

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет математики и компьютерных наук

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Избранные вопросы языков программирования

Кафедра дискретной математики и информатики
факультета математики и компьютерных наук

Образовательная программа бакалавриата
02.03.02 Фундаментальная информатика и информационные технологии

Направленность(профиль) программы:
Информатика и компьютерные науки

Форма обучения
очная

Статус дисциплины: входит в часть ОПОП формируемую участниками образовательных
отношений

Махачкала, 2022

Рабочая программа дисциплины “Избранные вопросы языков программирования” составлена в 2022 году в соответствии с требованиями ФГОС ВО – бакалавриат по направлению подготовки 02.03.02 – Фундаментальная информатика и информационные технологии от 23 августа 2017 г №808.

Разработчик(и): кафедра дискретной математики и информатики, преподаватель Ибавов Темирлан Ильмутдинович.

Рабочая программа дисциплины одобрена:

на заседании кафедры дискретной математики и информатики от «28» февраля 2022 г., протокол № 6.

Зав. кафедрой  Магомедов А.М.

(подпись)

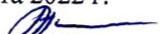
и

на заседании Методической комиссии ФМиКН от «24» марта 2022г., протокол № 4.

Председатель  Ризаев М.К.

(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением «31» марта 2022 г.

Начальник УМУ  Гасангаджиева А.Г.

(подпись)

Аннотация рабочей программы дисциплины

Дисциплина “Избранные вопросы языков программирования” входит в часть ОПОП, формируемую участниками образовательных отношений в форме факультатива программы бакалавриата по направлению 02.03.02 – Фундаментальная информатика и информационные технологии.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики и информатики.

Содержание дисциплины охватывает круг вопросов, связанных с формированием алгоритмического мышления у студентов, объектно-ориентированным программированием, созданием консольных и графических приложений.

Дисциплина нацелена на формирование следующих компетенций выпускника: профессиональных – ПК-2, ПК-4.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия, лабораторные занятия и самостоятельную работу студентов. Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости: 6 семестр - зачет. Объем дисциплины – 1 зачетная единица, в том числе в академических часах по видам учебных занятий:

Семестр	Всего	Учебные занятия				СРС, в том числе экзамен	Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен)
		В том числе					
		Контактная работа обучающихся с преподавателем					
		из них					
		Всего	Лекции	Лабораторные	Практические		
6	36	28	28			8	Зачет
Итого	36	28	28			8	

1. Цели освоения дисциплины

Целями освоения дисциплины “Избранные вопросы языков программирования” являются овладение знаниями в области технологии программирования; подготовка к осознанному использованию, как языков программирования, так и методов программирования. Формирование у студентов научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения. Получение необходимых знаний, умений и навыков в области применения современной вычислительной техники для решения практических задач обработки данных, математического моделирования, информатики, получение высшего профессионального (на уровне бакалавра) образования, позволяющего выпускнику успешно работать в избранной сфере деятельности с применением современных компьютерных технологий.

Основные задачи дисциплины: овладение методами структурного и объектно-ориентированного программирования; закрепление навыков алгоритмизации и программирования, полученных в предыдущих семестрах; создание практической базы для написания качественной выпускной квалификационной работы.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина входит в часть ОПОП, формируемую участниками образовательных отношений в форме факультатива образовательной программы бакалавриата по направлению 02.03.02 – Фундаментальная информатика и информационные технологии и изучается в соответствии с графиком учебного процесса в 6 семестре. Для изучения и освоения дисциплины нужны первоначальные знания полученные при изучении дисциплин «Технологии программирования и работа на ЭВМ», «Технологии и методы

программирования», «Основы web-программирования», «Программирование на основе классов». Знания, умения и навыки, полученные при изучении данной дисциплины, используются, закрепляются и развиваются при проведении учебной практики, курсов численных методов, вычислительного практикума, выполнении курсовых работ и выпускной квалификационной работы.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения и процедура освоения).

Код и наименование компетенции из ОПОП	Код и наименование индикатора достижения компетенций (в соответствии с ОПОП)	Планируемые результаты обучения	Процедура освоения
<p>ПК-2. Способен к преподаванию по программам профессионального обучения, среднего профессионального образования (СПО) и дополнительным профессиональным программам (ДПП), ориентированным на соответствующий уровень квалификации.</p>	<p>ПК-2.1. Знает основные методы решения прикладных задач, современные методы информационных технологий.</p>	<p>Знает: образовательный стандарт и программы профессионального обучения, среднего профессионального образования (СПО) и дополнительным профессиональным программам (ДПП). Умеет: профессионально грамотно пользоваться организационно-методическим и учебно-методическим обеспечением образовательной программы соответствующего уровня. Владеет: психолого-педагогическими и методическими основами преподавания дисциплин математики и информатики.</p>	<p>Устный опрос, письменный опрос;</p>
	<p>ПК-2.2. Умеет корректно оформлять результаты научного труда в соответствии с современными требованиями.</p>	<p>Знает: на достаточно высоком уровне учебные курсы математики и информатики в рамках программы соответствующего уровня. Умеет: оценивать объем материала, необходимого для освоения того или иного программного вопроса в области математики и информатики; устанавливать связи</p>	

		<p>между различными предметными разделами с учетом уровня подготовки и психологии данной аудитории.</p> <p>Владеет: достаточной информацией о современном состоянии развития различных областей математики и информатики и об актуальных вопросах преподавания математики и информатики.</p>	
	<p>ПК-2.3. Имеет практический опыт использования сети Интернет, аннотирования, реферирования, библиографического разыскания и описания, опыт работы с научными источниками.</p>	<p>Знает: разные подходы к определению основных понятий математики; основные понятия информатики; формулировки математических утверждений при различных изменениях их исходных условий; различные языки программирования.</p> <p>Умеет: оценивать объем материала, необходимого для освоения того или иного программного вопроса по математике и информатике по программам профессионального обучения, среднего профессионального образования (СПО) и дополнительным профессиональным программам (ДПП).</p> <p>Владеет: методикой изложения основного материала того или другого раздела математики и информатики по программам профессионального обучения, среднего профессионального образования (СПО) и дополнительным профессиональным программам (ДПП).</p>	
<p>ПК-4. Способность применять в профессиональной</p>	<p>ПК-4.1. Знает современные языки программирования</p>	<p>Знает: основы математического анализа и различные</p>	

<p>деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии.</p>	<p>и методы параллельной обработки данных. Знаком с содержанием Единого Реестра Российских программ для электронных вычислительных машин и баз данных.</p>	<p>приложения дифференциального и интегрального исчисления в математических и естественных науках; современные языки программирования и современные информационные технологии. Умеет: применять дифференциальное и интегральное исчисления для решения различных задач математических и естественных наук; составлять программы на современных языках программирования. Владеет: базовыми методами дифференциального и интегрального исчислений; навыками программирования на современных языках.</p>	
	<p>ПК-4.2. Умеет реализовывать численные методы решения прикладных задач в профессиональной сфере деятельности, пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии.</p>	<p>Знает: области применения дифференциального и интегрального исчисления; различные языки программирования. Умеет: решать задачи, связанные: с исследованием свойств функций и их производных, с изучением функциональных рядов, с оценкой погрешности аппроксимации функций; применять различные языки программирования в численном анализе. Владеет: методами дифференциального исчисления для исследования функций</p>	

		и навыками приложения интегрального исчисления к геометрии, физике.	
	ПК-4.3. Имеет практический опыт разработки интеграции информационных систем.	Знает: методы исследования функций с помощью производных, вычисления интегралов; методы исследования сходимости рядов; численные методы анализа; современные информационные технологии. Умеет: применять методы исследования функций с помощью производных, вычисления интегралов и методы исследования сходимости рядов в численном анализе с использованием современных информационных технологий. Владеет: навыками решения задач численного анализа с использованием методов дифференциального и интегрального исчисления.	

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 1 зачетную единицу, 36 академических часов.

4.2. Структура дисциплины.

4.2.1. Структура дисциплины в очной форме

№ п/п	Разделы и темы дисциплины по модулям	Семестр	Виды учебной работы, включая самостоятельную работу студентов (в часах)				Формы текущего контроля успеваемости и промежуточной аттестации
			Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа в т.ч. экзамен	
<i>Модуль 1. Избранные вопросы языков программирования</i>							
1	Введение в парадигмы программирования		2				Устный опрос
2	Язык Lisp и примеры его использования.		2			2	Устный опрос

3	Язык Scheme.		4		2	Устный опрос
4	Язык Prolog, логическое и декларативное программирование.		4			Устный опрос
5	Язык Refal и ситуационное программирование.		4			Устный опрос
6	Парадигматическая традиция языка Норе.		4			Устный опрос
7	Традиционные парадигмы.		4			Устный опрос
8	Многостилевое (мультипарадигмальное) программирование		4		4	Устный опрос
	Итого за модуль 1	36	28		8	Устный опрос
	ИТОГО:	36	28		8	Зачёт

4.3. Содержание дисциплины, структурированное по темам (разделам).

4.3.1. Содержание лекционных занятий по дисциплине.

Модуль 1. Избранные вопросы языков программирования

Тема 1. Введение в парадигмы программирования

История термина «парадигма» в научно-техническом дискурсе. Обзор основных стилей программирования. Парадигма как стиль мышления; примеры парадигм в традиционных (известных студентам) языках программирования. Влияние языка программирования на мышление. Уровни поддержки отдельных парадигм в языках программирования. Рекурсия как парадигма; виды рекурсии. Примеры решения простых задач в разных парадигмах.

Тема 2. Язык Lisp и примеры его использования.

Понятие S-выражения, атомы, точечные пары, списки, точечные списки. Вычисление S-выражений, символы, функции и спецформы; форма QUOTE и её сокращённая запись. Основные встроенные примитивы. Описание пользовательских функций. Примитив EVAL, интерпретирующее исполнение как парадигма. Разрушающие примитивы; пример создания кольцевых структур. Неизменность констант. Разделяемость списков и её обнаружение. Сборка мусора. Функция как объект данных. Функционалы. Безымянные функции; примитив FUNCTION. Особая роль символа LAMBDA. Локальные связывания, форма LET. Лексическое и динамическое связывание; функция как замыкание. Редукция списков. Левая и правая редукция, примеры. Остаточная (хвостовая) рекурсия. Преобразование рекурсии в остаточную с использованием накапливающего параметра. Макросы языка Lisp.

Тема 3. Язык Scheme.

Основные видимые отличия Scheme от традиционных диалектов Лиспа: обозначение пустого списка и логической лжи, функция как значение символа, порядок вычисления пр

стой формы, примитив DEFINE, вычисляемость LAMBDA -списков (и отсутствие потребности в аналоге примитива FUNCTION), создание вариативных функций. Соглашения об именовании предикатов и разрушающий примитивов. Континуации, их возможности, примеры применения. Континуация как обобщение механизма обработки исключений.

Тема 4. Язык Prolog, логическое и декларативное программирование.

Основы языка Prolog: атомы, функции, списки, цели, предложения, факты. Порядок выполнения программы: связывание переменных, локализация переменных в предложении, унификация, множественность решений, развилки, поиск с возвратами. Отсечение, отрицание, арифметические примитивы, основные встроенные примитивы. Prolog -машина как автомат для перебора. Миф о возможности применения метода резолюций и его истоки. Ограничения логической семантики Пролога, обусловленные поиском вглубь; технические причины несоответствия пролог - вычислителя нормам математической логики. Понятие процедурной семантики. Возможность поиска вширь, чистое логическое программирование, краткая характеристика языка Datalog. Парадигма декларативного программирования; логическое и декларативное программирование в применении к управлению базами данных, SQL как пример декларативного языка. База данных в Прологе, примитивы assert/retract. Примитивы findall, bagof, setof и возможность их реализации средствами Пролога. Устаревшие парадигматические особенности Пролога, подход к обработке строк.

Тема 5. Язык Refal и ситуационное программирование.

История создания Рефала. Основные принципы работы Рефал -вычислителя. Диалект Рефал -5 и его особенности; возможность поиска с возвратами. Парадигматическая характеристика языка Рефал, отсутствие функций как объекта данных, «автоматическая» оптимизация от точной рекурсии.

Тема 6. Парадигматическая традиция языка Норе.

Ленивые вычисления и их возможности. Строгая типизация, автоматический вывод типов. Функциональный тип, понятие карринга. Синтаксис и основные возможности языка Норе (диалект интерпретатора Nopeless).

Тема 7. Традиционные парадигмы.

Основные характеристики императивного и императивно-процедурного программирования, термин «фоннеймановское программирование». Парадигматическое различие между языками Pascal и C, побочный эффект как парадигма, язык C как язык побочных эффектов. Объектно -ориентированное программирование и различные терминологические подходы к его описанию. Разграничение между абстрактными типами

данных и объектами в смысле ООП. Командно-скриптовое программирование; строка текста как универсальное представление для программы и данных. ASCII-текст как парадигма программирования

Тема 8. Многостилевое (мультипарадигмальное) программирование

Мотивация многостилевого программирования; технические сложности интеграции разнородных программистских инструментов. Основные подходы к многостилевой интеграции и их характеристики.

5. Образовательные технологии

В процессе обучения для самостоятельной работы студентов используются операционные системы семейства Unix (Linux или FreeBSD), интерпретатор SBCL, компилятор Chicken Scheme, система SWI-Prolog, интерпретатор Hopeless. Студентам для самостоятельной работы предоставляется удалённый доступ к системе типа Unix, в которой установлены вышеперечисленные инструментальные средства.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

6.1. Виды и порядок выполнения самостоятельной работы.

1. Изучение конспектов лекций и рекомендованной литературы.
 2. Подготовка к опросу на практических занятиях.
 3. Решение задач и упражнений.
 4. Подготовка к коллоквиуму и контрольным работам.
 5. Поиск материала на интернет-форумах.
 6. Подготовка к экзамену
- Примерное распределение времени самостоятельной работы студентов.

Виды самостоятельной работы с указанием часов

Вид самостоятельной работы	Примерная трудоемкость в часах		
	Очная	Очно-заочная	Заочная
Текущая СРС			
Изучение конспектов лекций и рекомендованной литературы	2		
Подготовка к опросу на практических занятиях	2		
Решение задач и упражнений	1		
Подготовка к коллоквиуму и контрольным работам	1		
Поиск материала на интернет-форумах	1		
Подготовка к экзамену	1		

Итого СРС:	8		
------------	---	--	--

6.2. Порядок контроля:

1. Опрос на лабораторном занятии
2. Проверка выполнения домашних заданий и контрольных работ
3. Коллоквиум
4. Зачет

Текущий контроль:

1. Проверка программ на языке высокого уровня по заданиям;
2. Проверка выполнения домашних заданий;
3. Промежуточная аттестация в форме письменной работы.

Текущий контроль включает, кроме еженедельного опроса и проверки знаний по текущему материалу, ведение электронного журнала посещаемости, проверку выполнения компьютерных программ. Подразумевается непрерывное общение по электронной почте (общение по скайпу не целесообразно, т.к. не позволяет осуществлять доскональную проверку заданий).

Промежуточный контроль проводится в виде письменной работы, рассчитанной на 20- 30 минут.

Итоговый контроль проводится в виде письменной работы с обязательным устным собеседованием по результатам предварительной проверки.

Критерии выставления оценок «отлично», «хорошо», «удовлетворительно» определяются степенью владения материалом и достигнутым уровнем компетентности в решении задач дискретной математики. В исключительных случаях учитываются успехи на всероссийских олимпиадах и конкурсах по номинации данной дисциплины.

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Список вопросов для зачета.

1. Понятие "парадигма программирования" и его история. Взаимосвязь языков программирования и парадигм программирования.
2. Рекурсия. Виды рекурсии. Примеры.
3. Остаточная (хвостовая) рекурсия.
4. Императивное и императивно-процедурное программирование: ключевые понятия.
5. Язык Лисп. Понятие S-выражения.
6. Язык Лисп. Функции высоких порядков, мапперы.
7. Язык Лисп. Лексическое и динамическое связывание. Лексические замыкания.
8. Функция REDUCE и примеры её использования.
9. Понятие функционального программирования. Ключевые особенности и термины.
10. Основные понятия языка Пролог: атом, терм, предикат, переменная. Сопоставление

(унификация).

11. Выполнение программы на Прологе. Декларативная и процедурная семантика. Отрицание и отсечение.

12. Инвертирование предикатов в Прологе. Причины, по которым инвертирование может не работать.

13. Логическое программирование и базы данных. Понятие декларативного программирования.

14. Основы языка Рефал. Ситуационное программирование. Поиск с возвратами в Рефале-5.

15. Понятие ленивых вычислений. Достоинства и недостатки.

16. Основы языка Норе.

17. Обработка потенциально бесконечных структур данных с помощью ленивых вычислений.

18. Командно-скриптовые языки программирования и их ключевые особенности.

19. «Текст» как универсальное представление.

20. Основы объектно-ориентированного программирования. Понятие класса как множества.

21. Наследование в ООП. Наследование как сужение множества. Виртуальные методы.

22. Полиморфизм, его теоретико-множественное и техническое описание.

23. Обобщенное программирование.

24. Обработка исключительных ситуаций и её связь с основными стилями программирования.

25. Событийно-ориентированное программирование.

26. Основные методы многостилевой (мультипарадигмальной) интеграции.

Типовые контрольные задания

ВАРИАНТ - 1

1. Порядковые типы данных.

2. Составной оператор, оператор выбора.

3. Вычислить: $n!$

4. В двумерном вещественном массиве 3×4 найти строку с наименьшим элементом.

5. Пусть заданы координаты: $x_1, y_1; x_2, y_2; x_3, y_3$ вершин треугольника и координаты x, y любой точки на плоскости. Определить: лежит ли точка с заданными координатами x, y внутри треугольника?

ВАРИАНТ – 2

1. Строки. Процедуры и функции для работы со строками.

2. Условный оператор, оператор цикла с постусловием.

3. Вычислить: $\sin(x) + \sin(xn) + \dots + \sin(xn)$;

4. В двумерном целочисленном массиве 3×4 найти сумму элементов столбца с максимальным элементом.

5. Проверить корректность расстановки скобок в арифметическом выражении.

ВАРИАНТ – 3

1. Множества. Описание и операции над множествами.

2. Оператор безусловного перехода. Оператор цикла с предусловием.
3. Вычислить: $A*(A+1)*(A+2)*\dots*(A+n)$;
4. В заданной строке найти количество слов. Слова могут быть разделены и несколькими пробелами.
5. По некоторому каналу связи передается сообщение, имеющее вид последовательности нулей и единиц. Из-за помех возможен ошибочный прием некоторых сигналов: нуль может быть воспринят как единица и наоборот. Для повышения вероятности правильного приема сигналов было решено передавать каждый сигнал трижды. Теперь передатчик вместо 1 всегда передает 111, а вместо 0 всегда 000. Вам предлагается написать программу, которая будет восстанавливать исходное сообщение. При передаче могли произойти ошибки, поэтому вместо каждой тройки цифр программа должна вывести ту цифру, которая встречается в этой тройке по крайней мере два раза.

ВАРИАНТ – 4

1. Тип запись. Тип массив.
2. Оператор цикла for.
3. Вычислить: $\text{Cos}(x)+\text{Cos}(2*x)+\dots+\text{Cos}(n*x)$;
4. В двумерном целочисленном массиве 3×4 найти сумму элементов главной диагонали.
5. Даны координаты центров столбов, вбитых в вершины выпуклого многоугольника. Известен диаметр столбов. Найти длину натянутой вокруг столбов нити.

8. Учебно-методическое обеспечение дисциплины.

а) основная литература:

1. Э. Хьювёнен, Й. Сеппянен. Мир Лиспа. В 2-х т. Т. 1: Введение в язык Лисп и функциональное программирование. М.: Мир, 1990. 447 с.
2. А. Филд, П. Харрисон. Функциональное программирование. М.: Мир, 1993. 637 с.
3. Х. Абельсон, Д. Д. Сассман. Структура и интерпретация компьютерных программ. М.: Добросвет, 2006. 608 с.
4. И. Братко. Программирование на языке Пролог для искусственного интеллекта. М.: Мир, 1990. 560 с.
5. В. Ш. Кауфман. Языки программирования: концепции и принципы. М.: ДМК Пресс, 2010. 464 с.
6. Т. Бадд. Объектно-ориентированное программирование в действии. СПб.: Питер, 1997. 464 с.

б) дополнительная литература:

1. Т. Кун, Структура научных революций. М.: Прогресс, 1997. 300 с.
2. С.Черри, Г.Готлоб, Л.Танка. Логическое программирование и базы данных. М.: Мир, 1992. 352 с.
3. Дж. Малпас. Реляционный язык Пролог и его применение. М.: Наука, 1990. 464 с.
4. Г. Буч. Объектно-ориентированный анализ и проектирование. СПб.: «Невский диалект», 1999. 560 с.
5. G. L. Steele. Common Lisp the Language, 2nd ed. Digital Press, 1990. 1029 pgs.
6. R. Kelsey, W. Clinger, J. Rees (eds.), Revised5 Report on the Algorithmic Language Scheme, Higher-Order and Symbolic Computation, Vol. 11, No. 1, August, 1998

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

1. <https://common-lisp.net>
2. <http://www.schemers.org>
3. <https://call-cc.org/>
4. <http://www.swi-prolog.org/>

5. <http://www.refal.net> 6. <http://hopelog.com>

10. Методические указания для обучающихся по освоению дисциплины.

1) При проведении аттестации студентов важно всегда помнить, что систематичность, объективность, аргументированность - главные принципы, на которых основаны контроль и оценка знаний студентов. Проверка, контроль и оценка знаний студента, требуют учета его индивидуального стиля в осуществлении учебной деятельности. Знание критериев оценки знаний обязательно для преподавателя и студента. Помимо выполнения заданий на лабораторных занятиях рекомендуется самостоятельно решить упражнения, предложенных к каждой лекции.

2) Планирование времени на самостоятельную работу, необходимого на изучение настоящей дисциплины, студентам лучше всего осуществлять на весь семестр, предусматривая при этом регулярное повторение пройденного материала. Самостоятельная работа студентов заключается в решении всех разобранных на занятиях упражнений, материала учебника и соответствующих форумов интернет, решения всех заданий из индивидуальных практических заданий, решения рекомендуемых задач, подготовки к сдаче промежуточных отчетов и экзамена и дополнительной работы в компьютерном классе самостоятельно. 11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем Для проведения полноценных занятий необходимо следующее программное обеспечение: Операционная система Windows 7, 8.1 и 10, JDK, Microsoft Visual Studio Express, NetBeans, Ubuntu Linux.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

В процессе обучения для самостоятельной работы студентов используются операционные системы семейства Unix (Linux или FreeBSD), интерпретатор SBCL, компилятор Chicken Scheme, система SWI-Prolog, интерпретатор Hopeless. Студентам для самостоятельной работы предоставляется удалённый доступ к системе типа Unix, в которой установлены вышеперечисленные инструментальные средства.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Для преподавания дисциплины требуется лекционная аудитория, оборудованная маркерной или меловой доской, а также компьютерный класс. Компьютеры должны работать под управлением ОС Linux или FreeBSD, в системе должны быть установлены интерпретатор SBCL, компилятор Chicken Scheme, система SWI-Prolog, интерпретатор Hopeless. Необходимо наличие доступа к сети «Интернет».