

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Саратовский государственный технический университет имени Гагарина Ю.А.»

Кафедра «Информационная безопасность автоматизированных систем»

РАБОЧАЯ ПРОГРАММА

по дисциплине

Б. 1.2.9. «Функциональное и логическое программирование»

направления подготовки

09.03.01 «Информатика и вычислительная техника»

*Профиль «Программное обеспечение средств вычислительной техники и
автоматизированных систем»*

форма обучения – заочная

курс – 4

семестр – 8

зачетных единиц – 5

всего часов – 180

в том числе:

лекции – 8

лабораторные занятия – 16

самостоятельная работа – 156

контрольные работы – 1

экзамен – 8 семестр

1. Цели и задачи дисциплины

Цели освоения дисциплины «Функциональное и логическое программирование». Цель данного курса заключается в том, чтобы дать представление студенту о логической и функциональной парадигмах программирования, провести параллели с уже изученными парадигмами, объяснить основные плюсы и минусы данного подхода применительно к решению задач при разработке продуктов используемых в информационной безопасности.

Задачи изучения дисциплины:

- изучить логическое программирование;
- изучить основы языка Lisp;
- изучить функциональное программирование;
- изучить язык F#;
- изучить представление основных алгоритмов обработки данных на функциональных и логических языках.

2. Место дисциплины в структуре ООП ВО

Б.1.2 Вариативная часть

Для освоения дисциплины «Функциональное и логическое программирование» студенты используют знания, умения и виды деятельности, формируемые при изучении дисциплин «Информатика», «Программирование».

«Информатика» – знать формы и способы представления данных в персональном компьютере, классификацию современных компьютерных систем, типовые структуры и принципы организации компьютерных сетей; уметь применять типовые программные средства сервисного назначения (средства восстановления системы после сбоев, дефрагментации и очистки диска и т.п.), пользоваться сетевыми средствами и внешними носителями информации для обмена данными; владеть навыками обеспечения безопасности информации с помощью типовых программных средств, навыками поиска и обмена информацией в глобальной сети Интернет;

«Программирование» – знать основные парадигмы программирования: линейную, структурную, объектно-ориентированную. Знать основные приёмы программирования. Знать основы тестирования ПО. Знать принципы работы компиляторов и интерпретаторов. Пользоваться технической литературой для работы с платформой .Net.

Освоение дисциплины «Функциональное и логическое программирование» является необходимой для последующего изучения дисциплин:

1. для успешного прохождения итоговой государственной аттестации.

3. Требования к результатам освоения дисциплины:

Изучение дисциплины направлено на формирование следующих компетенций:

общефессиональных компетенций:

-способность осваивать методики использования программных средств для решения практических задач (ОПК-2)

профессиональных компетенций

- способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования (ПК-2)

Студент должен **знать:**

- основы языка Lisp;
- язык программирования F#;
- парадигмы функционального и логического программирования
- абстракцию и декомпозицию при функциональном подходе;
- подходы к параллельным вычислениям.

Студент должен **уметь:**

- устанавливать инструменты для разработки в основных операционных системах: Microsoft Windows, Linux, Mac OS X;
- работать с инструментами тестирования ПО NUnit;
- решать задачи обеспечения информационной безопасности, используя функциональную и логическую парадигмы программирования
- использовать функциональную парадигму программирования в языках C# V.6.0, C++ V11.

Студент должен **владеть:**

- навыками разработки на языке Visual Lisp;
- навыками разработки на языке Visual F#;
- навыками тестирования приложений с использованием NUnit;
- навыками работы параллельными системами;
- навыками работы с «ленивыми» вычислениями и асинхронными процессами.

В приложении 1 раскрыт процесс формирования компетенций и приведены критерии оценки знаний, умений и навыков

**4. Распределение трудоемкости (час.) дисциплины по темам
и видам занятий**

№ мо- ду- ля	№ неде- ли	№ те- мы	Наименование темы	Часы				
				Всего	Лек- ции	Лабора- торные	Прак- тичес- кие	СРС
1	2	3	4	5	6	7	8	9
1	1	1	Введение функциональное и логическое программирование языка Visual Lisp и Visual F#	36	2	4	-	30
1	2	2	Язык F#	30	2	4	-	24
1	3	3	Деревья.	42	2	4	-	36
1	4	4	Логические языки программирования.	72	2	4	-	66
Всего				180	8	16	-	156

5. Содержание лекционного курса

№ тем ы	Вс ег о ча со в	№ лекции	Тема лекции. Вопросы, отрабатываемые на лекции	Учебно- методическое обеспечение
1	2	3	4	5
1	2	1	Определение и краткая история функционального и логического программирования. Абстракция и декомпозиция. Декларативное программирование. Парадигмы программирования. Функциональное программирование в реальной жизни. Основные принципы функционального программирования. Введение в лямбда-исчисление. Visual Lisp. Редукция. Язык функционального программирования как лямбда-редуктор. Функции нескольких аргументов. Каррирование. Условное выражение. Определение имен. Области видимости. Сопоставление с образцом. Рекурсия. Циклы	[1,2,3] ИОС

2	2	2	<p>Язык F# Рекурсивные структуры данных. Списки Рекурсивные структуры данных. Основные операции работы со списками. Примеры работы со списками. Перестановки. Вычисление простых чисел. Работа с изображениями. Синтаксис порождения списка list comprehension. Хвостовая рекурсия. Порядковое представление списков и матриц. Функциональные структуры данных. Представление очереди. Многомерные массивы.</p>	[2,5,7] ИОС
3	2	3	<p>Деревья. Деревья общего вида и двоичные деревья. Обход дерева. Реализация обхода с помощью функции с отложенным вычислением. Деревья поиска. Деревья выражений. Хвостовая рекурсия для деревьев. Продолжения (continuations). Основные модели вычислений. Синтаксис л-исчисления. Чистое и прикладное л-исчисление. Преобразования л-выражений. Редукция. Бетта-редукция и замена переменной. Нормальный и аппликативный порядок редукции. Ленивые и энергичные вычисления. Механизмы вызова и проблема разделения. Теорема Чёрча-Россера и теорема стандартизации. Экстенциональность. Слабая заголовочная нормальная форма. Описание рекурсивных функций. Оператор неподвижной точки. Комбинаторы и комбинаторная логика.</p>	[2,3] ИОС
4	2	4	<p>Логические языки программирования. Представление условных выражений, списков и натуральных чисел в лямбда исчислении. Вычислимость. Эквивалентность алгоритмических моделей. Переопределение имен. Замыкания. Генераторы - как способ работы с бесконечными последовательностями, отложенные вычисления. F# sequences. Примеры. Ленивые вычисления. Мемоизация. Реализация машины Тьюринга. Определение, пример в Visual Lisp. Реализация на F#. Зиппер. Классификация языков программирования по видам типизации.</p>	[2,5,7] ИОС

			Типизированное лямбда исчисление. Вывод типов. Доказательство корректности программ на примерах. Проблема самоприменимости. Метапрограммирование. Quotations. Примеры. DLinq технология. Императивное ядро в функциональных языках. Монады.

6. Содержание коллоквиумов

№ темы	Всего часов	№ занятия	Тема практического занятия. Вопросы, отрабатываемые на практическом занятии	Учебно-методическое обеспечение
1	2	3	4	5
Учебным планом не предусмотрены				

7. Перечень практических занятий

№ темы	Всего часов	№ занятия	Тема практического занятия. Вопросы, отрабатываемые на практическом занятии	Учебно-методическое обеспечение
1	2	3	4	5
Учебным планом не предусмотрены				

8. Перечень лабораторных работ

№ темы	Всего часов	Наименование лабораторной работы. Вопросы, отрабатываемые на лабораторном занятии	Учебно-методическое обеспечение
1	2	4	5
1	4	Многомерные массивы. Реализация Сортировки	ИОС
2	4	Построение графика 2D-функции и построение графика трехмерной функции	
3	4	Порядковое представление списков и матриц. пример	
4	4	Циклические конструкции. Виды рекурсии. Монады	

9. Задания для самостоятельной работы студентов

Методические указания по самостоятельному изучению отдельных разделов дисциплины приведены в соответствующем разделе ИОС [14]

№ темы	Всего часов	Вопросы для самостоятельного изучения (задания)	Учебно-методическое обеспечение
1	6	Пример факториала F#.	[1,2,3,4,5,6,7,8,9,10,11,12,13,14] ИОС
1	6	Сортировка Хоара.	
1	6	Pattern Matching пример.	
1	6	Вывод данных в графический файл .	
1	6	Генерация перестановок.	
2	6	Типы .NET Framework.	
2	6	Разреженные матрицы. Транспонирование.	
2	6	Функциональные структуры данных.	
2	6	Конкатенация списков.	
3	6	Отличия деревьев от «общего» вида	
3	6	Разбор выражения в дерево.	
3	6	Разбор префиксной формы в дерево	
3	6	Проблема рекурсии.	
3	6	Комбинаторная логика (Логика комбинаторов)	
3	6	Числа ДеБрейна.	
4	6	Натуральные числа. Списковый подход	
4	6	Типизированное лямбда исчисление.	
4	6	Пределы верификации.	
4	6	Сборка мусора в Lisp и F#.	
4	6	Общий подход к разбору искусственных языков.	
4	6	Метапрограммирование.	
4	6	Работа с Raw Quotations.	
4	6	Монады ввода-вывода.	
4	6	Монады недетерминированных вычислений.	
4	6	Асинхронные и параллельные вычисления.	
4	6	Asychrnous Workflows.	

10. Расчетно-графическая работа

Расчетно-графическая работа учебным планом не предусмотрена.

11. Курсовая работа

Учебным планом не предусмотрена.

12. Курсовой проект

Учебным планом не предусмотрен

13. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

В ходе учебного курса реализуются компетенции ОК–2,ПК–2

Ход процесса формирования описан в приложении 1

Для реализации компетентного подхода в соответствии с требованиями ФГОС ВО в рамках учебного курса предусмотрены активные и интерактивные формы проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

В связи с этим предусмотрено применение мультимедийных средств и презентаций, обсуждение докладов студентов, лекции с элементами деловых игр, тестирование, консультации, решение ситуационных задач, дискуссии

Вопросы для зачета

Учебным планом не предусмотрен.

Вопросы для экзамена

1. Абстракция и декомпозиция.
2. Декларативное программирование.
3. Парадигмы программирования.
4. Функциональное программирование в реальной жизни. Основные принципы функционального программирования.
5. Лямбда-исчисление. Common Lisp.
6. Пример факториала F#
7. Сортировка Хоара.
8. Pattern Matching.
9. Вывод данных в графический файл в F# .
10. Генерация перестановок
11. Редукция. Язык функционального программирования как лямбда-редуктор. Функции нескольких аргументов.

12. Каррирование.
13. Условное выражение.
14. Определение имен. Области видимости.
15. Сопоставление с образцом.
16. Рекурсия. Циклы в F#
17. Деревья общего вида и двоичные деревья. Обход дерева. Реализация обхода с помощью функции с отложенным вычислением.
18. Деревья поиска. Деревья выражений. Хвостовая рекурсия для деревьев. Продолжения (continuations).
19. Основные модели вычислений. Синтаксис л-исчисления.
20. Чистое и прикладное л-исчисление. Преобразования л-выражений.
21. Редукция. Бетта-редукция и замена переменной.
22. Нормальный и аппликативный порядок редукции.
23. Ленивые и энергичные вычисления.
24. Механизмы вызова и проблема разделения.
25. Теорема Чёрча-Россера и теорема стандартизации.
26. Экстенциональность.
27. Слабая заголовочная нормальная форма.
28. Описание рекурсивных функций.
29. Оператор неподвижной точки.
30. Комбинаторы и комбинаторная логика. Рекурсивные структуры данных. Списки
31. Рекурсивные структуры данных. Основные операции работы со списками. Примеры работы со списками. Перестановки.
32. Вычисление простых чисел.
33. Синтаксис порождения списка list comprehension.
34. Хвостовая рекурсия. Порядковое представление списков и матриц.
35. Представление очереди. Функциональные структуры данных. Многомерные массивы.
36. Логические языки программирования.
37. Представление условных выражений, списков и натуральных чисел в лямбда исчислении. Вычислимость.
38. Эквивалентность алгоритмических моделей. Переопределение имен. Замыкания.
39. Генераторы - как способ работы с бесконечными последовательностями, отложенные вычисления. F# sequences. Примеры.
40. Ленивые вычисления. Мемоизация.
41. Реализация машины Тьюринга. Определение.
42. Реализация на F#. Зиппер.
43. Классификация языков программирования по видам типизации. Типизированное лямбда исчисление. Вывод типов.

44. Доказательство корректности программ на примерах. Проблема самоприменимости.
45. Метапрограммирование. Quotations.
46. DLink технология.
47. Императивное ядро в функциональных языках.
48. Монады. Виды назначения.

Тестовые задания по дисциплине

Для проведения тестирования используются тестовые материалы, разработанные в среде «Система тестирования знаний AST-Test версия 3»

Ниже представлен один из вариантов тестирования

1. Какой принцип построения функциональных программ?

- 1 программа представляет собой одно большое арифметическое выражение
- 2 программа строится из набора функций, каждая из которых перерабатывает входные данные в выходные. Существует четкое разделение между данными и функциями
- 3 программа строится из набора функций, каждая из которых перерабатывает входные данные в выходные. Функции также могут рассматриваться как данные
- 4 программа строится из набора вызывающих друг друга подпрограмм (процедур и функций)

2. В чем отличия функционального программирования и императивного?

- 1 в функциональном программировании происходит автоматический поиск решения задачи по ее декларативному описанию
- 2 функциональное программирование оперирует функциями и их применением к данным, императивное – операторами и тем, как они изменяют состояние памяти
- 3 все вышеперечисленное
- 4 в функциональном программировании каждая функция может оперировать только с той областью памяти, которая для нее выделена

3. Почему функциональные программы не содержат побочных эффектов?

- 1 отсутствует понятие переменной и оператора присваивания
- 2 запрещено модифицировать внутренние переменные функции извне самой функции

- 3 функция может оперировать только над переменными, описанными внутри нее
- 4 отсутствует понятие области видимости

5. Сортировка Хоара это...

- 1. Сортировка типа Bubble
- 2. qsort
- 3. Odd Even

6. Рекурсивные структуры данных. Это..

- 1. Типовые данные
- 2. Сомовызывающиеся данные
- 3. Целочисленные данные

7. Перестановки в F# осуществляются

- 1. Reverse
- 2. Reserve
- 3. Repeat

8. Вычисление простых чисел осуществляется

- 1. Лямбда способом
- 2. Методом делений
- 3. Методом Герона

9. Синтаксис порождения списка list comprehension определяется

- 1. Лямбда списком
- 2. Ультра списком
- 3. Конечным методом

10. Хвостовая рекурсия.

- 1. Вызов метода от «наследника»
- 2. Вызов метода от Лямбда-группы
- 3. Сомовывоз

14. Образовательные технологии

Для реализации компетентного подхода в соответствии с требованиями ФГОС ВО в рамках учебного курса предусмотрены активные и интерактивные формы проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

В связи с этим предусмотрено применение мультимедийных средств и презентаций, обсуждение докладов студентов, лекции с элементами деловых игр, тестирование, консультации, решение ситуационных задач, дискуссии.

15. Перечень учебно-методического обеспечения для обучающихся по дисциплине

ОСНОВНАЯ ЛИТЕРАТУРА

1. Курячий Г.В. Операционная система Unix [Электронный ресурс]: курс лекций. Учебное пособие / Курячий Г.В. – Москва Интернет-Университет Информационных Технологий (ИНТУИТ), 2004. – 288 с.

Режим доступа <http://www.iprbookshop.ru/22419>

2. Астахова И.Ф. Компьютерные науки. Деревья, операционные системы, сети [Электронный ресурс] учебное пособие / Астахова И.Ф. – Москва: ФИЗТМАЛИТ, 2013. – 88 с.

Режим доступа <http://www.iprbookshop.ru/24489>

3. Сергиевский, Г. М. Функциональное и логическое программирование [Текст] : учеб. пособие / Г. М. Сергиевский, Н. Г. Волченков. - М. : ИЦ "Академия", 2010. - 320 с. : ил. ; 22 см. - (Высшее профессиональное образование). - Библиогр.: с. 313-314. - Гриф: допущено Умо вузов по университет. политехн. образованию в качестве учеб. пособия для студ. вузов, обуч. по напр. "Информатика и вычислительная техника".

Экземпляры всего: 21

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

4. Сеницын, С. В. Операционные системы : учебник / С. В. Сеницын, А. В. Батаев, Н. Ю. Налютин. - М. : ИЦ "Академия", 2010. - 304 с. : ил. ; 22 см. - (Высшее профессиональное образование). - Библиогр.: с. 295 (15 назв.). - Гриф: рек. Умо по образованию в обл. приклад. информатики в качестве учеб. для студентов вузов, обучающихся по спец. "Приклад. информатика (по областям)" и др. экон. спец. - Имеется электронный аналог печатного издания.

Экземпляры всего: 20

Операционные системы [Электронный ресурс] / С. В. Сеницын, А. В. Батаев, Н. Ю. Налютин. - Электрон. текстовые дан. - М. : ИЦ "Академия", 2010. - 1 эл. опт. диск

(CD-ROM). - (Высшее профессиональное образование). - Систем. требования: Pentium II, 128 Мб ОЗУ, Windows 98/2000/ME/XP/Vista/7, CD/DVD ROM, Adobe Acrobat Reader. - Загл. с контейнера. - Гриф: рек. УМО по образованию в обл. приклад. информатики в качестве учеб. для студентов вузов, обучающихся по спец. "Приклад. информатика (по областям)" и др. экон. спец. - Электронный аналог печатного издания. - Диск помещен в контейнер 14X19 см.
Режим доступа: http://lib.sstu.ru/books/Ld_147.pdf.

5. Ложников П.С. Обеспечение безопасности сетевой инфраструктуры на основе операционных систем Microsoft[текст]: практикум/Ложников П.С. – Москва : БИНОМ Лаборатория знаний. Интернет-Университет Информационных Технологий (ИНТУИТ), 2007. – 264 с.
Режим доступа : <http://www.iprbookshop.ru/22415>

6. Пластун И.Л. Операционные системы : учеб. пособие для студ. спец. 220400, 071900 / И. Л. Пластун ; Сарат. гос. техн. ун-т (Саратов) . - Саратов : СГТУ, 2006. - 80 с. : ил. ; 21 см. - Библиогр.: с. 78 (12 назв.).
Экземпляры всего: 39

7. Симонович С. В. Информатика : базовый курс : учеб. пособие / под ред. С. В. Симоновича. - 2-е изд. - СПб. [и др.] : Питер, 2010. - 640 с. : ил. ; 24см. - (Учебник для вузов). - Гриф: рек. М-вом образования РФ в качестве учеб. пособия для студ. вузов.
Экземпляры всего: 51

8. Сырецкий, Г. А. Информатика. Фундаментальный курс [Текст] : учебник / Г. А. Сырецкий. Основы информационной и вычислительной техники. - 2005. - 832 с.
Экземпляры всего: 10

ПЕРИОДИЧЕСКИЕ ИЗДАНИЯ

9. Вестник Саратовского государственного технического университета [Текст]. : науч.-техн. журн. / Саратов. гос. техн. ун-т (Саратов); гл. ред. И. Р. Плеве. - Саратов : СГТУ. - Саратов : СГТУ, 2003. - . - Выходит ежеквартально. - ISSN 1999-8341

Режим доступа: http://elibrary.ru/title_about.asp?id=9567

10. Вычислительные методы и программирование: Новые вычислительные технологии. – ISSN: 1726-3522

Режим доступа: http://elibrary.ru/title_about.asp?id=2772

11. Информационно-технологический вестник. – ISSN: 2409-1650

Режим доступа: http://elibrary.ru/title_about.asp?id=53225

12. Проблемы информатики. – ISSN: 2073-0667

Режим доступа: http://elibrary.ru/title_about.asp?id=30275

ИНТЕРНЕТ-РЕСУРСЫ

13. Сеть разработчиков Microsoft. Режим доступа
<https://msdn.microsoft.com/ru-ru>. Дата обращения 20.02.2016

ИСТОЧНИКИ ИОС

14. <https://portal3.sstu.ru/Facult/INETM>

16. Материально-техническое обеспечение дисциплины.

Для проведения лекционных занятий требуется типовая лекционная аудитория, требуется комплект технических средств обучения в составе:

- персональный компьютер (в конфигурации не хуже: процессор Intel Core 2 Duo, 2 Гбайта ОЗУ, 500 Гбайт НЖМД);
- проектор (разрешение не менее 1280x1024);
- экран для проектора.

Для проведения лабораторных занятий требуется компьютерный класс, оборудованный ПЭВМ в конфигурации не худшей чем: процессор Pentium IV 3 ГГц, ОЗУ 2 Гбайта, НЖМД 200 Гбайт с установленными операционными системами семейств Microsoft Windows 7/ Linux.

Для проведения лабораторных занятий требуется компьютерный класс, оборудованный ПЭВМ в конфигурации не худшей чем: процессор Pentium IV 3 ГГц, ОЗУ 2 Гбайта, НЖМД 200 Гбайт с установленными операционными системами семейств Microsoft Windows 7, с установленной IDE Microft Visual Studio Express.

Для проведения тестирования по дисциплине используются технические средства в составе:

– персональный компьютер (в конфигурации не хуже: процессор Intel Core 2 Duo, 2 Гбайта ОЗУ, 500 Гбайт НЖМД) с установленным ПО «Система тестирования знаний AST-Test версия 3».

17. Особенности освоения для инвалидов и лиц с ограниченными возможностями здоровья

Для студентов с ограниченными возможностями здоровья предусмотрены следующие формы организации педагогического процесса и контроля знаний: *-для слабовидящих:*

обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения контрольных заданий при необходимости предоставляется увеличивающее устройство;

задания для выполнения, а также инструкция о порядке выполнения контрольных заданий оформляются увеличенным шрифтом (размер 16-20);

- для глухих и слабослышащих:

обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости студентам предоставляется звукоусиливающая аппаратура индивидуального пользования;

- для лиц с тяжелыми нарушениями речи, глухих, слабослышащих все контрольные задания по желанию студентов могут проводиться в письменной форме.

Основной формой организации педагогического процесса является интегрированное обучение инвалидов, т.е. все студенты обучаются в смешанных группах, имеют возможность постоянно общаться со сверстниками, легче адаптируются в социуме.

Приложение 1
Фонд оценочных средств текущего контроля и промежуточной аттестации

1.Карта компетенций

Контролируемые компетенции (шифр компетенции)	Планируемые результаты обучения (знает, умеет, владеет, имеет навык)
ОПК-2 способность осваивать методики использования программных средств для решения практических задач	Знает: парадигму функционального программирования
	Умеет: Использовать парадигму функционального программирования
	Владеет: навыками функционального программирования
ПК-2 способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования	Знает: Языки программирования F# и Lisp
	Умеет: Использовать язык F# для разработки компонент аппаратно-программных комплексов и баз данных.
	Владеет: навыками разработки, тестирования и отладки программ на языке F# и Lisp

2 КРИТЕРИИ ОЦЕНКИ ЭКЗАМЕНА

оценка «отлично»	знать: подходы к параллельным вычислениям. уметь: -использовать функциональную парадигму -программирования в языках C# V.6.0, C++ V11. -использовать язык F# для разработки компонент аппаратно-программных комплексов и баз данных владеть: -навыками работы параллельными системами; -навыками работы с «ленивыми» вычислениями и асинхронными процессами.
оценка «хорошо»	знать: язык программирования F#; уметь: -работать с инструментами тестирования ПО NUnit; -на продвинутом уровне использовать язык F# для разработки компонент аппаратно-программных комплексов и баз данных

	<p>владеть: навыками разработки на языке Common Lisp;</p>
<p>оценка «удовлет- ворительно»</p>	<p>знать: парадигмы функционального и логического программирования</p> <p>уметь: инсталлировать инструменты для разработки в основных операционных системах: Microsoft Windows, Linux, Mac OS X;</p> <p>владеть: базовыми навыками разработки на языке Common Lisp;</p>
<p>оценка «неудовлет- ворительно»</p>	<p>имеет фрагментарные представления о функциональном программировании</p>