

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
для промежуточной аттестации по дисциплине

Функциональное программирование

Код, направление подготовки	09.03.04 Программная инженерия
Направленность (профиль)	Программное обеспечение компьютерных систем
Форма обучения	Заочная
Кафедра разработчик	Автоматики и компьютерных систем
Выпускающая кафедра	Автоматики и компьютерных систем

Типовые задания для контрольной работы

Тема 1. Декларативное программирование. Лямбда-исчисление.

Цель сформировать представление о декларативной парадигме программирования и в частности, функциональном программировании, как одной из современных методологий разработке программного обеспечения. Лямбда-исчисление как математическая основа функционального программирования. Нормальный и аппликативный порядок вычислений.

Примеры вопросов контрольной работы:

- в чем принципиальное отличие императивной парадигмы от декларативной?
- каким образом осуществляются повторные вычисления в функциональных программах при отсутствии состояния программы и «операции присваивания»?
- какого назначения анонимных функций?
- выполнить редукцию терма $(\lambda x y z. x y z)(\lambda x. x x)(\lambda x. x) x$

Тема 2. Основы языка Common Lisp: типы данных, лямбда-функции, формы, универсальные и базовые функции.

Примеры вопросов контрольной работы:

- Построить функцию, реализующую математическое выражение $\sqrt{a^2 + b^2 + c^2}$.
- Построить функцию, реализующую математическое выражение $\frac{a^2 + b^2}{c^2}$.
- Построить функцию, реализующую математическое выражение $\frac{\sin a + \sin b + \sin c}{\cos abc}$.

Тема 3. Организация вычислений в Common Lisp. Условные вычисления. Рекурсия. Хвостовая рекурсия.

Примеры вопросов контрольной работы:

- Построить функцию, реализующую математическое выражение так, чтобы повторные вычисления не осуществлялись $\frac{\sin ab}{\cos^2 ab}$

– Определена функция

```
(defun f(x) (if (null (cdr x)) nil (cons (f (cdr x)) (cons (car x) (cdr x)))))
```

Определить, какое значение вернет вызов:

```
(f '(a b c))
```

– Реализовать рекурсивную функцию (не используя специальные функции и функционалы), получающую список (четной длины) чисел и возвращающую список пар исходных элементов

```
(1 2 3 4) => ((1 2) (3 4))
```

– Построить функцию, реализующую математическое выражение так, чтобы повторные вычисления не осуществлялись $\frac{abc}{abc + \ln ab}$

– Определена функция

```
(defun f(x) (if (null x) () (cons (car x) (f (cdr x)))))
```

Определить, какое значение вернет вызов:

```
(f '(a b c))
```

– Реализовать рекурсивную функцию (не используя специальные функции и функционалы), получающую список (четной длины) чисел и возвращающую список, содержащий элементы, которые находились на четных позициях исходного списка

```
(1 2 3 4) => (2 4)
```

Тема 4. Функции высокого порядка. Встроенные функционалы. Функции как объекты первого рода.

Примеры вопросов контрольной работы:

– Построить функционал, получающий две функции одного аргумента и возвращающий функцию одного аргумента, представляющую композицию функций (f1 (f2)).

– Построить функционал, получающий две функции одного аргумента и возвращающий функцию одного аргумента, представляющую сумму функций (f1 + f2).

– Построить функционал, получающий две функции одного аргумента и возвращающий функцию одного аргумента, представляющую произведение функций (f1 * f2).

Типовые задания для экзамена

1. Декларативное и императивное программирование. Языки функционального программирования.
2. Лямбда-исчисление Черча.
3. Основные элементы семантики и синтаксиса языка Lisp. Области применения функционального программирования.
4. Структурные единицы Lisp. Понятия атомов, списков и s-выражений.
5. Базовые функции языка Lisp. Основные функции обработки списков.
6. Точечная пара. Предикаты. Дополнительные функции обработки списков.
7. Арифметические функции. Логические функции.
8. Понятие символа. Функции связывания символа и его значения.
9. Анонимные функции, λ -выражения и λ -вызовы. Именованные функции (функции пользователя). Ключевые слова. Функциональная блокировка.
10. Организация ввода-вывода средствами языка Lisp. Функции ввода-вывода. Форматированный ввод-вывод.

11. Управляющие конструкции: последовательные вычисления. Разветвление вычислений.
12. Управляющие конструкции: условные предложения. Циклические вычисления.
13. Функционалы. Отображающие и применяющие функционалы.
14. Функции высоких порядков.
15. Энергичные и ленивые вычисления.
16. Определение рекурсии. Трассировка. CAR- и CDR-рекурсия.
17. Множественные терминальные ветви. Множественные рекурсивные ветви.
18. Представление списков через списочные ячейки и через точечные пары. Сборка мусора.