

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
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
02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) программы:
Информатика и компьютерные науки

Уровень высшего образования
бакалавриат

Форма обучения
очная


Статус дисциплины:
входит в часть ОПОП формируемую участниками образовательных
отношений в форме факультатива

Рабочая программа дисциплины “Избранные вопросы языков программирования” составлена в 2021 году в соответствии с требованиями ФГОС ВО – бакалавриат по направлению подготовки 02.03.02 – Фундаментальная информатика и информационные технологии от «23» августа 2017 №808.


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

Рабочая программа дисциплины одобрена:

на заседании кафедры дискретной математики и информатики от «30» мая 2021 г., протокол № 9.

Зав. кафедрой  Магомедов А.М.
(подпись)

на заседании Методической комиссии ФМиКН от «23» июня 2021г., протокол №6.

Председатель  В.Д.Бейбалаев
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением «9» июля 2021 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	КСР	Консультации		
6	36	34	34					2	Зачет
Итого	36	34	34					2	

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 акультатива 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
ПК-2. Способность понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии	ПК-2.1. Знает основные методы решения прикладных задач, современные методы информационных технологий.	Знает: современные математические методы, применяющиеся для решения задач в области естественных наук, экономики, социологии и информационно коммуникационных технологий. Умеет: применять современные методы постановки и анализа задач в области математики и информатики. Владеет: навыками оптимального выбора современных методов и средств постановки и анализа задач в области математики и информатики	Устный опрос, письменный опрос;
	ПК-2.2. Умеет корректно оформить результаты научного труда в соответствии с современными требованиями.	Знает: современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения. Умеет: применять современные методы разработки и реализации	

		алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения. Владеет: навыками оптимального выбора современных методов разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	
	ПК-2.3. Имеет практический опыт использования сети Интернет, аннотирования, реферирования, библиографического разыскания и описания, опыт работы с научными источниками.	Знает: современные методы реализации различных математических алгоритмов в виде программных комплексов, особенности современных вычислительных комплексов. Умеет: применять современные методы реализации различных математических алгоритмов в виде программных комплексов с учетом особенностей современных вычислительных комплексов Владеет: навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	
ПК-4. Способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных,	ПК-4.1. Знает современные языки программирования и методы параллельной обработки данных. Знаком с содержанием Единого Реестра Российских программ для	Знает: базовые основы современного математического аппарата, связанного с проектированием программных продуктов и программных комплексов. Умеет: использовать математический	Устный опрос, письменный опрос;

<p>операционные системы, электронные библиотеки и пакеты программ, сетевые технологии.</p>	<p>электронных вычислительных машин и баз данных.</p>	<p>аппарат профессиональной деятельности. Владеет: практическим опытом применения современного математического аппарата, связанного с проектированием и разработкой программных продуктов и программных комплексов.</p>	<p>В</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: базовые понятия в области математических наук и программирования. Умеет: находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности. Владеет: практическим опытом научно-исследовательской деятельности в математике и информатике</p>	
	<p>ПК-4.3. Имеет практический опыт разработки интеграции информационных систем.</p>	<p>Знает: этапы подготовки программ, подробную структуру программы, простые и структурированные данные, управляющие структуры Умеет: составлять программы средней сложности, воплощать в исполняемые приложения простые базовые алгоритмы Владеет: навыками компиляции, отладки и тестирования программ</p>	

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		4				Устный опрос
2	Язык Lisp и примеры его использования.		4				Устный опрос
3	Язык Scheme.		4				Устный опрос
4	Язык Prolog, логическое и декларативное программирование.		4			2	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		6				Устный опрос
	<i>Итого за модуль 1</i>	36	34			2	Устный опрос
	ИТОГО:	36	34			2	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, т очечные 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 простой 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 отсутствия 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 Nopeless. 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	1		
Подготовка к опросу на практических занятиях			
Решение задач и упражнений			
Подготовка к коллоквиуму и контрольным работам	1		
Поиск материала на интернет-форумах			
Подготовка к экзамену			
Итого СРС:	2		

6.2. Порядок контроля:

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

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.

8. Учебно-методическое обеспечение дисциплины.

а) адрес сайта курса <http://cathedra.dgu.ru/EducationalProcess.aspx?Value=18&id=6>

1. <http://www.isi.edu/nsnam/ns/>
2. Литература по Simulink [Электронный ресурс].
3. <http://matlab.exponenta.ru/simulink/default.php>
4. www.intuit.ru.

б) основная литература:

1. Э. Хювёнен, 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.

с) 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».