

**«Функциональное программирование»: вопросы к экзамену  
СПБАУ, осенний семестр 2015г., II курс**

Экзаминационный билет содержит два вопроса: первый — по  $\lambda$ -исчислению и прочей теории, второй — по языку Haskell.

**Теория**

1. Сравнение функционального и императивного подходов к программированию.
2. Основы  $\lambda$ -исчисления.  $\lambda$ -термы, свободные и связанные переменные.
3. Подстановка  $\lambda$ -терма. Лемма подстановки.
4.  $\alpha$ - и  $\beta$ -конверсии.  $\eta$ -конверсия и экстенциональная эквивалентность.
5. Расширения чистого лямбда-исчисления.  $\delta$ -конверсия.
6. Кодирование булевых значений, кортежей в чистом бестиповом  $\lambda$ -исчислении.
7. Кодирование чисел Чёрча в чистом бестиповом  $\lambda$ -исчислении.
8. Теорема о неподвижной точке.  $Y$ -комбинатор.
9. Редуксы. Одношаговая и многошаговая редукция. Нормальная форма. Редукционные графы.
10. Теорема Чёрча-Россера и её следствия.
11. Стратегии редукции. Теорема о нормализации. Механизмы вызова в функциональных языках.
12. Функция предшествования для чисел Чёрча. Комбинатор примитивной рекурсии.
13. Просто типизированное  $\lambda$ -исчисление в стиле Карри. Предтермы. Утверждения о типизации. Контексты. Правила типизации.
14. Просто типизированное  $\lambda$ -исчисление в стиле Чёрча. Предтермы. Утверждения о типизации. Контексты. Правила типизации.
15. Свойства систем просто типизированного  $\lambda$ -исчисления.
16. Связь между системами Карри и Чёрча. Проблемы разрешимости. Сильная и слабая нормализация.
17. Понятие главного (наиболее общего) типа. Подстановки типа и их композиция. Унификаторы.
18. Алгоритм унификации.
19. Алгоритм построения системы ограничений.
20. Теорема Хиндли-Милнера.
21. Рекурсивные типы.  $\mu$ -нотация.
22. Катаморфизмы и анаморфизмы.
23. Зипперы. Контексты с дыркой.

## Haskell

1. Основы программирования на Haskell. Связывание. Рекурсия. Базовые конструкции языка.
2. Основные встроенные типы языка Haskell. Система модулей. Частичное применение, каррирование.
3. Операторы и их сечения в Haskell. Бесточечный стиль.
4. Ошибки. Основание. Строгие и нестрогие функции. Ленивое и энергичное исполнение.
5. Алгебраические типы данных. Сопоставление с образцом, его семантика.
6. Объявления `type` и `newtype`. Метки полей.
7. Списки, стандартные функции для работы с ними. Генерация (выделение) списков.
8. Специальный полиморфизм. Классы типов. Объявления представителей. Классы типов `Eq`, `Ord`, `Functor`.
9. Внутренняя реализация классов типов.
10. Стандартные классы типов: `Enum` и `Bounded`, `Num` и его наследники, `Show` и `Read`.
11. Свёртки списков. Правая и левая свёртки. Энергичные версии.
12. Законы для функторов. Аппликативные функторы. Класс типа `Applicative` и его представители.
13. Моноиды. Представители класса типов `Monoid`.
14. Классы типов `Foldable` и `Traversable`.
15. Монады. Класс типов `Monad`. Законы для монад. `do`-нотация.
16. Стандартные монады: `Maybe` и списки.
17. Ввод-вывод в чистых языках. Монада `IO`. Взаимодействие с файловой системой.
18. Монады `Reader` и `Writer`.
19. Монада `State`.
20. Классы типов `Alternative` и `MonadPlus`.
21. Монадическая обработка исключений: класс `MonadError`.
22. Трансформеры монад.
23. Линзы ван Лаарховена.