**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Поволжский государственный университет телекоммуникаций и информатики»**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**«УТВЕРЖДАЮ»**

Декан факультета ИСТ

наименование факультета

Салмин А.А.

подпись Фамилия И.О.

« 28 » августа 2015 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ**

**Функциональное и логическое программирование (ФИЛП)**

наименование учебной дисциплины (полное, сокращенное)

|  |  |
| --- | --- |
| **Направление (специальность) подготовки** | 02.03.03 - Математическое обеспечение и  администрирование информационных систем |
|  | код и наименование направления (специальности) подготовки |
| **Профиль (специализация) подготовки** | Технология программирования |
|  | указывается при наличии |
| **Квалификация (степень) выпускника** | бакалавр |
|  | бакалавр, магистр, дипломированный специалист |
| **Факультет** | Информационных систем и технологий (ИСТ) |
|  | наименование факультета |
| **Кафедра** | Программное обеспечение и управление в технических системах (ПОУТС) |
|  | наименование кафедры |
| **Форма обучения** | очная |
|  | очная, заочная и т. п. |
|  |  |
| **Курс / семестр** | 3 / 6 |

**ФОС обсужден на заседании кафедры**  ПОУТС

Протокол № 1 от « 28 » 08 2015 г.

Заведующий кафедрой ПОУТС

наименование кафедры

Тарасов В.Н.

подпись Фамилия И.О.

« 28 » 08 2015 г.

**Самара**

**2015**

**Фонд оценочных средств дисциплины**

***«Функциональное и логическое программирование»***

***Ответственный исполнитель, уполномоченный по качеству кафедры ПОУТС***

доцент Мезенцева Е.М. \_\_\_\_\_\_\_\_\_

должность подпись фамилия, имя, отчество дата

***Исполнители:***

доцент \_\_\_\_\_\_\_\_\_\_\_ Стефанов М.А. \_\_\_\_\_\_\_\_\_

должность подпись фамилия, имя, отчество дата

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

должность подпись фамилия, имя, отчество дата

***Рецензент***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

должность уч. степень, уч. звание подпись фамилия, имя, отчество

« \_\_\_ » \_\_\_\_\_\_\_\_\_ 2015 г.

**1. Карта компетенций дисциплины**

***«Функциональное и логическое программирование»***

**Цель дисциплины:** получение базовых знаний и формирования навыков в области декларативных языков программирования.

**Задачи:** изучение списочных структур, лямбда-исчисления А.Черча, рекурсивных алгоритмов различных видов и порядков, функций с функциональным аргументом и функциональным значением, и применение указанных инструментов для решения задач на конкретных языках программирования.

В процессе освоения данной дисциплины студент формирует и демонстрирует следующие компетенции:

Общепрофессиональные компетенции:

ОПК-5: Владение информации о направлениях развития компьютеров с традиционной (нетрадиционной) архитектурой; о тенденциях развития функций и архитектур проблемно-ориентированных программных систем и комплексов

ОПК-7: Способность использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Перечень компонентов | Технологии формирования | Форма оценочного средства |
| *ОПК-5* | Знать принципы работы Lisp-машин  Уметь использовать рекурсивные алгоритмы  Владеть навыками работы с интерпретатором Lisp | Лекции, лабораторные работы, самостоятельная работа | Устный опрос, лабораторная работа,  зачет |
| Уровни освоения компетенции (*ОПК-5*) | | | |
| Пороговый уровень  Знать о существовании различных типов данных в Lisp  Уметь распознавать в коде рекурсивные функции  Владеть навыком запуска интерпретатора Lisp  Продвинутый уровень  Знать различные типы данных в Lisp  Уметь создавать простые рекурсивные функции  Владеть навыками установки и удаления интерпретатора Lisp  Высокий (превосходный) уровень  Знать все типы данных в Lisp  Уметь создавать рекурсивные функции 2 и более порядков  Владеть навыками создания функциональных программ с использованием интерпретатора Lisp | | | |
| *ОПК-7* | Знать основные концептуальные положения функционального и логического программирования  Уметь разрабатывать программы в рамках этих направлений  Владеть навыками программирования на декларативных языках | Лекции, лабораторные работы, самостоятельная работа | Устный опрос, лабораторная работа,  зачет |
| Уровни освоения компетенции (*ОПК-7*) | | | |
| Пороговый уровень  Знать нотацию лямбда-выражений и лямбда-вызовов  Уметь определять функции с использованием лямбда-выражений  Владеть навыками решения задач с использованием рекурсивных функций  Продвинутый уровень  Знать нотацию лямбда-выражений и лямбда-вызовов, рекурсивные алгоритмы, понятие функционалов  Уметь определять рекурсивные функции и функционалы  Владеть навыками решения задач с использованием рекурсивных функций и функционалов  Высокий (превосходный) уровень  Знать нотацию лямбда-выражений и лямбда-вызовов, рекурсивные алгоритмы, понятие функционалов, смысл макросов  Уметь определять рекурсивные функции, функционалы, макросы  Владеть навыками решения задач с использованием рекурсивных функций, функционалов, макросов | | | |

Профессиональные компетенции:

ПК-4: Способность к выбору архитектуры и комплексирования современных компьютеров, систем, комплексов и сетей системного администрирования

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Перечень компонентов | Технологии формирования | Форма оценочного средства |
| *ПК-4* | Знать представление основных типов данных Lisp в оперативной памяти  Уметь использовать функции для изменения структуры списков  Владеть навыками создания функционалов, замыканий и макросов | Лекции, лабораторные работы, самостоятельная работа | Устный опрос, лабораторная работа,  зачет |
| Уровни освоения компетенции (*ПК-4*) | | | |
| Пороговый уровень  Знать списочную структуру данных  Уметь работать с головой и хвостом списка  Владеть навыками определения функционалов  Продвинутый уровень  Знать структуры данных: список и точечная пара  Уметь соединять, разъединять списки  Владеть навыками определения функционалов и замыканий  Высокий (превосходный) уровень  Знать структуры данных: список и точечная пара, а так же способы хранения информации о свойствах символов  Уметь использовать структуроразрушающие функции  Владеть навыками определения функционалов, замыканий, макросов | | | |

**2. Паспорт фонда оценочных средств дисциплины**

**«***Функциональное и логическое программирование***»** очнаяформа обучения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Контролируемые разделы (темы) дисциплины\*** | **Код контролируемой компетенции\*\*** | **ФОС** | |
| **Форма оценочного средства** | **Комплект оценочных средств и кол-во вариантов заданий** |
| **1** | **2** | **3** | **4** | **5** |
| **1.** | Введение |  |  |  |
| **2.** | Основы языка ЛИСП | ОПК-5,7, ПК-4 | Опрос. Отчет по лабораторным работам | Тематическая структура банка контрольных вопросов |
| **3.** | Вычисления в ЛИСП | ОПК-5,7, ПК-4 | Опрос. Отчет по лабораторным работам | Тематическая структура банка контрольных вопросов |
| **4.** | Рекурсия | ОПК-5,7, ПК-4 | Опрос. Отчет по лабораторным работам | Тематическая структура банка контрольных вопросов |
| **5.** | Функциональное программирование | ОПК-5,7, ПК-4 | Опрос. Отчет по лабораторным работам | Тематическая структура банка контрольных вопросов |
| **6.** | Макросы | ОПК-5,7, ПК-4 | Опрос. Отчет по лабораторным работам | Тематическая структура банка контрольных вопросов |
|  | **Промежуточная аттестация**  *(зачет)* | ОПК-5,7, ПК-4 | **Задачи творческого уровня** | **Список задач к зачету** |

**3. Комплект оценочных средств дисциплины**

**«***Функциональное и логическое программирование***»** очнаяформа обучения

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование комплекта оценочных средств** | **№ приложения** |
| 1 | Тематическая структура банка контрольных вопросов | А |
| 2 | Список задач к зачету | Б |

**4. ЛИСТ согласования ФОС дисциплины на 2015/2016 учебный год**

**Направление подготовки:** 02.03.03 Математическое обеспечение и

администрирование информационных систем

шифр и наименование

**Профиль подготовки** Технология программирования

наименование

**Квалификация (степень)**

**выпускника:** Бакалавр

наименование

**Дисциплина:** Функциональное и логическое программирование

наименование

**Форма обучения:**  очная

(очная, очно-заочная, заочная)

**Учебный год 2015 / 2016**

**Рекомендована заседанием кафедры** ПОУТС

наименование кафедры

протокол № \_\_\_ от «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

Заведующий кафедрой ПОУТС

наименование кафедры

\_\_\_\_\_\_\_\_\_\_\_ Тарасов В.Н. \_\_\_\_\_\_\_\_\_

подпись расшифровка подписи дата

***Ответственный исполнитель, уполномоченный по качеству кафедры***

доцент Мезенцева Е.М. \_\_\_\_\_\_\_\_\_

должность подпись фамилия, имя, отчество дата

***Исполнители:***

Доцент Стефанов М.А.

должность подпись расшифровка подписи дата

***СОГЛАСОВАНО:***

Заведующий кафедрой ПОУТС

наименование кафедры

\_\_\_\_\_\_\_\_\_\_\_ Тарасов В.Н. \_\_\_\_\_\_\_\_\_

подпись расшифровка подписи дата

Декан ФЗО \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

подпись расшифровка подписи дата

Зам. декана по (направлению) специальности 02.03.03

шифр наименование

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ Чернова С.В. \_\_\_\_\_\_\_\_\_

должность подпись расшифровка подписи дата

**5. ЛИСТ изменений и дополнений в ФОС дисциплины**

**Дополнения и изменения в ФОС дисциплины на 20\_\_/20\_\_ уч.г.**

В ФОС дисциплины вносятся следующие изменения:

1) …………………………………..;

2) …………………………………..;

3) ……………………………………

или делается отметка о нецелесообразности внесения каких-либо изменений на данный учебный год

**ФОС дисциплины пересмотрен и одобрен на заседании кафедры \_\_\_\_\_\_\_\_\_**

Протокол № \_\_\_\_\_ от « \_\_\_ » \_\_\_\_\_\_\_\_\_ 201\_\_ г.

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

наименование кафедры

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

подпись расшифровка подписи дата

***СОГЛАСОВАНО:***

Декан ФЗО \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

подпись расшифровка подписи дата

Зам. декана по (направлению) специальности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

шифр наименование

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

должность подпись расшифровка подписи дата

**Внесенные изменения на 201\_\_/201\_\_ учебный год *УТВЕРЖДАЮ***

Декан факультета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

наименование факультета

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись расшифровка подписи

« \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_\_ г.

Приложение А

Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Поволжский государственный университет телекоммуникаций и информатики»

Кафедра ПОУТС

(наименование кафедры)

Тематическая структура банка контрольных вопросов по дисциплине «Функциональное и логическое программирование (ФИЛП)»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование темы  (раздела) | Контрольные вопросы по разделам дисциплины | Коды контролируемых  компетенций |
| **1** | Введение |  |  |
| **2** | Основы языка ЛИСП | 1. Какие объекты языка ЛИСП называются атомами.  2. Назовите правила составления списков.  3. Назовите базовые функции обработки списков.  4. Каковы отличия функций присваивания SET, SETQ, SETF.  5. Какой синтаксис записи лямбда-выражения, и в его чем отличие от лямбда-вызова. | ОПК-5,7, ПК-4 |
| **3** | Вычисления в ЛИСП | 1. В чем состоит отличие между статическими и динамическими переменными.  2. Назовите основные управляющие предложения языка ЛИСП.  3. Использование списочных ячеек памяти.  4. Как задаются и выводятся значения свойств символов.  5. Функции ввода и вывода. | ОПК-5,7, ПК-4 |
| **4** | Рекурсия | 1. Проблемы ресурсоемкости рекурсивных алгоритмов.  2. Назовите виды рекурсии и их особенности.  3. Проблемы рекурсии высоких порядков. | ОПК-5,7, ПК-4 |
| **5** | Функциональное программирование | 1. Определение функции с функциональным значением.  2. Способы композиции функций с функциональным значением.  3. Понятие авторепликативных и автоаппликативных функций.  4. Назовите основные применяющие и отображающие функционалы.  5. Определение замыкания.  6. Проблемы абстрактного подхода. | ОПК-5,7, ПК-4 |
| **6** | Макросы | 1. Чем отличается процесс вычисления макросов и функций (форм).  2. Назовите особенности рекурсивного макровызова.  3. Основные этапы тестирования макровызовов.  4. Назовите структуроразрушающие макросы. | ОПК-5,7, ПК-4 |

Коды контролируемых компетенций ОПК-5,7, ПК-4.

**Критерии оценки:** - оценка «***зачтено***» выставляется студенту, если он ответил на свыше 51% от количества вопросов;

- оценка «***не*** ***зачтено***» выставляется студенту, если он ответил на менее 50% от количества вопросов.

Составитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Стефанов

(подпись)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2015 г.

Приложение Б

Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Поволжский государственный университет телекоммуникаций и информатики»

Кафедра ПОУТС

Список задач к зачету по дисциплине

«Функциональное и логическое программирование (ФИЛП)»

1. Найти сумму двух чисел, используя только прибавление единицы в вычислениях.

(num\_sum 4 5).

9

2. Найти произведение двух чисел, используя только операцию сложения.

(num\_mult 3 4).

12

3. Найти сумму элементов заданного списка

(list\_sum (1 2 3 4)).

10

4. Найти НОД (наибольший общий делитель) двух натуральных чисел.

(Nod 20 10).

5

5. Написать предикат, определяющий является ли заданное натуральное число простым (делящимся только на 1 и само себя).

(is\_simple13).

T

6. Отсортировать элементы в числовом списке, используя рекурсию (например, методом простого выбора)

(Sort (4 2 1 3)).

(1 2 3 4)

7. Проверить, является ли заданный список "палиндромным" (симметричным)

(palind\_list (1 2 3 4 5 4 3 2 1))

T

8. Осуществить перевод числа из десятичной системы счисления в двоичную; результат представить в виде списка двоичных чисел, читаемых слева-направо

(binary 10).

(0 1 0 1)

9. Найти НОК (наименьшее общее кратное) двух натуральных чисел.

(nok 4 5).

20

10. Найти N-й член арифметической прогрессии, заданной первым элементом (a0) и коэффициентом разницы (d).

(arifm\_elem a0 N d).

(arifm\_elem 1 3 3).

10

11. Найти N-й член геометрической прогрессии, заданной первым элементом (b0) и знаменателем прогрессии (q).

(arifm\_elem b0 N q).

(arifm\_elem 1 3 3).

9

12. Найти максимальную длину непрерывной подпоследовательности из положительных значений

в числовом списке.

(max\_pos\_seq\_len (1 -2 -3 4 5 6 7 -7 0 8 9 -1 -2))

4

13. Удалить элементы-дубликаты из числового списка.

(del\_rep (1 2 5 2 3 4 4 5)).

(1 2 5 3 4)

14. Найти все отрицательные элементы в исходном числовом списке.

(pos\_list\_elem (1 -2 3 -4 -5 6))

(-2 -4 -5)

15. Обратить числовой список (переставить его элементы в обратном порядке)

(inverse\_list (1 2 3 4))

(4 3 2 1)

16. Опеределить, является ли заданное число степенью двойки и найти показатель степени.

?- deg\_of\_two(16, X).

X = 4

17. Сложить два числа, заданных в виде списка составляющих их цифр (1234+5678=5912)

(big\_sum (1 2 3 4) (5 6 7 8))

(6 9 1 2)

18. Определить, является ли заданное натуральное число полным квадратом другого натурального числа.

(nat\_square 12 144).

T

19. Найти 2^n по заданному n

(bin\_exp 4)

16

20. Определить, является ли заданный список, упорядоченным по возрастанию или убыванию (2 функции).

(po\_vozr (1 2 3)).

T

(po\_ubiv (3 2 3))

NIL

21. Разбить заданный список пополам, на списки с четными и нечетными порядковыми номерами.

(div\_list\_nechet (a b c d e)).

(a c e)

(div\_list\_chet (a b c d e)).

(b d)

(div\_list\_chet\_nechet (a b c d e)).

((a c e) (b d))

22. Реализовать слияние двух упорядоченных по возрастанию списков

(lists\_union (1 2 3) (2 3 4)).

(1 2 2 3 3 4)

23. Определить, что список не содержит одинаковых элементов (является множеством).

(is\_multitude (1 2 3 4)).

T

24. Вставить число в упорядоченный список не потеряв упорядоченность.

(list\_insert 2 (1 2 3 4)).

( 1 2 2 3 4)

25. Выяснить, является ли заданное множество подмножеством другого

(is\_sub\_mult (3 4) (1 2 3 4 5).

T

26. Найти количество минимальных элементов числового списка

(min\_count (-1 3 4 -1 6)).

27. Найти n-й положительный элемент списка

(find\_positive\_n (1 -2 3 -7 -8 4) 3)

4

28. Осуществить замену всех вхождений заданного элемента на следующее после него значение

(replace\_list\_elem (1 2 3 4 2 5 6 2) 2).

(1 3 3 4 5 5 6 nil)

29. Осуществить замену n-го вхождения заданного элемента на другое заданное значение

(replace\_nth\_elem (1 2 2 2 3 4) 2 3).

(1 2 2 3 3 4)

30. Разделить список на 3 части, в первый занести все элементы меньше заданного A, во второй равные заданному A и в третий больше заданного A.

(divide\_list (1 2 3 4 5 3 6) 3 X Y Z).

X ⬄ (1 2)

Y ⬄ (3 3)

Z ⬄ (5 6)

31. Подсчитать количество положительных (отрицательных) значений в списке чисел.

(count\_positive (1 -2 -3 4 5 0)).

3

(count\_negative (1 -2 -3 4 5 0)).

2

32. В числовом списке заменить все вхождения положительных значений на соседние отрицательные

элементы этого же списка, проходя список по порядку. Если отрицательных рядом нет, заменить на NIL.

(replace\_positive ( 1 -2 3 -4 -5 6 7 8)).

(-2 -2 -4 -4 -5 -5 nil nil)

33. Подсчитать количество четных значений в списке целых чисел.

(count\_chet ( 1 2 3 4 5)).

2

34. Удалить все четные значения в списке целых чисел.

(delete\_chet (1 2 3 4 5)).

(1 3 5)

35. Найти индекс первого отрицательного значения в списке чисел.

(find\_1st\_negative (1 2 -3 4 5).

3

36. Удалить все отрицательные значения в списке чисел.

(delete\_all\_negative (1 2 -3 4 5)).

(1 2 4 5)

37. Найти встречающееся наибольшее кол-во раз число в списке

(find\_popular (1 2 2 3 3 3 4).

3

38. Удалить первую непрерывную подпоследовательность из нечетных значений в списке чисел.

(delete\_first\_even\_seq (1 2 3 5 7 8 9 11)).

(1 2 8 9 11)

39. Заменить последнее вхождение положительного значения на 0 в списке чисел.

(replace\_last\_positive (1 -2 3 -4)).

(1 -2 0 -4)

40. Заменить все вхождения положительных значений на значение первого элемента в списке чисел.

(replace\_positive\_to\_first (-1 -2 -3 4 5)).

(-1 -2 -3 -1 -1)

41. Найти индекс первого вхождение отрицательного значения в списке чисел.

(find\_1st\_negative (1 2 3 -4 -5)).

4

42. Заменить первое вхождение отрицательного значения на максимальное значение в списке чисел.

(replace\_1st\_neg\_to\_max (1 2 -3 4 5)).

(1 2 5 4 5)

43. Найти сумму отрицательных значений в списке чисел.

(find\_neg\_sum (1 2 -3 -4)).

-7

44. Вернуть список индексов отрицательных значений в списке чисел.

(find\_neg\_indexes (1 2 -3 -4 0 -1)).

(3 4 6)

45. Вернуть индекс первого вхождения заданного значения в список.

(find\_first (1 4 2 3 2 4) 2).

3

46. Вернуть максимальное значение в списке чисел и индекс его первого вхождения.

(find\_max\_index (1 2 4 3 4)).

4

3

47. Удалить n-ый элемент из списка.

(del\_nth\_list\_elem (1 2 3 4) 2).

(1 3 4)

48. Удалить N элементов из списка, начиная с K-го.

(del\_list\_sequence (1 2 3 4 5 6 7) 3 4). (удаляем 3 элемента, начиная с 4й позиции)

(1 2 3 7)

49. Вставить новый элемент перед n-ым элементом в списке

(insert\_before\_nth (1 2 3 4) 5 4). (вставляем 5 перед 4м элементом списка)

(1 2 3 5 4)

50. Вставить новый подсписок перед K-ым элементом в списке

(insert\_sublist (1 2 3 4) 3 (5 6)).

(1 2 5 6 3 4)

51. Определить длину первой непрерывной подпоследовательности из значений в диапазоне от A до B в списке из целых чисел.

(find\_ranged\_seq\_len (1 2 3 4 5 6 7 8 3 4) 3 6).

4

52. Удалить все числа в диапазоне от A до B в списке из целых чисел.

(delete\_ranged\_elements (1 2 3 4 5 6 7 8 3 4) 3 6).

(1 2 7 8)

53. Найти индекс последнего вхождения значения равного A в списке из целых чисел.

(find\_last (1 2 3 4 5 4 4 6) 4).

7

54. Заменить n-е вхождение значения равного A на n в списке из целых чисел.

(replace\_nth\_element (1 2 3 4 5 4 4 6) 4 2).

(1 2 3 4 5 2 4 6)

55. Найти последовательную разность положительных значений в списке чисел.

(find\_positive\_diff (1 2 -3 -4 5 6)).

-12

56. Поменять местами первый отрицательный элемент и первый положительный элемент

(replace\_neg\_vs\_pos (1 2 -3 -4)).

(-3 2 1 -4)

57. Поменять местами два элемента списка с заданными номерами.

(replace\_a\_b (a b c d e f g) 2 5)

(a e c d b f g)

58. Осуществить циклический сдвиг элементов списка на один влево (вправо).

(shift\_left (1 2 3 4 5))

(2 3 4 5 1)

(shift\_right (1 2 3 4 5))

(5 1 2 3 4)

59. Осуществить циклический сдвиг элементов списка на заданное количество шагов влево (вправо).

(shift\_left (1 2 3 4 5) 3)

(4 5 1 2 3)

(shift\_right (1 2 3 4 5) 3)

(5 4 3 1 2)

60. Произвести поэлементное перемножение соответствующих элементов двух исходных списков.

(app (1 2 3 4) (3 4 5))

(3 8 15)

61. Вычислить скалярное произведение векторов, заданных списками целых чисел.

(sk\_app (1 2 3 4 5) (1 2 3 4 5))

55

62. Подсчитать число вхождений каждого элемента исходного списка. Ответом должен быть список пар, в которых первая компонента - элемент исходного списка, вторая - число его вхождений в первоначальный список.

(spis\_nth (1 2 2 4 2 1))

((1 2) (2 3) (4 1))

63. Cформировать список простых чисел, не превосходящих данного числа.

(spis\_simple 30)

(1 2 3 5 7 11 13 17 19 23 29)

64. Транспонировать матрицу, заданную списком списков.

(transpose ((1 2 3)

(4 5 6)

(7 8 9)))

((1 4 7)

(2 5 8)

(3 6 9))

Коды контролируемых компетенций ОПК-5,7, ПК-4.

**Критерии оценки знаний, умений и навыков**

Промежуточная аттестация осуществляется в форме зачета, который проводится по билетам, включающим 2 задачи. Оценка знаний студентов производится по следующим критериям:

оценка «*зачет*» выставляется студенту, если он предоставил корректное решение хотя бы одной задачи;

оценка «*не зачет*» выставляется студенту, если он не предоставил корректного решения ни одной из задач;

Составитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Стефанов

(подпись)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2015 г