Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

02.03.02

Функциональное и логическое программирование

Лабораторная работа № 5. Функции высших порядков. Рекурсия.

Каждое задание должно быть загружено на личный git-репозиторий отдельным коммитом. Лабораторная работа выполняется в одном файле. Защита работы возможна на любой лабораторной работе от 5 до 16.

Если часть задач выполнена в один коммит, работа не проверяется. Если все коммиты сделаны в один час, работа не проверяется.

Общее задание. Познакомиться с синтаксисом языка F#.

Для этого необходимо выполнить следующие задания.

Задание 1. Реализовать Hello World.

Задание 2. Написать функцию решения квадратного уравнения.

Задание 3. Написать функцию вычисления площади круга и другую функцию вычисления объема цилиндра с использованием написанной функции. Прочитать данные и вывести результат. программу реализовать в функции main с помощью только оператора суперпозиции, потом только с помощью оператора каррирования.

Задание 4. Новая программа «Работа с числами». Написать рекурсивную функцию для нахождения суммы цифр числа. Использовать рекурсию вверх.

Задание 5. Написать рекурсивную функцию с телом выражение для нахождения суммы цифр числа. Использовать рекурсию вниз. Написать рекурсивную функцию с телом выражение для нахождения суммы цифр числа на основе хвостовой рекурсии.

Задание 6. Написать функцию, которая принимает аргумент и возвращает функцию. Первый аргумент, тип логический, если он ИСТИНА, возвращаем функцию, считающую сумму цифр числа, если он ЛОЖЬ, возвращаем функцию, считающую факториал числа. Проверить эту функцию на нескольких примерах.

Задание 7. Написать функцию обход числа, которая выполняет операции на цифрами числа, принимает три аргумента, число, функция (например, сумма, произведение, минимум, максимум) и инициализирующее значение. Функция должна иметь два Int аргумента и возвращать Int.

Задание 8. Протестировать эту функцию на операциях сумма, произведение, минимум, максимум цифр числа. Для тестирования и передачи аргумента использовать лямбда выражения. Инициализирующее заполнение должно иметь значение по умолчанию.

Задание 9. Реализовать функцию обход числа с условием, которая выполняет операции над цифрами, если цифры удовлетворяют заданному условию. Аргументы функции: число, функция с двумя аргументами Int, возвращающая Int, инициализирующее заполнение, функция с одним аргументом Int, возвращающая true-false.

Задание 10. Проверить функцию на 3 различных примерах.

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

Задание 11. Спросить у пользователя, какой язык у него любимый, если это F# или Prolog, ответь пользователю, что он — подлиза, для других языков придумать комментарий, реализовать функцию, принимающую аргументом ответ пользователя и возвращающую наш ответ пользователю.

Задание 12. Предыдущую программу реализовать в функции main с помощью только оператора суперпозиции, потом только с помощью оператора каррирования.

Задание 13. Написать функцию обход взаимно простых компонентов числа, которая выполняет операции над числами, взаимно простыми с данным, принимает три аргумента, число, функция (например, сумма, произведение, минимум, максимум, количество) и инициализирующее значение. Функция должна иметь два Int аргумента и возвращать Int.

Задание 14. Протестировать написанную функцию. Построить на её основе функции для вычисления числа Эйлера.

Задание 15. На основе написанных функций построить функции обход взаимнопростых с условием. Протестировать.

Задание 16. «Работа с числами». Составить 3 функции для работы с цифрами или делителей числа на основании варианта с использованием только хвостовой рекурсии и функций высших порядков. Каждый метод отдельный коммит.

Вариант № 1.

Метод 1. Найти сумму простых делителей числа.

Метод 2. Найти количество нечетных цифр числа, больших 3.

Метод 3. Найти прозведение таких делителей числа, сумма цифр которых меньше, чем сумма цифр исходного числа.

Вариант № 2.

Метод 1. Найти количество чисел, взаимно простых с заданным.

Метод 2. Найти сумму цифр числа, делящихся на 3.

Метод 3. Найти делитель числа, являющийся взаимно простым с наибольшим количеством цифр данного числа.

Вариант № 3.

Метод 1. Найти максимальный простой делитель числа.

Метод 2. Найти произведение цифр числа, не делящихся на 5.

Метод 3. Найти НОД максимального нечетного непростого делителя числа и прозведения цифр данного числа.

Вариант № 4.

Метод 1. Найти количество четных чисел, не взаимно простых с данным

Метод 2. Найти максимальную цифры числа, не делящуюся на 3.

Метод 3. Найти произведение максимального числа, не взаимно простого с данным, не делящегося на наименьший делитель исходно числа, и суммы цифр числа, меньших 5.

Вариант № 5.

Метод 1. Найти количество делителей числа, не делящихся на 3.

Метод 2. Найти минимальную нечетную цифру числа.

Метод 3. Найти сумму всех делителей числа, взаимно простых с суммой цифр числа и не взаимно простых с произведением цифр числа.

Вариант № 6.

Метод 1. Найти сумму непростых делителей числа.

Метод 2. Найти количество цифр числа, меньших 3.

Метод 3. Найти количество чисел, не являющихся делителями исходного числа, не взамно простых с ним и взаимно простых с суммой простых цифр этого числа.

Вариант № 7.

Метод 1. Найти сумму простых делителей числа.

Метод 2. Найти количество нечетных цифр числа, больших 3.

Метод 3. Найти прозведение таких делителей числа, сумма цифр которых меньше, чем сумма цифр исходного числа.

Вариант № 8.

Метод 1. Найти количество чисел, взаимно простых с заданным.

Метод 2. Найти сумму цифр числа, делящихся на 3.

Метод 3. Найти делитель числа, являющийся взаимно простым с наибольшим количеством цифр данного числа.

Вариант № 9.

Метод 1. Найти максимальный простой делитель числа.

Метод 2. Найти произведение цифр числа, не делящихся на 5.

Метод 3. Найти НОД максимального нечетного непростого делителя числа и прозведения цифр данного числа.

Вариант № 10.

Метод 1. Найти количество четных чисел, не взаимно простых с данным

Метод 2. Найти максимальную цифры числа, не делящуюся на 3.

Метод 3. Найти произведение максимального числа, не взаимно простого с данным, не делящегося на наименьший делитель исходно числа, и суммы цифр числа, меньших 5.

Задание 19. «Работа с числами». Модифицировать функции из предыдущего задания с использованием функций, построенных в заданиях 1 — 17. Выполнить максимально-возможное разделение на функции. Придерживаться принципа чистых функций, корректно именовать все выделенные функции.

Задание 20. Напишите программу, в которой пользователь вводит кортеж из двух чисел, где первое число это номер одной из трех функций вашего варианта, второе число аргумент этой функции. Построить функцию, которая принимает номер от 1 до 3 и возвращает одну из трех написанных функций. Далее программа выполняет указанную функцию и выдает результат на экран. Для реализации функции main использовать только оператор каррирования, потом только оператор суперпозиции.

Вопросы для защиты.

1. Объясните смысл ключевого слова let.

2. Расскажите, как задавать функции? Как указывать типы?

3. Что такое кортеж? Объясните принцип каррирования. Какой оператор используется, приведите примеры. Объясните разницу в написании и работе функций, принимающих кортеж и использующих каррирование.

4. Что такое сопоставление с образцом? Объясните, как это синтаксически реализуется, приведите примеры.

5. Как вы понимаете смысл фразы функциональный тип? Приведите примеры.

6. Что такое рекурсия вверх и рекурсия вниз? Поясните разницу на примерах.

7. Напишите рекурсию вверх и рекурсию вниз без помощи компилятора и IDE для вычисления факториала числа и числа фибонначи с заданным номером.

8. Объясните смысл хвостовой рекурсии и необходимость ее применения. Покажите примеры, где дана хвостовая и не хвостовая рекурсия.

9. Объясните смысл термина функции высших порядков. Приведите примеры.

10. Объясните смысл термина чистые функции. Приведите примеры.