный опрос

Устный опрос позволяет оценить знания и кругозор студента, умение логически построить ответ, владение монологической речью и иные коммуникативные навыки.

Обучающая функция состоит в выявлении деталей, которые по каким-то причинам оказались недостаточно осмысленными в ходе учебных занятий и при подготовке к зачёту.

Собеседование (УО-1) – средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.

Презентация / сообщение (УО-3) – продукт самостоятельной работы обучающегося, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы.

Письменные работы

Письменный ответ приучает к точности, лаконичности, связности изложения мысли. Письменная проверка используется во всех видах контроля и осуществляется как в аудиторной, так и во внеаудиторной работе.

Реферат (ПР-4) – Продукт самостоятельной работы обучающегося, представляющий собой краткое изложение в письменном виде полученных

результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее.

Практическая работа (ПР-6) – средство для закрепления и практического освоения материала по определенному разделу.

Проект (ПР-9) - Конечный продукт, получаемый в результате планирования и выполнения комплекса учебных и исследовательских заданий. Позволяет оценить умения обучающихся самостоятельно конструировать свои знания в процессе решения практических задач и проблем, ориентироваться в информационном пространстве и уровень сформированности аналитических, исследовательских навыков, навыков практического и творческого мышления. Может выполняться в индивидуальном порядке или группой обучающихся

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Оценочные средства для промежуточной аттестации

Промежуточная аттестация студентов по дисциплине «Языки, алгоритмы и методы программирования» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Объектами оценивания выступают:

- степень усвоения теоретических знаний - оценивается по результатам защиты реферата (отчет, ответы на вопросы по теме реферата, участие в дискуссии на защите рефератов);
- уровень овладения практическими умениями и навыками – оценивается в форме защиты индивидуального задания (проекта), выполняемого в рамках практических занятий.

Методические указания по сдаче зачета

Зачет принимается ведущим преподавателем. При большом количестве групп у одного преподавателя или при большой численности потока по распоряжению заведующего кафедрой (заместителя директора по учебной и воспитательной работе) допускается привлечение в помощь ведущему преподавателю других преподавателей. В первую очередь привлекаются преподаватели, которые проводили лабораторные занятия по дисциплине в группах.

В исключительных случаях, по согласованию с заместителем директора Школы по учебной и воспитательной работе, заведующий кафедрой имеет право принять зачет в отсутствие ведущего преподавателя.

Время, предоставляемое студенту на доклад/сообщение должно составлять не более 20 минут (10 минут на доклад+10 минут ответы на вопросы).

Присутствие на зачете посторонних лиц (кроме лиц, осуществляющих проверку) без разрешения соответствующих лиц (ректора либо проректора по учебной и воспитательной работе, директора Школы, руководителя ОПОП или заведующего кафедрой), не допускается. Инвалиды и лица с ограниченными возможностями здоровья, не имеющие возможности самостоятельного передвижения, допускаются зачет с сопровождающими.

Критерии оценивания сообщения/доклада по теме реферата

Баллы (рейтинговой оценки)	Оценка зачета/ экзамена (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	магистрант точно определил содержание и составляющие части задания, умеет аргументировано отвечать на вопросы, связанные с заданием. Продемонстрировано знание и владение навыками самостоятельной исследовательской работы по теме. Фактических ошибок, связанных с пониманием проблемы, нет.
76-85	«зачтено»/ «хорошо»	работа магистранта характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет.
61-75	«зачтено»/ «удовлетворительно»	проведен достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимание базовых основ и теоретического обоснования выбранной темы. Привлечены основные источники по рассматриваемой теме. Допущено не более 2 ошибок в смысле или содержании проблемы
0-60	«не зачтено»/ «неудовлетворительно»	работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Допущено три или более трех ошибок смыслового содержания раскрываемой проблемы

Оценочные средства для текущей аттестации

Текущая аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация проводится в форме контрольных мероприятий (собеседования, презентации, эссе, практических работ, контрольно-расчетных работ, творческого задания) по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

– учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);

– степень усвоения теоретических знаний;

– уровень овладения практическими умениями и навыками по всем видам учебной работы;

– результаты самостоятельной работы.

Вопросы для зачета/ устного опроса

1. Общее понятие языка. Естественные и искусственные языки. Язык программирования
2. Классификация ЯП. Исторический обзор развития ЯП.
3. Схема рассматривания ЯП: базис, средства развития и средства защиты.
4. Классификация типов данных и средств описания данных в ЯП.
5. Простые типы данных, операции над ними.
6. Составные типы данных, операции над ними.
7. Ассоциативные массивы и записи.
8. Управление последовательностью вычислений.
9. Разновидности управляющих конструкций в современных языках программирования.
10. Условные операторы и многовариантные развилки.
11. Циклы. Особенности реализации циклов-итераторов в современных ЯП.
12. Подпрограммы и сопрограммы. Операторы возврата и возобновления. Процедуры и функции в современных ЯП.
13. Передача параметров: семантика и способы реализации.
14. Статический полиморфизм и перегрузка имен подпрограмм.
15. Подпрограммные типы данных.
16. Концепция уникальности типа в традиционных языках программирования и понятие строгой типизации.
17. Понятие инкапсуляции. Понятие абстрактного типа данных (АТД) и его достоинства.
18. Принцип разделения определения, реализации и использования.
19. Походы к определению новых типов данных: модули и классы.

20. Области видимости и пространства имен. Управление видимостью и управление доступом.
21. Преобразование типов. Явные и неявные преобразования. Управление преобразованиями в современных ЯП: проблемы и способы их решения.
22. Классы и перегрузка имен. Перегрузка встроенных знаков операций. Итераторы и индексаторы.
23. Понятие исключительной ситуации (ИС) и его эволюция. ИС и ошибки в программах.
24. Четыре аспекта рассмотрения ИС: определение, возникновение, распространение и обработка.
25. Понятие единичного наследования. Наследование и модель представления объекта в памяти.
26. Иерархии типов, статические и динамические типы в объектно-ориентированных ЯП. Управление видимостью и доступом при наследовании.
27. Понятие о множественном наследовании. Проблемы множественного наследования: конфликт имен, реализация динамического связывания. Динамический полиморфизм.
28. Понятие абстрактного класса (АК). Воплощение концепции АК в современных ЯП.
29. Интерфейс как языковая конструкция. Интерфейсы и иерархии классов. Реализация интерфейсов и ее особенности современных ЯП.
30. Обобщенное программирование. Понятие о родовых объектах. Понятие о статической параметризации и родовых объектах. Достоинства статической параметризации.
31. Обобщенное программирование. Механизм шаблонов в языке
32. Основы логической парадигмы. Исчисление предикатов как язык представления знаний. Виды импликации. Логика предикатов первого порядка, хорновские дизъюнкты. Вычислительная модель. Анализ структуры термов. Принцип Робинсона.
33. Логическое программирование. Логическая программа: основные конструкции, операционная и декларативная семантика, интерпретация, корректность.
34. Логическое программирование. Рекурсивное представление данных и программ. Рекурсивные функции.
35. Основы функциональной парадигмы. Рекурсивные функции и лямбда-исчисление А. Черча.
36. Функциональное программирование. Строго функциональный язык. Представление и интерпретация функциональных программ. Функции высшего порядка.

37. Функциональное программирование. Рекурсия. Простая рекурсия. Параллельная и взаимная рекурсия. Хвостовая рекурсия.

Критерии выставления оценки магистранту на зачете

К зачету допускаются обучающиеся, выполнившие программу обучения по дисциплине, прошедшие все этапы текущей аттестации.

Оценка «отлично» (зачтено) - ставится студенту, если он продемонстрировал сформированность всех вышеперечисленных навыков компетенции (85-100 баллов).

Оценка «хорошо» (зачтено) – если сформированы большинство знаний, умений и навыков, но допускается не более 1 недостаточно освоенного навыка компетенции (65 – 84 балла).

Оценка «удовлетворительно» (зачтено) – если сформированы большинство навыков, но допускается не более 2 недостаточно освоенных навыков компетенции (45 – 64 балла).

Оценка «неудовлетворительно» (не зачтено) выставляется, если практические задания выполнены студентом не в полном объеме, и часть навыков компетенции не сформированы (менее 45 баллов).

Критерии оценивания практических работ

Практические работы проводятся с применением метода проектов. (результатом выполнения всех практических работ должен стать конечный программный продукт (ПП)). Проект может выполняться в индивидуальном порядке или группой обучающихся. Цель проекта и способы ее достижения определяются обучающимся/обучающимися на основе его/их интересов, индивидуальных особенностей, потребностей, мотивов, способностей.

Цель практических работ на практике показать, что «разработка мультипарадигмальных языков программирования состоит, как правило, в том, чтобы позволить программистам использовать лучший инструмент для работы, признавая, что никакая парадигма не решает все проблемы самым лёгким или самым эффективным способом».

Обучающимся необходимо:

1. Определить тему проекта (выбрать предметную область, модель которой можно представить АТД (дерево, граф, многосвязные списки, множества и т.д.), определить класс решаемых задач в этой предметной области).
2. Пройти все этапы разработки программного продукта от анализа требований до реализации (в соответствии с применяемой парадигмой) и тестирования.
3. Провести обзор современных мультипарадигменных языков программирования и сред программирования. Обосновать выбор языка(ов)

программирования для выполнения практических работ.

4. Подготовить отчеты (документацию) по всем этапам разработки ПП для каждой практической работы.

Критерии оценки проекта

Оценка	Требования
«зачтено»	Студент выполняет практическую работу в полном объеме с соблюдением необходимой последовательности проведения, самостоятельно выбирает предметную область, определяет класс решаемых задач в выбранной предметной области, самостоятельно выбирает язык и среду программирования. Грамотно и логично описывает ход работы, правильно формулирует выводы, отчет содержит описание всех этапов разработки программного продукта. Допускается два/три недочёта или одна негрубая ошибка и один недочёт. Работа соответствует требованиям и выполнена в срок.
«не зачтено»	Студент выполнил работу не полностью, объём выполненной части не позволяет сделать правильные выводы; не определяет самостоятельно цель работы; в ходе работы допускает одну и более грубые ошибки, которые не может исправить, или неверно производит наблюдения, измерения, вычисления и т.п.; не умеет обобщать фактический материал. Лабораторная работа не выполнена.