

МИНОБРНАУКИ РОССИИ



Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГБОУ ВО «РГГУ»)

ОТДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ГУМАНИТАРНОЙ СФЕРЕ
*Учебно-научный центр программного и лингвистического обеспечения
интеллектуальных систем*

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

45.03.04 Интеллектуальные системы в гуманитарной сфере

Разработка и программирование интеллектуальных систем в гуманитарной сфере

Уровень высшего образования: бакалавриат

Форма обучения очная

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2022

Функциональное программирование
Рабочая программа дисциплины

Составитель:
доцент
М.Е. Епифанов

УТВЕРЖДЕНО
Протокол заседания кафедры МЛиИС
№ 5 от 24.03.2022

ОГЛАВЛЕНИЕ

1. Пояснительная записка

1.1 Цели и задачи дисциплины

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения дисциплине

1.3. Место дисциплины в структуре образовательной программы

2. Структура дисциплины

3. Содержание дисциплины

4. Образовательные технологии

5. Оценка планируемых результатов обучения

5.1. Система оценивания

5.2. Критерии выставления оценок

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Список источников и литературы

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

7. Материально-техническое обеспечение дисциплины

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья

9. Методические материалы

9.1. Планы практических (семинарских, лабораторных) занятий

9.2. Методические указания для обучающихся по освоению дисциплины

Приложения

Приложение 1. Аннотация дисциплины

Приложение 2. Лист изменений

1. Пояснительная записка

1.1. Цели и задачи дисциплины

Цели дисциплины:

- повысить квалификацию, расширить эрудицию студента, как программиста, показав ему методы и средства функционального программирования;
- способствовать формированию у студента навыков работы самостоятельного программиста.

Задачи дисциплины:

- приобретение студентами знаний, требуемых для правильного понимания роли и места функционального программирования в процессе разработки прикладных программных систем (в том числе интеллектуальных);
- формирование у студентов элементов логического и алгоритмического мышления, способствующих, в случаях, когда это полезно, построению алгоритмов решения ориентированных на программную реализацию задач в виде композиции функций с использованием ленивых (отложенных) вычислений и рекурсии;
- выработка у студентов навыков практического применения функциональных языков программирования.

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения дисциплине:

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
<p>ПК-3 Способен разрабатывать и тестировать новые программы и интерфейсы систем</p> <p>ПК-4 Способен разрабатывать, модернизировать и применять системы, использующие средства баз данных и лингвистического обеспечения</p>	<p>ПК-3.1 Знает современные парадигмы программирования, способы описания формальных языков</p> <p>ПК-4.1 Знает теоретические основы разработки баз данных и систему требований, предъявляемых к лингвистическому обеспечению</p> <p>ПК-4.2 Умеет применять современные системы управления базами данных для практической работы по созданию и использованию баз данных в разных предметных областях</p> <p>ПК-4.3 Умеет использовать лингвистическое обеспечение информационных систем</p> <p>ПК-4.4 Имеет практический опыт разработки, модернизации и использования баз данных, а также использования лингвистического обеспечения информационных систем</p>	<p><i>Знать.</i> Студенты должны овладеть следующими основными понятиями: - интерпретатор, интерпретируемый язык программирования; - декларативное программирование, директивное программирование; - парадигма функционального программирования, функциональный стиль программирования, функциональная среда, функциональный язык программирования; - композиция функций, «чистое» функциональное программирование, рекурсивный алгоритм; - ленивое (отложенное) вычисление и «энергичное» вычисление, частичное вычисление; - лямбда исчисление, лямбда оператор, лямбда выражение. Методы объектно-ориентированного программирования в языке Common Lisp. Методы построения сложных многоссылочных сетевых структур. <i>Уметь:</i> разрабатывать алгоритмы решения задач в функциональном стиле, применяя, по необходимости, рекурсию; разрабатывать несложные программы на языке Common Lisp; применять возможности ленивых вычислений при реализации «открытых» систем. <i>Владеть:</i> навыками рекурсивного программирования.</p>
<p>ПК-3 Способен разрабатывать и тестировать новые программы и интерфейсы систем</p>	<p>ПК-3.2 Умеет использовать интегрированные среды разработки, включая средства визуального программирования, умеет использовать средства автоматизации этапов анализа и проектирования</p> <p>ПК-3.3 Имеет практический опыт разработки и</p>	<p><i>Знать:</i> основные принципы использования интегрированной среды разработки (IDE) Cogman Common Lisp. <i>Уметь:</i> эффективно использовать руководство пользователя в целях изучения языка и справочное руководство при реализации приложений. <i>Владеть</i> навыками: - анализа документации к программным</p>

	тестирования прикладных программ	системам; - самостоятельного специалиста в области программной реализации интеллектуальных систем.
--	----------------------------------	---

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Функциональное программирование» относится к формируемой участниками образовательных отношений части блока Б1 дисциплин учебного плана. Для освоения дисциплины желательно владение компетенциями, сформированными (в рамках бакалавриата) в ходе изучения следующих дисциплин: «Алгебра», «Математическая логика», «Программирование» (на С и С++ - 1 и 2 курсы соответственно), «Логическое программирование», «Структуры данных», «Методы объектно-ориентированного программирования».

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: «Интеллектуальные системы», «Концепции и технологии современного программирования», а также связанные с программированием дисциплины, изучаемые в магистратуре ОИС в ГС.

2. Структура дисциплины

Структура дисциплины для очной формы обучения

Общая трудоёмкость дисциплины составляет 6 з.е., 216 ч., в том числе контактная работа обучающихся с преподавателем 84 ч., самостоятельная работа обучающихся 132 ч.

№ п/п	Раздел дисциплины/темы	Се ме стр	Виды учебной работы (в часах)					Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам)	
			Контактная				Пром ежут очная аттес тация		Са мос тоя тел ь- ная раб ота
			Лек ции	Сем ина р	Практ ическ ие заняти я	Лабор аторн ые заняти я			
1.	Теоретические основы функционального программирования. Лямбда-исчисление.	5				4	4	Опрос	
2.	Функциональные языки	5				2	2	Опрос	
3.	Введение в программирование на языке Common Lisp. Рекурсивное программирование, «чистый» Лисп	5				8	16	Оценка выполнения практических заданий, контрольная работа	
4.	Интегрированная среда разработки Corman Common Lisp	5				2	4		
5.	Язык Common Lisp. Общий обзор средств	5				26	46	Оценка выполнения	

								практических заданий
	Промежуточная аттестация	5						зачет
6.	Язык Common Lisp. Общий обзор средств (продолжение)	6			2		14	Оценка выполнения практических заданий
7.	CLOS (Common Lisp Object System)	6	6		4		10	Оценка выполнения практических заданий
8.	Практикум: сетевая объектная модель	6	14		16		36	Оценка совместной работы со студентом, как с исполнителем
	Итоговая аттестация	6						зачет с оценкой
	Итого		20		64		132	

Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Теоретические основы функционального программирования. Лямбдаисчисление.	Краткое изложение основных сведений о формальном обосновании функционального программирования: лямбда исчисление, лямбда оператор, лямбда выражение, редукции и нормальные формы.
2.	Функциональные языки	Интерпретатор, интерпретируемый язык программирования. Декларативное программирование, директивное программирование. Парадигма функционального программирования, функциональный стиль программирования, функциональная среда, функциональный язык программирования. Композиция функций, «чистое» функциональное программирование, рекурсивный алгоритм. Ленивое (отложенное) вычисление и «энергичное» вычисление, частичное вычисление
3.	Введение в программирование на языке Common Lisp. Рекурсивное программирование, «чистый» Лисп	Синтаксис: основной тип данных – списки; атомы, их типы; S-выражения, функции; - основные встроенные функции: -- «чистый Лисп» и интерпретатор (eval, apply, funcall).
4.	Интегрированная среда разработки Cogman Common Lisp	Контекстный редактор, рабочий лист консоли, особенности отладки
5.	Язык Common Lisp. Общий обзор средств	Функции. Символы. Функционалы (map- и другие функции). Управляющие конструкции: ветвления, циклы, блоки (функции и макросы их реализующие).
6.	Язык Common Lisp. Общий обзор средств (продолжение)	Обзор типов атомов. Макросы и их определение.
7.	CLOS (Common Lisp Object System)	Определения классов. Объекты. Множественное наследование. Методы и обобщенные функции. Системные функции для работы с объектами и определениями классов
8.	Практикум: сетевая объектная модель	Основные «сложные» динамические структуры данных (списки, динамические массивы, множества, стеки, очереди, деки, графы (в частности, деревья и сети) и возможные способы их реализации

		(процедурные языки, Common Lisp). Объектные модели. Реализация многосвязной сетевой объектной модели, функциональность которой включает поиск узла, редактирование сети и навигацию по сети (блуждание по ее узлам). Разработка входных языков описания такой модели. Сериализация и десериализация модели: сохранение состояния модели в файле. Программная реализация на языке Common Lisp.
--	--	--

4. Образовательные технологии

№ п/п	Наименование раздела	Виды учебных занятий	Образовательные технологии
1	2	3	4
1.	Теоретические основы функционального программирования. Лямбдаисчисление.	Лабораторные занятия 1-2 Самостоятельная работа	Теоретическая лекция. Семинар-обсуждение. Консультирование посредством электронной почты
2.	Функциональные языки	Лабораторное занятие 3 Самостоятельная работа	Теоретическая лекция. Семинар-обсуждение. Демонстрация (runtime) выполнения кодов. Консультирование посредством электронной почты
3.	Введение в программирование на языке Common Lisp. Рекурсивное программирование, «чистый» Лисп	Лабораторные занятия 4-7 Самостоятельная работа	Семинар-обсуждение. Демонстрация (runtime) выполнения кодов. Практикум по решению задач. Консультирование посредством электронной почты
4.	Интегрированная среда разработки Common Common Lisp	Лабораторное занятие 8 Самостоятельная работа	Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
5.	Язык Common Lisp. Общий обзор средств	Лабораторные занятия 9-21 Самостоятельная работа	Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
6.	Язык Common Lisp. Общий обзор средств (продолжение)	Лабораторное занятие 22 Самостоятельная работа	Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
7.	CLOS (Common Lisp Object System)	Лекция 1 Лабораторное занятие 23 Самостоятельная работа	Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
8.	Практикум: сетевая объектная модель	Лекции 2-6 Лабораторные занятия 24-29 Самостоятельная	Теоретические лекции. Спецификация учебного проекта. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты

		работа	
--	--	--------	--

5. Оценка планируемых результатов обучения

5.1. Система оценивания

Форма контроля	Срок отчетности	Макс. количество баллов	
		За одну работу	Всего
Текущий контроль:			
• Опрос (1—2)	3—16 недели	5 баллов	5 баллов
• контроль за программным выполнением практических заданий (темы 3—5)	2—16 недели		45 баллов
• контр. работа (тема 3)	12 неделя	20 баллов	10 баллов
Промежуточная аттестация (зачет)	17 неделя		40 баллов
Итого за семестр (дисциплину)			100 баллов
Текущий контроль:			
• контроль за программным выполнением практических заданий (темы 6,7)			20 баллов
• учебный проект (тема 8)	7—16 недели	40 баллов	40 баллов
Промежуточная аттестация (экзамен)	17 неделя		40 баллов
Итого за семестр (дисциплину)			100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала	Шкала ECTS
95 – 100	отлично	A
83 – 94		B
68 – 82	хорошо	C
56 – 67	удовлетворительно	
50 – 55		
20 – 49	неудовлетворительно	FX
0 – 19		не зачтено

5.2. Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	«отлично»/ «зачтено»	Выставляется обучающемуся, если он: - грамотно выполнил большинство (в том числе и все обязательные) практические задания (программные образцы – samples); - глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации; - исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения; - свободно ориентируется в учебной и профессиональной литературе.

		<p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ С	«хорошо»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> - грамотно, возможно с помощью преподавателя, выполнил достаточное количество практических заданий (программные образцы – samples), в том числе и все обязательные; - знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей; - правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами; - достаточно хорошо ориентируется в учебной и профессиональной литературе. <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	«удовлетворительно»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> - выполнил (возможно с помощью преподавателя) достаточное количество практических заданий (программные образцы – samples); - знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации; - испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами; - демонстрирует достаточный уровень знания учебной литературы по дисциплине. <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворительно»/ не зачтено	<p>Выставляется обучающемуся, если он не явился по неуважительной/неизвестной причине на аттестацию или:</p> <ul style="list-style-type: none"> - не выполнил (несмотря на возможную помощь преподавателя) достаточное количество практических заданий (программные образцы – samples); - не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. - испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами; - демонстрирует фрагментарные знания учебной литературы по дисциплине. <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции на уровне «достаточный», закреплённые за</p>

		дисциплиной, не сформированы.
--	--	-------------------------------

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

5.3.1. Образцы заданий для самостоятельного выполнения

Контрольная работа 1

Контрольная работа выполняется программно в IDE Corman Common Lisp в компьютерном классе.

В контрольной работе «Рекурсивное программирование на “чистом” Лиспе» к разделу 4 каждому студенту предлагается индивидуальный вариант, включающий две задачи – с рекурсией только по хвосту списка (более простой вариант рекурсии) и с рекурсией и по хвосту и по голове (т.е. список понимается как дерево, как список общего вида). Время выполнения ограничивается преподавателем в зависимости от сложности задач.

Пример. По заданному списку S построить другой, показывающий число элементов каждого его подписка на каждом уровне вложения:
 (a 23 ("fff" (b c d 7)) e (a d z) ((k) q w (1 m 77 dd b))) → (6
 (2 (4)) (3) (4 (1) (5)))
 (((a))) → (1 (1 (1 (1))))
 (((a)) b (d d d d (c c c (b (a) b)))) → (3 (1 (1 (1))) (4 (3 (2
 (1))))))
 (()) → (1 (0))

(рекурсия по хвосту и по голове списка)

На зачете в 5-м семестре предлагаются несколько простых задач (к разделу 5) на применение функционалов (map-функций и некоторых других), функций интерпретаторов (eval, apply и funcall), lambda-выражений, блоковых выражений, различных способов задания формальных параметров функций.

На итоговом зачете с оценкой в 6-м семестре проверяется успешность освоения студентами CLOS (Common Lisp Object System). Каждому учащемуся индивидуально предлагается всего одна задача, в которой, исходя из условия, надо определить некоторый класс и/или обобщенную функцию, класс и суперкласс с наследованием и т.п.

Пример.

Предварительные сведения.

Кватернионы — система гиперкомплексных чисел, образующая векторное пространство размерностью четыре над полем вещественных чисел.

Кватернионы можно определить как формальную сумму $a+bi+cj+dk$, где a, b, c, d — вещественные числа, а i, j, k — мнимые единицы со следующим свойством:

$i^2=j^2=k^2=ijk=-1$. Таким образом, таблица умножения базисных кватернионов — $1, i,$

j, k — выглядит так:

×	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Например, $ij=k$, а $ji=-k$.

Как вектор и скаляр, кватернион представляет собой пару (a, \mathbf{u}) , где \mathbf{u} — вектор трёхмерного пространства, а a — скаляр, то есть вещественное число.

Операция сложения определена следующим образом: $(a, \mathbf{u}) + (b, \mathbf{v}) = (a+b, \mathbf{u}+\mathbf{v})$

Произведение определяется следующим образом: $(a, \mathbf{u})(b, \mathbf{v}) = (ab - \mathbf{u} \bullet \mathbf{v}, a\mathbf{v} + b\mathbf{u} + \mathbf{u} \times \mathbf{v})$, где \bullet обозначает скалярное произведение, а \times — векторное произведение.

Задача.

Определить класс для представления кватернионов, метод to-string его «печати в строку» (для удобного вывода в протокол) в виде $a+bi+cj+dk$, «распространить» операции сложения и умножения на кватернионы.

6. Учебно-методическое и информационное обеспечение дисциплины

Замечание: многие источники полностью или в виде актуальных фрагментов имеются в электронной библиотеке (ЭБ) Учебно-методического кабинета (УМК) Отделения интеллектуальных систем (ОИС), папка [d:_ois_lib] (либо [c:_ois_lib] в случае единственного логического диска) на компьютерах деканата.

6.1. Список источников и литературы

а) Основная литература

1. Грэм, П. ANSI Common Lisp – Пер. с англ. – СПб.: Символ-Плюс, 2012. – 448 с. (ЭБ УМК ОИС: [d:_ois_lib\3fp\ANSI_Common_Lisp_(Rus).pdf])
2. Steele, Guy L., Common Lisp the Language, 2nd edition - Digital Press, 1990 – в электронном виде: <http://www-prod-gif.supelec.fr/docs/cltl/cltl2.html>
3. Филд А., Харрисон П. Функциональное программирование.: – М.: Мир, 1993

б) Дополнительная литература

1. Барендрегт Х. Лямбда-исчисление. Его синтаксис и семантика. – М.: Мир, 1985.
2. Роганова Н.А. Функциональное программирование: Учебное пособие для студентов высших учебных заведений – М.: ГИНФО, 2002.
3. Хювёнен Э., Сеппянен И. Мир Лиспа. (В 2-х томах) М.: Мир, 1990.
4. Душкин Р.В. Функциональное программирование на языке Haskell. – М.: ДМК Пресс, 2006
5. <http://www.haskell.org/> – сайт, посвящённый функциональному программированию в общем и языку Haskell в частности.
6. Уинстон П. Искусственный интеллект. – М.: Мир, 1980.
7. Элти Дж., Кумбс М., Экспертные системы: концепции и примеры. – М.: Финансы и статистика, 1987.

6.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

<http://www.cormanlisp.com/> – официальный сайт Corman Technologies Inc. (скачивание IDE Corman Common Lisp и технической документации к ней)

Перечень БД и ИСС

№п/п	Наименование
1	Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2019 г. Web of Science Scopus
2	Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2019 г. Журналы Cambridge University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis
3	Профессиональные полнотекстовые БД JSTOR Издания по общественным и гуманитарным наукам Электронная библиотека Grebennikon.ru
4	Компьютерные справочные правовые системы Консультант Плюс, Гарант

7. Материально-техническое обеспечение дисциплины

Занятия проводятся в компьютерных классах ауд. 307 и 706, расположенных по адресу 125993, Москва, Миусская пл., д. 6, стр.2,.

Этот компьютерный класс оснащен

- достаточным количеством объединенных в локальную сеть рабочих станций,
- медиапроектором и экраном,
- меловой доской.

В классе имеются возможности

- подключения ноутбука к медиапроектору,
- одновременного доступа в Интернет для преподавателя и студентов.

1. Перечень ПО

№п/п	Наименование ПО	Производитель	Способ распространения (лицензионное или свободно распространяемое)
1	Adobe Master Collection CS4	Adobe	лицензионное
2	ОС «Альт Образование» 8	ООО «Базальт СПО	лицензионное
3	Windows 10 Pro	Microsoft	лицензионное
4	Kaspersky Endpoint Security	Kaspersky	лицензионное
5	Microsoft Office 2016	Microsoft	лицензионное
6	IDE <u>Corman Common Lisp v. 3.1</u>	<u>Corman Technologies Inc.</u>	лицензионное

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
 - обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
 - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
 - письменные задания оформляются увеличенным шрифтом;
 - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.
- для глухих и слабослышащих:
 - лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;
 - письменные задания выполняются на компьютере в письменной форме;
 - экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.
- для лиц с нарушениями опорно-двигательного аппарата:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
 - в форме аудиофайла.
- для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;
 - в форме электронного документа;
 - в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих:
 - устройством для сканирования и чтения с камерой SARA CE;
 - дисплеем Брайля PAC Mate 20;
 - принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих:
 - автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
 - акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - передвижными, регулируемые эргономическими партами СИ-1;
 - компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1. Планы семинарских занятий

Лабораторные занятия по «блокам» схожих тем однотипны, они проводятся по единой схеме. На занятиях объясняются основные теоретические положения изучаемых тем, формулируются постановки для части задач, предлагаются методы, возможные сценарии

и технологические элементы для их решения. Часть задач объясняется непосредственно на практических занятиях.

Углубленно текущий материал изучается студентами самостоятельно.

Почти все практические задания также выполняются студентами в процессе самостоятельной работы.

На практических занятиях при необходимости проводится разбор текущего материала и контроль в форме опросов и дискуссий его усвоения.

Наконец, на практических занятиях осуществляется контроль за выполнением практических заданий, при необходимости проводятся индивидуальные или групповые консультации.

(Конкретнее эти положения изложены в п. 5.3.)

9.1. Пример планов лабораторных занятий

1. Изучение средств программирования (темы 1-6, 9-12).

Темаб (6ч.) CLOS (Common Lisp Object System)

Цель занятий: изучение встроенной в язык подсистемы ООП и программирование классов.

Форма проведения – опрос, дискуссии, решение задач.

Вопросы для обсуждения:

Определения классов. Объекты.

Множественное наследование.

Методы и обобщенные функции.

Системные функции для работы с объектами и определениями классов.

Примеры контрольных вопросов:

1. Парадигма объектно-ориентированного программирования, в том числе на примере CLOS.
2. Сравнение CLOS и ООП в C++, C#, Java языке сценариев JavaScript.
3. Применение CLOS для программирования динамических структур данных.

Задания:

см. в п.5.3 настоящего документа.

Список источников и литературы:

1. Грэм, П. ANSI Common Lisp – Пер. с англ. – СПб.: Символ-Плюс, 2012. – 448 с. (ЭБ УМК ОИС: [d:_ois_lib\3fp\ANSI_Common_Lisp_(Rus).pdf])
2. Steele, Guy L., Common Lisp the Language, 2nd edition - Digital Press, 1990 – в электронном виде: <http://www-prod-gif.supelec.fr/docs/cltl/cltl2.html>

АННОТАЦИЯ ДИСЦИПЛИНЫ

Дисциплина «Функциональное программирование» реализуется на Отделении интеллектуальных систем в гуманитарной сфере Учебно-научным центром программного и лингвистического обеспечения интеллектуальных систем в 5 и 6 семестрах.

Цели дисциплины:

- повысить квалификацию, расширить эрудицию студента, как программиста, показав ему методы и средства функционального программирования;
- способствовать формированию у студента навыков работы самостоятельного программиста.

Задачи дисциплины:

- приобретение студентами знаний, требуемых для правильного понимания роли и места функционального программирования в процессе разработки прикладных программных систем (в том числе интеллектуальных);
- формирование у студентов элементов логического и алгоритмического мышления, способствующих, в случаях, когда это полезно, построению алгоритмов решения ориентированных на программную реализацию задач в виде композиции функций с использованием ленивых (отложенных) вычислений и рекурсии;
- выработка у студентов навыков практического применения функциональных языков программирования.

Дисциплина направлена на формирование следующих компетенций:

ПК-3 - Способен разрабатывать и тестировать новые программы и интерфейсы систем.

ПК-4 - Способен разрабатывать, модернизировать и применять системы, использующие средства баз данных и лингвистического обеспечения.

В результате освоения дисциплины (*модуля*) обучающийся должен:

Знать.

- Студенты должны овладеть следующими основными понятиями:
 - интерпретатор, интерпретируемый язык программирования;
 - декларативное программирование, директивное программирование;
 - парадигма функционального программирования, функциональный стиль программирования, функциональная среда, функциональный язык программирования;
 - композиция функций, «чистое» функциональное программирование, рекурсивный алгоритм;
 - ленивое (отложенное) вычисление и «энергичное» вычисление, частичное вычисление;
 - лямбда исчисление, лямбда оператор, лямбда выражение.
- Методы объектно-ориентированного программирования в языке Common Lisp.
- Методы построения сложных многоссылочных сетевых структур.

Уметь:

- разрабатывать алгоритмы решения задач в функциональном стиле, применяя, по необходимости, рекурсию;
- разрабатывать несложные программы на языке Common Lisp;
- применять возможности ленивых вычислений при реализации «открытых» систем;
- эффективно использовать руководство пользователя в целях изучения языка и справочное руководство при реализации приложений.

Владеть навыками:

- рекурсивного программирования;
- анализа документации к программным системам;

- самостоятельного специалиста в области программной реализации интеллектуальных систем.

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц.

