***«Введение в функциональное программирование»***

***полугодовой, осенний семестр***

***лектор – доцент Малышко В.В***

***общая трудоемкость – 3 зачетные единицы***

***форма отчетности – экзамен***

***Аннотация.***

Спецкурс «Введение в функциональное программирование» посвящён знакомству слушателей с парадигмой функционального программирования на примере языка Scheme. Для успешного прохождения курса слушателям необходимо выполнить задания по написанию программ на Scheme. Первое задание посвящено созданию аналога известной программы ELIZA, второе – решению задач с помощью генетических алгоритмов.

## Программа курса

Основные сведения о языке Scheme. Имена, связывания, окружения. Выражения и правила их вычисления. Внешнее представление значений. Функции и специальная форма define. Анонимные функции и специальная форма lambda. Специальные формы: cond, if, case, and, or, begin, let, quote. Базовые типы данных: числа, литеры, строки, списки. Блоки и блочная структура программы. Подстановочная модель вычислений. Аппликативный и нормальный порядки вычислений.

Рекурсия и итерация. Функции и процессы, которые они порождают. Рекурсивные и итеративные процессы. Характеристики сложности процесса: количество шагов и размер памяти. Хвостовая рекурсия. Примеры реализации функций, порождающих рекурсивные и итеративные процессы.

Функции высшего порядка. Функции в роли аргументов других функций. Функции как обобщённые схемы вычислений (map, filter, foldl, foldr). Функции как возвращаемые значения. Остаточные вычисления. Программирование в стиле передачи остаточных вычислений.

Структуры данных. Проектирование структур данных. Типовые операции структуры данных: конструкторы и селекторы. Построение слоистых систем с помощью структур данных. Точечные пары как основа реализации структур данных. Функции работы с точечными парами (cons, car, cdr). Стрелочные диаграммы. Деревья и бинарные деревья поиска в Scheme.

Присваивание. Модель вычислений с окружениями. Внутренние переменные состояния. Специальная форма set! Преимущества и издержки присваивания. Мутаторы и точечные пары (set-car!, set-cdr!). Связывание, кадры, окружения. Правила вычисления в модели с окружениями. Стрелочные диаграммы окружений. Тождественность и эквивалентность (eq?, equal?). Реализация в Scheme стека, таблиц. Мемоизация. Векторы в Scheme. Реализация xеш-таблиц.

Макросы в языке Scheme. Система макросов syntax-rules. Основания для использования макросов в программе. Характеристики системы syntax-rules: гигиеничность, прозрачность ссылок, закрытость. Язык образцов. Язык шаблонов. Спецсимволы в образцах. Специальные формы и макросы.

Объектно-ориентированное программирование в Scheme. Обобщённые операции. Объекты данных как альтернатива обобщённым операциям. Базовые понятия объектно-ориентированного программирования: класс, экземпляр класса, механизм передачи сообщений, наследование, множественное наследование, ассоциация. Три точки зрения на ООП: модель, использование, реализация. Диаграммы классов и диаграммы объектов. Объектная система в Scheme и её элементы (функции create-instance, ask, get-method; схема реализации обработчика сообщений). Структура экземпляра класса, self. Описание объектной системы моделью окружений.

Математические основы функционального программирования. Основные сведения о λ-исчислении. λ-Нотация. Классическое λ-исчисление. λ-Выражения. Свободные и связанные переменные. Подстановки. α-Редукция и β-редукция. Эквивалентность и λ-редукция. Нормальная форма. Стратегии редукции при поиске нормальной формы. Теорема Черча-Россера и её следствия. Комбинаторы.

***Литература***

1. Абельсон Х., Сассман Дж. и др. Структура и интерпретация компьютерных программ. М.: Добросвет. 2010. 608 с.

2. Харрисон Дж. Введение в функциональное программирование. Перевод с английского. Новосибирск. 2009. [PDF]. [https://funprog-ru.googlecode.com/files/intro2fp-20090518.pdf]

3. Чернов А. В. «Доктор». Задание практикума по функциональному программированию. М.: Издательский отдел ВМК МГУ. 2006. [PDF]. [http://ejudge.ru/study/5sem/doctor.pdf]

4. Brown J. H. Advanced Scheme: Some Naughty Bits. Scheme Macros. MIT. 2003. [PDF]. [http://people.csail.mit.edu/jhbrown/scheme/macroslides03.pdf]

Ссылки:

1. Сайт среды программирования DrRacket [http://drracket.org]

2. Веб-страница курса «Введение в функциональное программирование» [http://sp.cs.msu.ru/scheme]

3. Веб-страница онлайновой учебной группы в социальной сети Вконтакте [http://vk.com/sp\_scheme]

4. Чернов А. В. Веб-страница практикума на языке Scheme. [https://ejudge.ru/study/teaching/index.php/3\_курс,\_5\_семестр]

***«Основы обработки текстов»***

***полугодовой, осенний семестр***

***лектор – доцент Турдаков Д.Ю.***

***общая трудоемкость – 3 зачетные единицы***

***форма отчетности – экзамен***

***Аннотация***

Курс знакомит слушателей с основными проблемами компьютерной обработки текстов и современными подходами к их решению. Рассматриваются как фундаментальные понятия и идеи, так и современные исследования в данной области. Особое внимание уделяется статистическим методам, которые активно развиваются в настоящее время и показывают лучшие результаты.

Актуальность обработки текстовой информации на естественном языке связана с ростом количества документов, доступных для автоматической обработки, потребностью в их анализе и наличием вычислительных ресурсов. Слушатели получат базовые знания в области обработки текстов на естественном языке, понимание проблем, возникающих при разработке систем текстового анализа, и подходов к их решению.

## Программа курса

Введение. Задачи обработки текста. Многозначность при обработке текста. Проблема понимания. Тест Тьюринга. Китайская комната.

Регулярные выражения и конечные автоматы. Распознавание языка с помощью КА. Построение КА для регулярных выражений. Методы поиска словосочетаний. Общая схема. Методы поиска кандидатов. Проверка статистических гипотез.

Языковые модели и задача определения частей речи. Модель N-грамм. Оценка вероятности высказывания. Методы сглаживания. Оценка качества. Тренировочный и проверочный корпуса. Задача определения частей речи. Существующие подходы. Алгоритмы, основанные на правилах. Алгоритмы, основанные на трансформации.

Скрытые марковские модели. Вероятность последовательности. Прямой алгоритм. Наиболее правдоподобное объяснение. Использование скрытой марковской модели для определения частей речи. Алгоритм Витерби. Методы классификации документов. Наивный байесовский классификатор. Логистическая регрессия. Модель максимальной энтропии.

Контекстно-свободные грамматики и синтаксический анализ. Типы грамматик. Грамматика составляющих. Грамматика зависимостей. Категориальная грамматика. Контекстно-свободные грамматики. КС грамматики и регулярные языки. Банк деревьев. синтаксический разбор. Разбор сверху вниз и снизу вверх. Алгоритм Кока-Янгера-Касами (CKY parsing). Эквивалентность КС грамматик. Группировка (chunking).

Статистические методы синтаксического анализа. Стохастические контекстно-свободные грамматики. Разрешение синтаксической многозначности. Моделирование языка. Обучение стохастических КС грамматик. Вероятностная версия алгоритма Кока-Янгера-Касами. Оценка качества. Проблемы стохастический КС грамматик. Алгоритм Коллинза.

Лексическая семантика. WordNet. Значения слов. Разрешение лексической многозначности. Алгоритмы классификации. Самонастройка. Методы, основанные на словарях и тезаурусах. Варианты алгоритма Леска. Методы оценки качества. Семантическая близость слов. Подходы на основе тезаурусов. Подходы на основе статистик. Методы оценки качества.

Информационный поиск. Ранжирование документов. Векторная модель. Взвешивание терминов. Индексирование. Инвертированный индекс. Запросы с джокером. Исправление опечаток. Вопросно-ответные системы: общая архитектура. Обработка запроса. Извлечение фрагментов текста. Обработка ответа. Автоматическое реферирование: общая архитектура.

Машинный перевод. Классические подходы. Статистический машинный перевод. Модель зашумленного канала. Модель перевода на основе фраз. Выравнивание фраз. Декодирование. Выравнивание слов. Модель IBM Model 1. Тренировка моделей выравнивания.

Тематическое моделирование. Вероятностная латентная семантическая модель. Латентное размещение Дирихле. Робастные модели.

***Литература***

1. Daniel Jurafsky and James H. Martin. 2008. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Second Edition. Prentice Hall.
2. Christopher D. Manning and Hinrich Schütze. 1999. Foundations of Statistical Natural Language Processing. MIT Press.
3. Steven Bird, Ewan Klein, and Edward Loper. Natural Language Processing with Python. O'Reilly Media, 2009 (<http://www.nltk.org/book>)
4. Mark Lutz. Learning Python, 5th Edition. O'Reilly Media, June 2013
5. Jimmy Lin and Chris Dyer. Data-Intensive Text Processing with MapReduce. Morgan & Claypool Publishers, 2010.
6. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. Издательство Вильямс. 2011
7. Тоби Сегаран, “Программируем коллективный разум” . издательство Символ-Плюс. 2008

***«Конструирование компиляторов»***

***полугодовой, весенний семестр***

***лектор - профессор Гайсарян С. С.***

***общая трудоемкость – 3 зачетные единицы***

***форма отчетности – экзамен***

***Аннотация.***

Спецкурс «Конструирование компиляторов» обеспечивает слушателей необходимыми знаниями и навыками в области оптимизирующей компиляции, а также в таких смежных областях, как статический анализ для выявления дефектов, обратная инженерия, генерация тестовых покрытий и др.. Целью освоения курса является получение базовых знаний в области разработки и применения современных компиляторов и компиляторных сред для разработки программ и для решения некоторых задач по обеспечению безопасного функционирования программ, для решения которых применяются компиляторные технологии.

## Программа курса

Введение. Описание процесса компиляции. Структура оптимизирующего компилятора. Основные вопросы, изучаемые в курсе. Построение промежуточного представления программы. Базовые блоки и граф потока управления. Биткод среды LLVM – пример промежуточного представления.

Локальная оптимизация. Метод нумерации значений: представление базового блока в виде направленного ациклического графа. Анализ потока данных – основной метод глобальной оптимизации. Примеры анализа потока данных – анализ достигающих определений и анализ живых переменных. Вынесение инвариантных вычислений за пределы цикла.

Граф потока управления: остовное дерево, обход, нумерация вершин, классификация дуг. Отношение доминирования и построение дерева доминаторов. Построение естественных циклов и гнезд циклов. SSA-форма промежуточного представления и ее построение. Граница доминирования. Анализ потока данных в SSA-форме. Выявление доступных выражений. Исключение избыточности.

Обоснование анализа потока данных: полурешетки, передаточные функции, общий итерационный алгоритм. Методы ускорения анализа потока данных. Суперблоки и другие области графа потока управления.

Вычисление передаточных функций областей по передаточным функциям составляющих их базовых блоков. Пример – анализ достигающих определений.

Глобальный метод нумерации значений с использованием дерева доминаторов. Глобальный анализ указателей. Псевдонимы (алиасы). Недостаточность глобального анализа.

Межпроцедурный анализ. Граф вызовов. Методы учета контекста.

Задачи, решаемые на этапе машинно-ориентированной оптимизации. Планирование кода.

Задачи решаемые на этапе машинно-ориентированной оптимизации. Распределение регистров.

Другие методы оптимизации: оптимизация потока управления, возвраты из рекурсивных функций. Раскрутка циклов. Открытая вставка функций.

Генерация объектного кода методом переписывания дерева

***Литература***

1. Keith D. Cooper, Linda Torczon. Engineering a Compiler, Second Edition. 2012 Elsevier, Inc..

2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools. 2nd Edition Addison-Wesley.2006 (last modified 2008) Есть русский перевод 2010.

3. Steven S. Muchnick Advanced Compiler Design & Implementation. Morgan Kaufman Publishers, 1997

4. Y.N. Srikant, Priti Shankar. The Compiler Design Handbook, Second Edition, CRC Press, 2008.

***«Основы программной инженерии»***

***полугодовой, весенний семестр***

***лектор - доцент Кулямин В.В.***

***общая трудоемкость – 3 зачетные единицы***

***форма отчетности – экзамен***

***Аннотация.***

Курс знакомит слушателей с технологическими проблемами разработки крупномасштабных программных систем, отличающими ее от разработки небольших программ, и методами решения этих проблем.

Слушатели курса получат хорошее представление об использовании современных высокотехнологичных методов в разработке сложного промышленных программного обеспечения, о практических способах построения качественных, гибких и масштабируемых систем в условиях жесткой конкурентной борьбы на рынке ПО.

## Программа курса

Разработка сложного ПО: источники проблем и методы их решения. Основные приемы преодоления сложности.

Жизненный цикл ПО. Основные процессы и модели жизненного цикла.

Моделирование предметной области. Потребности и требования к ПО. Качество ПО. Методы его обеспечения и контроля.

Архитектура ПО. Элементы UML. Проектирование ПО и шаблоны проектирования. Архитектура распределенных приложений на примере современных Web-приложений.

Компонентная разработка ПО. Элементы языков программирования Java и C#. Разработка Web-приложений в технологиях J2EE и .NET.

Принципы разработки пользовательского интерфейса. Разработка пользовательского интерфейса с помощью JSP и ASP.NET.

Направления развития компонентных технологий.

Управление проектами по разработке ПО.

***Литература***

1. И. Соммервилл. Инженерия программного обеспечения. М.: Вильямс, 2002.
2. Л. Басс, П. Клементс, Р. Кацман. Архитектура программного обеспечения на практике. СПб.: Питер, 2006.
3. У. Ройс. Управление проектами по созданию программного обеспечения. М.: Лори, 2002.
4. Г. Буч, Дж. Рамбо, А. Джекобсон. Язык UML. Руководство пользователя. М.: ДМК, 2000.
5. Л. Константайн, Л. Локвуд. Разработка программного обеспечения. СПб.: Питер, 2004.
6. Э.Дж. Брауде. Технология разработки программного обеспечения. СПб.: Питер, 2004.
7. Е.А. Жоголев. Лекции по технологии программирования: Учебное пособие. М., Издательский отдел факультета ВМиК МГУ, 2001.
8. Д. Леффингуэлл, Д. Уидриг. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. М.: Вильямс, 2002.
9. А. Якобсон, Г. Буч, Дж. Рамбо. Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2002.
10. К. Бек. Экстремальное программирование. СПб.: Питер, 2002.
11. В.В. Липаев. Методы обеспечения качества крупномасштабных программных средств. М., Синтег, 2003.
12. Э. Хант, Д. Томас. Программист-прагматик. М.: Лори, 2004.
13. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер-ДМК, 2001.
14. М. Фаулер и др. Архитектура корпоративных программных приложений. М.: Вильямс, 2004.
15. Р. Монсон-Хейфел. Enterprise JavaBeans. СПб ., Символ-Плюс, 2002.
16. Р. Андерсон и др. ASP.NET для профессионалов. Тома I, II. Лори, 2005.
17. J. Gosling, B. Joy, G. Steele, and G. Bracha. Java Language Specification, 3-rd edition. Addison Wesley Professional, 2005.

<http://java.sun.com/docs/books/jls/download/langspec-3.0.pdf>

1. C# Language Specification. Standard ECMA-334. 4-th edition, June 2006.

<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>.