ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО  
  
протокол № 18 / 03   
  
от « 31 » мая 2018 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ (FUNCTIONAL PROGRAMMING)

|  |  |
| --- | --- |
| Направление подготовки (специальность) | 09.03.04 Программная инженерия |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Семестр** | **Интерактив** | **Трудоемкость, кред.** | **Общий объем курса, час.** | **Лекции, час.** | **Практич. занятия, час.** | **Лаборат. работы, час.** | **СРС, час.** | **КСР, час.** | **Форма(ы) контроля, экз./зач./КР/КП** |
| 8 |  | 2 | 72 | 10 | 20 | 0 | 42 | 0 | З |
| ИТОГО | 0 | 2 | 72 | 10 | 20 | 0 | 42 | 0 |  |

Группа: Б18-504, Б18-514

АННОТАЦИЯ

Дисциплина «Функциональное программирование» формирует у студентов понимание принцыпов функционального программирования.

1. ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целями освоения учебной дисциплины «Функциональное программирование» являются:

– изучение теоретических аспектов функционального программирования на примере абстрактного функционального языка и языка функционального программирования Haskell;

– изучение основ языка Haskell и привитие навыков практической работы с этим языком.

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Данная дисциплина является курсом по выбору в цикле дисциплин направления.

Для успешного обучения требуются сложившиеся навыки работы с компьютером и успешное освоение следующие дисциплин:

• Информатика

• Дискретная математика (математическая логика)

• Дискретная математика (логические исчисления)

• Дискретная математика

Желательно знакомство с основами лямбда-исчисления.

3. КОМПЕТЕНЦИИ СТУДЕНТА, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ/ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ОБРАЗОВАНИЯ И КОМПЕТЕНЦИИ СТУДЕНТА ПО ЗАВЕРШЕНИИ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОПК-1 – Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности

ОПК-2 – Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности

ОПК-3 – Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

ПК-3 – владением навыками использования различных технологий разработки программного обеспечения

4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п.п** | **Наименование раздела учебной дисциплины** | **Недели** | **Лекции, час.** | **Практ. занятия / семинары, час.** | **Лабораторные работы, час.** | **Обязат. текущий контроль (форма\*, неделя)** | **Аттестация раздела (форма\*, неделя)** | **Максимальный балл за раздел\*\*** |
|  | *8 Семестр* |  |  |  |  |  |  |  |
| 1 | Раздел 1 | 1-5 | 5 | 10 |  |  | КИ-5 | 26 |
| 2 | Раздел 2 | 6-10 | 5 | 10 |  |  | КИ-10 | 26 |
|  | *Итого за 8 Семестр* |  | 10 | 20 | 0 |  |  | 52 |
|  | **Контрольные мероприятия за 8 Семестр** |  |  |  |  |  | З | 48 |

\* – сокращенное наименование формы контроля

\*\* – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

|  |  |
| --- | --- |
| **Обозначение** | **Полное наименование** |
| КИ | Контроль по итогам |
| З | Зачет |

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Недели** | **Темы занятий / Содержание** | **Лек., час.** | **Пр./сем., час.** | **Лаб., час.** |
|  | *8 Семестр* | 10 | 20 | 0 |
| **1-5** | **Раздел 1** | 5 | 10 |  |
| 1 | **Тема 1. Исходные определения.**  Введение. Структуры данных. Типы. Аппликация. Абстракция. Выражение в форме оператор/операнд. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 2 | **Тема 2. Абстрактный язык функционального программирования.**  Нотация (клозы, образцы, охрана). Операционная семантика. Функции высших порядков. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 3 | **Тема 3. Приемы программирования.**  Разбор случаев. Введение подфункций. Накапливающий параметр. Хвостовая рекурсия. Итерационная форма программ. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 4 | **Тема 4. Введение в Лисп.**  Атомы и списки. Точечное представление. Базовые функции обработки данных и примитивы управления. Связывание, деструктивное присваивание. Имя и значение символа. Определение функций. Применяющие функционалы. Замыкания. Макросы. Ввод и вывод. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 5 | **Тема 5. Синтаксически-ориентированное программирование .**  Язык абстрактного описания областей. Построение абстрактного описания (синтаксиса) списков, списочных структур, деревьев, бинарных деревьев. Построение типовых алгоритмов обработки данных для этих областей. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| **6-10** | **Раздел 2** | 5 | 10 |  |
| 6 | **Тема 6. Доказательства свойств программ.**  Определение индукции по структурам данных. Списочная индукция, sexpr-индукция. Примеры. Ограничения. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 7 | **Тема 7. Тематические основания.**  Формальная система. Аксиоматезируемость. Синтаксическое и семантическое равенство. Элементы лямбда-исчисления (выражение, подстановка, аксиомы, правила вывода, стратегии редукции, нормальная форма, Теорема Черча-Россера, пародоксальный комбинатор). | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 8 | **Тема 8. Трансформации программ.**  Эквивалентные и корректные преобразования. Система преобразований Берсталла и Дарлингтона: instantiation, folding, unfolding, low, abstraction. Трансформационный синтез программ и его проблемы. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 2 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 9 - 10 | **Тема 9. Частичные вычисления.**  Остаточная программа. Интерпретатор. Компилятор. Генератор компиляторов. Проекции Футамуры. Принципы построения частичных вычислителей. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 4 |  |
| Онлайн | Онлайн | Онлайн |
|  |  |  |

Сокращенные наименования онлайн опций:

|  |  |
| --- | --- |
| **Обозначение** | **Полное наименование** |
| ЭК | Электронный курс |
| ПМ | Полнотекстовый материал |
| ПЛ | Полнотекстовые лекции |
| ВМ | Видео-материалы |
| АМ | Аудио-материалы |
| Прз | Презентации |
| Т | Тесты |
| ЭСМ | Электронные справочные материалы |
| ИС | Интерактивный сайт |

ТЕМЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

|  |  |
| --- | --- |
| **Недели** | **Темы занятий / Содержание** |
|  | *8 Семестр* |
| 1 - 2 | **Элементы программирования на абстрактном функциональном языке.** Элементы программирования на абстрактном функциональном языке. |
| 3 - 4 | **Программирование функций для работы со списками на Haskell.** Программирование функций для работы со списками на Haskell. |
| 5 - 6 | **Программирование функций вычисления сумм рядов** Программирование функций вычисления сумм рядов |
| 7 - 8 | **Программирование аналитических преобразований.** Программирование аналитических преобразований. |
| 9 - 10 | **Программирование аналитических преобразований.** Программирование аналитических преобразований. |
| 11 - 12 | **Программирование синтаксических анализаторов** Программирование синтаксических анализаторов |
| 13 - 14 | **Доказательство свойств функций.** Доказательство свойств функций. |
| 15 - 16 | **Реализация семестровых проектов по индивидуальным заданиям.** Реализация семестровых проектов по индивидуальным заданиям. |

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Основными используемыми образовательными технологиями являются лекции и практические занятия, на которых студенты прослушивают теоретический материал, решают предлагаемые задачи, консультируются с преподавателем относительно семестровых заданий.

Дополнительно на усмотрение преподавателя могут применяться презентационные технологии, предусматривающие использование компьютерной техники как преподавателем, так и студентами. В частности лекции могут сопровождаться демонстрациями слайдов и трансляцией работы преподавателя с интерпретатором функционального языка, а демонстрация готовых семестровых работ может выполняться на ноутбуках студентов.

6. ТРЕБОВАНИЯ К ФОНДУ ОЦЕНОЧНЫХ СРЕДСТВ В РАМКАХ РЕАЛИЗУЕМОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Итоговый балл за каждый раздел составляется из суммы оценок за тесты и сданные домашние задания.

Количество баллов за тест формируется следующим образом:

Оценка Кол-во баллов

5 3

4 2

3 1

2, неявка 0

Максимальная оценка за домашнее задание составляет 20 баллов. Критерии его оценки приведены ниже. Оценка домашнего задания проводится индивидуально для каждого студента.

В каждом разделе курса содержится два теста и одно домашнее задание. Максимальный балл за каждый раздел — 26.

Самостоятельная работа студента включает повторение теоретического материала и выполнение ДЗ.

Зачет заключается в письменном ответе на два вопроса. Один по теоретической части курса, другой по практической. Каждый оценивается максимум в 11 баллов.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА:

1. ЭИ К88 Функциональное программирование : конспект лекций, Москва: НИЯУ МИФИ, 2020

2. ЭИ Ч-88 Функциональное программирование на С++ : учебное пособие, Москва: ДМК Пресс, 2020

3. 004 С32 Функциональное и логическое программирование : учебное пособие для вузов, Г. М. Сергиевский, Н. Г. Волченков , Москва: Академия, 2010

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

1. 004 Е42 Функциональное и логическое программирование : учебное пособие, Москва: Бином. Лаборатория знаний, 2011

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

Специальное программное обеспечение не требуется

LMS И ИНТЕРНЕТ-РЕСУРСЫ:

1. Haskell (http://Haskell.org)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Специальное материально-техническое обеспечение не требуется

9. ##Definition not found: 'static\_section\_edu\_stud'##

В качестве оценочного средства используется 100 бальная семестровая система, учитывающая посещаемость занятий, активность (выполнение домашних занятий, тестов). Каждый раздел проходит аттестацию.

Итоговый балл за каждый раздел составляется из суммы оценок за тесты и сданные домашние задания.

Количество баллов за тест формируется следующим образом:

Оценка Кол-во баллов

5 3

4 2

3 1

2, неявка 0

Максимальная оценка за домашнее задание составляет 20 баллов. Критерии его оценки приведены ниже. Оценка домашнего задания проводится индивидуально для каждого студента.

В каждом разделе курса содержится два теста и одно домашнее задание. Максимальный балл за каждый раздел — 26.

Самостоятельная работа студента включает повторение теоретического материала и выполнение ДЗ.

Зачет заключается в письменном ответе на два вопроса. Один по теоретической части курса, другой по практической. Каждый оценивается максимум в 11 баллов.

Методические указания по выполнению домашнего задания

За весь курс студент должен выполнить три домашних работы по соответствующим разделам курса. Задания распространяются на семинарских занятиях и в электронном виде по почте. Актуальное задание и методические указания можно получить по ссылке http://ubuntuone.com/p/1Hsy/. Работы выполняются на домашних компьютерах самостоятельно. Для ответов на вопросы по домашнему заданию выделяется время во время семинарского занятия.

Выполнение домашнего задания подразумевает:

 Реализацию задания в соответствии с номером варианта на выбранном языке

функционального программирования (F#, Haskell, O’Caml);

 Написание отчета по домашнему заданию.

При решении задания не следует применять императивные возможности языка. В тоже время поощряется использование приемов функционального программирования, такие как: функции высших порядков, накапливающий параметр (аккумулятор), мемоизация и др.

Оформление отчета.

Отчеты оформляются в электронном виде в одном из форматов: Open Document,

Microsoft Word, RTF, PDF, PostScript. Отчет должен содержать в себе:

 ФИО студента, номер группы, номер варианта;

 Номер и название лабораторной работы;

 Выполняемый вариант задания;

 Описание метода решения;

 Исходный текст программы;

 Соображения по тестированию программы и/или тестовые примеры там, где это требуется;

 Протокол работы программы;

 Выводы.

Оценка работ.

Оценка за работу выставляется исходя из:

 Эффективности / оригинальности предложенного решения;

 Полноты оформления отчета (в особенности описание метода, соображения по тестированию, выводы);

 Ответа на дополнительные вопросы;

 Временного фактора (задержка в предоставлении отчета).

До 8-й недели студентам даётся билет из 6 задач, общим весом 25 баллов. На 8-й неделе подводится итоговый контроль по результатам сдачи задач.

Пример билета для решения:

1. Создание конечных списков из N элементов:

Сконструировать список натуральных чисел. N = 20

Допускается использование генераторов списков, определений типов, конструирующих функций. В случае использования функций привести пример вызова построенной функции.

2. Создание бесконечных списков:

Сконструировать бесконечный список факториалов.

Допускается использование генераторов списков, определений типов, конструирующих функций. В случае использования функций привести пример вызова построенной функции.

3. Построить функцию, вычисляющую N-ый элемент ряда:

F(x, n) = xn

Дать пример вызова построенной функции и ожидаемого результата.

4. Функции работы со списками. Построить функцию:

GetN(L, n) — функция вычленения N-ого элемента из заданного списка

Дать пример вызова построенной функции и ожидаемого результата.

5. Функции с аккумулятором. Написать функцию из задания 3, но с использованием накапливающего параметра.

6. Функции с аккумулятором. Написать функцию из задания 4, но с использованием накапливающего параметра.

Первый две задачи имеют вес по 3 балла. Третья и четвёртая задача оцениваются по 4 балла. Пятая задача оценивается в 5 баллов и шестая задачи оценивается в 6 баллов.

До конца семестра студентам даётся творческое задание и сложная задача. Примеры задач:

Составить список всех функций библиотеки Prelude.hs, работающих с натуральными числами. Описать не менее трех.

Реализовать функцию перевода арифметической формулы из инфиксной записи в префиксную, где операции следуют перед операндами. Выражение представляется либо строкой, либо списком на выбор.

Первая задача оценивается в 5 баллов. Вторая задача оценивается в 20 баллов. Общее количество баллов за второй раздел – 25 баллов.

На зачёте студенты проходят тестирование на основе контрольно-измерительных материалов курса. Контрольно-измерительные материалы включают в себя задания трёх категорий:

А. Вопросы с четырьмя вариантами ответов, один из которых верный. По 1 баллу за правильный ответ.

B. Вопросы с простым открытым ответом, который является числом или короткой последовательностью символов. По 3 балла за правильный ответ.

C. Задача, требующая раскрытия хода решения. До 25 баллов за ответ.

Общее количество баллов за зачёт – 50.

Примеры заданий категории А:

В чистом функциональном программировании …

1) Отсутствует оператор присваивания

2) Допускается присваивание только локальным переменным

3) Всегда присутствует оператор присваивания

4) Не запрещается оператор присваивания

Что не является S-выражением?

1) a : b

2) a : []

3) a b

4) [a, b]

Пример задания категории B:

B1. Сконструировать конечный список из N = 200 натуральных чисел, используя конструктор списка.

Возможные ответы:

[1..200]

или

[1,2..200]

Пример задания категории С:

C1. Реализуйте функцию, вычисляющую N-ый член ряда Фибоначчи, в виде хвостовой рекурсии.

C2. Доказать, что .

C3. Докажите эквивалентность двух определений reverse

Определение 1 (рекурсивное):

reverse [] = []

reverse (H : T) = (reverse T) \* [H]

Определение 2 (с аккумулятором):

reverse' L = rev L []

rev [] L = L

rev (H : T) L = rev T (H : L)

Требуется доказать, что .

На зачёте студентам предлагается 15 вопросов категории А (по 1 баллу за ответ), 5 вопросов категории B (по 3 балла за ответ) и 1 вопрос категории С (20 баллов).

Таким образом, максимальное количество баллов за зачет – 15\*1+5\*3+1\*20 = 50.

Рабочая программа дисциплины составлена в соответствии с ОС НИЯУ МИФИ (ФГОС) и учебным планом основной образовательной программы (программ).

Автор(ы):

|  |  |  |
| --- | --- | --- |
|  | Рословцев Владимир Владимирович |  |