

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)
Кафедра автоматизации обработки информации

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Методические указания к лабораторным работам и организации
самостоятельной работы
для студентов направления
«Программная инженерия»
(уровень бакалавриата)

2022

Салмина Нина Юрьевна

Функциональное: Методические указания к лабораторным работам и организации самостоятельной работы для студентов направления «Программная инженерия» (уровень бакалавриата) / Н.Ю. Салмина. – Томск, 2022. – 32 с.

© Томский государственный университет
систем управления и радиоэлектроники,
2022

© Салмина Н.Ю., 2022

Оглавление

1 Введение.....	4
2 Методические указания к проведению лабораторных работ.....	5
2.1 Лабораторная работа «Простейшие функции и рекурсия»	5
2.2 Лабораторная работа «Функционалы и формирователи списков».....	12
2.3 Лабораторная работа «Пользовательские типы. Классы»	19
2.4 Лабораторная работа «Графы и деревья».....	23
2.5 Лабораторная работа «Работа с файлами»	26
3 Методические указания для организации самостоятельной работы	30
3.1 Общие положения	30
3.2 Проработка лекционного материала	30
3.3 Самостоятельное изучение тем теоретической части курса	31
3.3.1 Обработка и хранение знаний: свойства символов, ассоциативные списки	31

1 Введение

Цель дисциплины – ознакомление студентов с основами функционального программирования и λ -исчисления на примере языка Хаскел, формирование у студентов профессиональных знаний и практических навыков по разработке функциональных программ с использованием рекурсии, функций высших порядков.

Изучение данной части дисциплины включает в себя: теоретический раздел (изучение теоретического материала); практический раздел (выполнение лабораторных и контрольных работ); итоговый контроль результата изучения дисциплины. Данное пособие содержит в себе методические указания и варианты заданий для лабораторных работ, вопросы по организации самостоятельной работы.

2 Методические указания к проведению лабораторных работ

2.1 Лабораторная работа «Простейшие функции и рекурсия»

Цель работы

Целью данной работы является знакомство с интерактивной средой работы GHCi и получение первичных навыков работы в этой среде путем написания простейших функций с использованием базовых функций и рекурсии.

Рекомендации по подготовке к работе

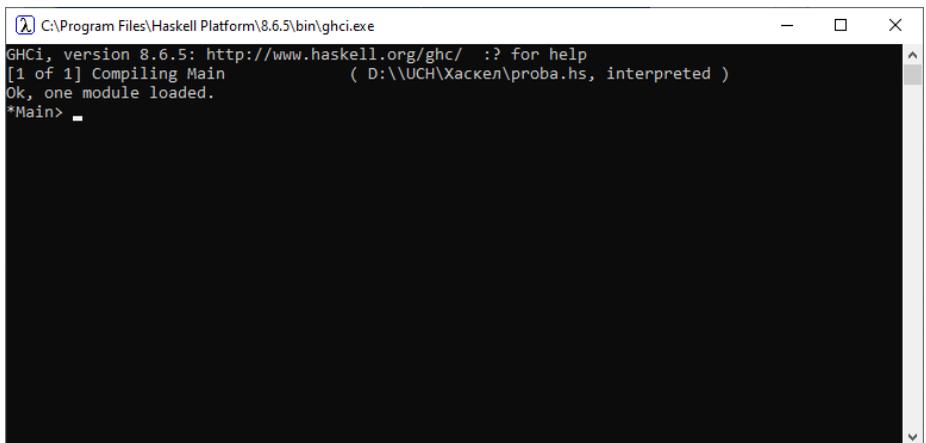
Для того, чтобы писать программу Haskell необходимо создать текстовый файл с расширением *.hs. В этом файле пишем текст программы: декларацию типов и описание функций.

Например, файл может содержать следующую программу:

```
leng :: [a] -> Integer
leng x = if null x then 0 else 1 + leng (tail x)
```

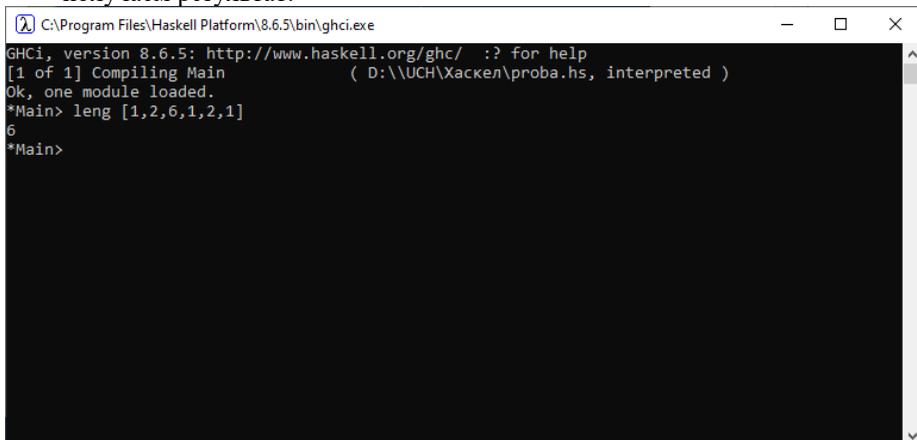
Запуск программы на выполнение можно осуществить двумя способами:

- 1) Ассоциативный запуск. Просто запускаем файл с расширением *.hs – в результате открывается окно с интерактивной средой GHCi, которая работает с Haskell-программами, в которой выдается приглашение на ввод выражений, которые мы хотим вычислить:



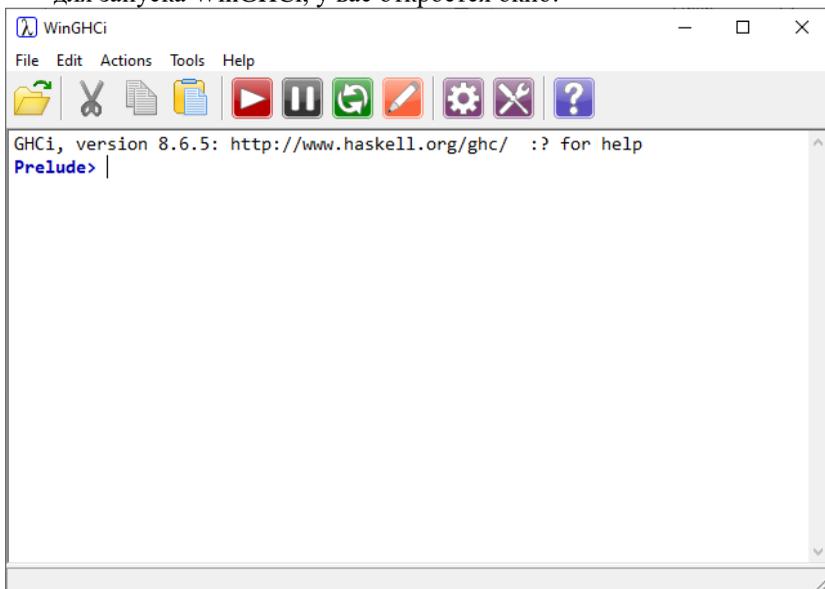
```
C:\Program Files\Haskell Platform\8.6.5\bin\ghci.exe
GHCi, version 8.6.5: http://www.haskell.org/ghc/ :? for help
[1 of 1] Compiling Main (D:\\UCH\Хаскел\proba.hs, interpreted)
Ok, one module loaded.
*Main> _
```

Вводим выражение (имя функции с фактическими аргументами), и получаем результат:



```
C:\Program Files\Haskell Platform\8.6.5\bin\ghci.exe
GHCi, version 8.6.5: http://www.haskell.org/ghc/  :? for help
[1 of 1] Compiling Main          ( D:\\UCH\Хаскел\proba.hs, interpreted )
Ok, one module loaded.
*Main> leng [1,2,6,1,2,1]
6
*Main>
```

2) Запуск WinGHCi: в Пуске найдите Haskell Platform 8.6.5 и выберите для запуска WinGHCi, у вас откроется окно:



```
WinGHCi
File Edit Actions Tools Help
[Icons: Folder, Scissors, Copy, Paste, Play, Stop, Refresh, Erase, Settings, Wrench, Help]
GHCi, version 8.6.5: http://www.haskell.org/ghc/  :? for help
Prelude> |
```

Для запуска собственной программы откройте файл с расширением *.hs, после чего можно вызывать собственные функции с фактическими параметрами:

```
GHCi, version 8.6.5: http://www.haskell.org/ghc/ :? for help
Prelude> :cd D:\UCH\Hackel
Prelude> :load "proba.hs"
[1 of 1] Compiling Main          ( proba.hs, interpreted )
Ok, one module loaded.
*Main> leng [1,4,6,12,3,4]
6
*Main> |
```

В процессе выполнения работы необходимо написать две функции, описанные в вашем варианте задания. При создании функций используйте рекурсию. Для этого вам необходимо сначала определить терминальную ветвь (при необходимости терминальных ветвей может быть несколько), которая определяет правило останова. Правило останова может определять две ситуации: если мы выполнили все возможные действия и не достигли нужного результата; если мы достигли нужного результата.

На втором этапе спланируйте рекурсивную ветвь (ветви), которая определяет рекурсивный вызов функции с упрощением аргументов. При каждом рекурсивном обращении мы должны приближаться к условию останова!

Внимание! Не использовать встроенные функции!

Исключением являются только те варианты, где явно указано, какую функцию можно использовать.

Варианты заданий

1. а) определите функцию умножения двух целых чисел через сложение и вычитание (учесть, что числа могут быть как положительными, так и отрицательными).
б) Напишите функцию, которая из данного списка строит список списков его элементов, например, $[1,2,3,4,5] \Rightarrow [[1],[2],[3],[4],[5]]$.

2. а) определите функцию от двух аргументов x и n , которая создает список из n раз повторенных элементов x .
б) напишите функцию, производящую взаимную замену y на x и x на y : $x \rightarrow y, y \rightarrow x$.
Например: $(f\ 2\ 0\ [0, 3, 2, 2, 5, 8, 0]) \Rightarrow [2, 3, 0, 0, 5, 8, 2]$

3. а) Определите функцию $(f\ n)$ (n кратное 3), вычисляющую сумму: $1*2*3+4*5*6+\dots+(n-2)*(n-1)*n$
б) Напишите функцию $p(L, N1, N2)$, которая удаляет элементы списка L с позиции $N1$ до позиции $N2$ включительно.

4. а) определите функцию, аргументом которой является список с четным количеством элементов – атомов. Функция должна менять местами каждую пару элементов списка (1-й элемент со 2-м, 3-й с 4-м и т.д.). Если количество элементов списка является нечетным, функция должна просто отбрасывать последний элемент.
б) Определите функцию, на вход которой подаются два списка – множества. Функция должна выдавать объединение этих множеств (можно использовать функцию `elem`).

5. а) напишите функцию, имеющую два аргумента: числовой список и целое число N . Функция должна возвращать список, в котором элементы, стоящие на четных позициях, увеличены в N раз.
б) Определите функцию, на вход которой подаются два списка – множества. Функция должна искать пересечение этих множеств (можно использовать функцию `elem`).

6. а) Написать функцию, на вход которой подается список из двух подсписков чисел. Результатом работы функции должен стать список, элементами которого являются поэлементные суммы этих двух подсписков. Например: $[[1, 3, 2],[4, 6, 3]] \implies [5, 9, 5]$
 б) Определить функцию, которая определяет, является ли данное натуральное число простым
7. а) Написать функцию, на вход которой подается список атомов. Функция должна формировать список из двух подсписков следующим образом: в 1-й подсписок включаются все элементы исходного списка, стоящие на нечетных позициях, во 2-й – на четных. Например: $[1,2,3,4,5] \implies [[1,3,5],[2,4]]$
 б) Определите функцию $(f\ s)$, вычисляющую знакопеременную сумму $a_1 - a_2 + a_3 - a_4 + \dots + a_k \cdot (-1)^{(k+1)}$ для списка s , имеющего вид $(a_1\ a_2\ a_3\ \dots\ a_k)$.
8. а) Определите функцию, аргументом которой является одноуровневый список чисел. Функция должна формировать новый список, состоящий из двух подсписков: в 1-й подсписок включаются положительные числа, во 2-й – все отрицательные числа.
 Например: $[1,2,-3,4,-5] \implies [[1,2,4],[-3,-5]]$
 б) Напишите функцию, на вход которой подается список, имеющий не менее двух элементов. Функция должна возвращать предпоследний элемент списка.
9. а) Напишите функцию, удаляющую из списка N -й элемент. Если N больше длины списка, то просто возвращаем исходный список.
 б) написать функцию, изменяющую исходный список следующим образом: после каждого встреченного нуля добавляется единица.
10. а) определите функцию, аргументом которой является целое число n , результатом которой является сумма цифр натурального числа n .
 б) Определите функцию, аргументом которой является одноуровневый список чисел. Функция должна менять все отрицательные числа на строку “minus”, положительные – на строку “plus”, нули – на строку “zero”.
 Например: $[1,0,-2,7] \implies [“plus”,”zero”,”minus”,”plus”]$

11. а) Напишите функцию, которая делает из списка множество, т.е. удаляет все повторяющиеся элементы (можно использовать функцию `elem`).
- б) Определите функцию $(f\ s\ n)$, где s – список, n – число. Функция должна возвращать n -й элемент из списка s . Будем считать, что n меньше длины списка.
12. а) функция $(f\ l)$ имеет единственный аргумент – список, состоящий из четного количества элементов – чисел. Каждая пара чисел в списке задает стороны прямоугольника. Функция должна возвращать список из периметров прямоугольников.
- б) Определите функцию двух аргументов $(f\ u\ v)$, которая возвращает список тех элементов списка u , которых нет в списке v . Например: $(f\ [1,2,3,4,5]\ [6,3,2,7]) \implies [1,4,5]$ (можно использовать функцию `elem`).
13. а) На вход функции подается два числовых списка $[a_1, a_2, \dots, a_n]$ и $[b_1, b_2, \dots, b_n]$. Функция должна формировать новый список $[\min(a_1, b_1), \min(a_2, b_2), \dots, \min(a_n, b_n)]$.
- б) Напишите функцию, зависящую от двух аргументов x и y , удаляющую все вхождения x в список y . X может быть атомом или списком.
14. а) напишите функцию, на вход которой подается числовой список и два числа: $n < k$. Функция должна формировать новый список, включая в него только те элементы исходного списка, которые удовлетворяют следующим условиям: выбираемые числа должны быть или меньше n , или больше k . Например: $(f\ [4,6,1,9,6]\ 2\ 7) \implies [1,9]$
- б) на вход функции подается список, элементами которого являются списки из двух атомов. Функция должна менять местами элементы в каждом из подсписков: $[[1,2],[3,4],[5,6]] \implies [[2,1],[4,3],[6,5]]$.
15. а). Даны два упорядоченных по возрастанию списка чисел X и Y . Написать функцию $f(X, Y)$, которая будет возвращать общий упорядоченный список элементов X и Y . Например, $f([1,3,5,7,8], [2,3,5,7]) \implies [1,2,3,3,5,5,7,7,8]$
- б) На вход функции подается числовой список с четным количеством элементов. Определить функцию, которая бы

проводила расчеты по следующей схеме: $(x_1 x_2 x_3 x_4 \dots)$

$\implies x_1 * x_2 + x_3 * x_4 + \dots$

Например: $[1, 2, 3, 4, 5, 6] \implies 44$

16. а) Напишите булеву функцию, которая проверяет, являются ли все элементы числового списка положительными.
б) определить функцию, которая находит максимальную сумму двух соседних чисел в заданном списке. Единственный аргумент функции является одноуровневым числовым списком
17. а) Определить функцию, на вход которой подается два числа K , L и числовой список, на выходе формируется список, положительные элементы которого увеличиваются в K раз, а отрицательные элементы увеличиваются в L раз. Например:
 $(2\ 3\ (-1\ 2\ 3\ -4\ 5)) \implies (-3\ 4\ 6\ -12\ 10)$
б) Написать функцию, на вход которой подается натуральное положительное число. Функция должна возвращать список его делителей.
18. а) На вход функции подается одноуровневый список чисел. Необходимо найти номер первого максимального элемента списка.
б) написать функцию, удаляющую из исходного списка строк все строки, начинающиеся на символ 'q'.
19. а) На вход функции подается одноуровневый список. Сформировать новый список, удалив из исходного все элементы, стоящие на нечетных позициях.
б) Напишите функцию от трех аргументов $(f\ s\ x\ y)$, которая возвращает True, если x и y являются соседними элементами в списке s , и False в противном случае.
20. а) написать функцию, изменяющую числовой список следующим образом: после каждой встреченной единицы следующий элемент заменяется на -99. Если единица – последний элемент списка, то -99 добавляется. Например:
 $[2,1,4,5,1,7,1] \rightarrow [2,1,-99,5,1,-99,1,-99]$
б) Определите функцию, на вход которой подаются два списка – множества. Функция должна выдавать объединение этих множеств (можно использовать функцию `elem`).

21. а) написать функцию, на вход которой подается числовой список, возвращающую список кортежей из числа и его названия. Например: [1,3,5] → [(1,one),(3,three),(5,five)]
б) Определите функцию, аргументом которой является одноуровневый список чисел. Функция должна формировать новый список, состоящий из двух подсписков: в 1-й подсписок включаются четные числа, во 2-й – все нечетные числа.
Например: [1,2,12,33,45] ==> [[2,12],[1,33,45]]
22. а) На вход функции подается список строк. Функция должна формировать список чисел, соответствующих длинам строк (можно использовать функцию length). Например: ["qw", "ssss", "xcv"] → [2,4,3]
б) Написать функцию, которая бы меняла знак каждого элемента исходного числового списка.

2.2 Лабораторная работа «Функционалы и формирователи списков»

Цель работы

Целью проведения данной работы является знакомство с формирователями списков, а также с функционалами и принципами их использования при написании новых функций.

Рекомендации по подготовке к занятию

В процессе выполнения лабораторной работы необходимо написать две функции, описанных в вашем варианте задания.

В первой функции обязательно использование формирователей списков.

Во второй функции обязательно использование функционалов (это могут быть как встроенные функционалы, так и написанные собственноручно).

При необходимости можно писать вспомогательные функции.

Также, по мере необходимости, не возбраняется при выполнении первого задания **дополнительно** использовать функционалы, а при выполнении второго задания **дополнительно** использовать формирователи списков.

P.S. Предикат – это функция, возвращающая в качестве результата булево значение.

Варианты заданий

1. а) Определите функцию, зависящую от одного аргумента – двухуровневого списка List, которая по данному списку формирует одноуровневый список из тех элементов List, которые кратны трем и при этом встречаются в List более одного раза.
б) Напишите функцию ($f \text{ p } x \text{ n}$), которая выдает истину только если ровно для n атомов двухуровневого списка x выполняется предикат p
2. а) Определить функцию трех аргументов: a , b и c . Первым аргументом является множество – список целых чисел, b и c – целые числа. Функция должна формировать список трех подмножеств данного множества, определяемых следующим образом:
 - а. Четные числа множества;
 - б. Числа множества, являющиеся квадратами числа;
 - с. Все числа $b \leq a_i \leq c$.Например: $f [1, 2, 3, 4] 0 2 \Rightarrow [[2, 4], [1, 4], [1, 2]]$
б) Напишите функцию ($f \text{ n list_p } x$), где n – число, list_p – список одноместных функций, x – числовой список. Функция должна возвращать список – результат применения n -й функции к элементам списка x
3. а) Напишите функцию трех аргументов: X , Y – числовые списки, Q – число. Функция должна находить сумму $S = x_i + y_j$, наиболее близкую к числу Q .
б) Задан список s кортежей вида ($\langle \text{символ} \rangle$, $\langle \text{число} \rangle$). Напишите функцию ($f \text{ p } s$), которая будет формировать список чисел, выбранных из кортежей, для символов которых выполняется предикат p . Например: $f_2_16 (\lambda x \rightarrow x > 'f')$
 $[('a',12),('x',3),('m',44)] \rightarrow [3,44]$
4. а) Задан список кортежей: $[(A_1, N_1), (A_2, N_2), \dots, (A_K, N_K)]$, где A_i – символьный атом, N_i – число. Написать программу, возвращающую список из двух символьных атомов исходного списка $[A_i, A_j]$ с максимальным произведением $N_i * N_j$.
б) Определите функцию $f \text{ p } s \text{ n}$, которая из списка чисел s создает новый список, изменив его следующим образом: те элементы, которые удовлетворяют предикату p , должны быть

увеличены в n раз, а те элементы, которые не удовлетворяют предикату p , должны быть увеличены на число n

5. а) Написать функцию, аргументом которой является числовой список. Функция должна возвращать список сумм: 1-го элемента, 2-х первых элементов, 3-х первых элементов и т.д. Например:
 $f [1, 2, 3, 4] \Rightarrow [1, 3, 6, 10]$.
б) Напишите функцию $(f\ s\ n)$, которая из двухуровневого списка чисел s создает новый одноуровневый список квадратов чисел, исключив все элементы исходного списка s , которые превышают заданное число n
6. а) Напишите функцию $(f\ s\ n)$, которая в двухуровневом списке чисел s суммирует только те числа, которые превышают заданное число n .
б) Определите функцию трех аргументов $f\ s\ k\ m$, которая из двухуровневого списка чисел s создает новый список, меняя знак у каждого атома, и при этом отрицательные числа должны быть уменьшены в k раз, а положительные – увеличены в m раз. Внимание: структура списка должна остаться прежней!
7. а) Напишите функцию $(f\ s)$, где s – двухуровневый список чисел, в котором каждый подсписок содержит ровно два числа: $[x, y]$. Функция должна возвращать одноуровневый список чисел u , для которых элементы x – положительные. Например:
 $f [[1, 12], [-4, 0], [-23, 23], [17, -3]] \rightarrow [12, -3]$.
б) Напишите функцию, на вход которой подается двухуровневый список и атом. Функция должна удалять из списка все элементы, совпадающие с указанным атомом. Внимание: структура списка должна остаться прежней!
8. а) На вход функции подается двухуровневый список чисел. Необходимо найти номер первого экстремального (максимального или минимального) элемента списка. Нумерация должна идти по атомам.
б) Определить функцию, которая находит максимальную сумму двух соседних чисел в заданном списке. Единственный аргумент функции является одноуровневым числовым списком.

9. а) Напишите программу, формирующую список простых чисел на заданном интервале.
 б) Написать функцию, имеющую два аргумента: V – список, N – число. Функция должна формировать список элементов из V , встречающихся в нем не менее N раз.
10. а) Написать программу, определяющую дизъюнктивную сумму между двумя множествами: $A+B=(A \cup B) - (A \cap B)$. Встроенные функции работы с множествами не использовать!
 б) Определите функцию $(f \text{ s } n)$, которая из двухуровневого списка чисел s создает новый одноуровневый список, прибавляя к каждому отрицательному числу число n , и оставляя положительные числа без изменений.
11. а) Последовательности Улама. Определим последовательность x_0, x_1, x_2, \dots следующим образом: x_0 - произвольное нечетное число, отличное от единицы. При $n > 0$ имеем $x_n = u(x_{n-1})$, где u - функция Улама, определяемая как
 $u(x) = x/2$, если x четно,
 $u(x) = 3x + 1$, если x нечетно.
 Последовательность заканчивается, когда в ней встречается значение 1. Например: 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.
 Напишите функцию $(f \text{ n})$, которая для данного нечетного n строит список - последовательность Улама, начинающуюся с n
 б) Напишите функцию $(\text{count } p \text{ x})$, которая подсчитывает, сколько атомов (элементов подсписков) в двухуровневом списке x удовлетворяет предикату p (p – функция или функциональное имя).
12. а) Простейшая система кодирования сообщений заключается в замене каждой буквы сообщения на букву, находящуюся на N -й по отношению к ней позиции в алфавите. Например, для $N=2$ буква “а” заменяется буквой “с”, буква “у” — буквой “а” и т.д. Напишите функцию *шифратор*, которая берет шифруемое слово и целое число и выдает слово, представляющее шифр данного слова, полученный с помощью указанного метода. Используйте для этого встроенные функции (модуль `Data.Char`) `chr x` (переводит целое в символ) и `ord x` (переводит символ в целое).

- б) Напишите функцию $(f\ n\ p\ x\ n)$, которая выдает истину только если ровно для n элементов списка x выполняется предикат p (p – функция или функциональное имя).
13. а) Напишите функцию $(f\ s\ n)$, которая из двухуровневого списка чисел s создает новый одноуровневый список квадратов чисел, исключив все элементы исходного списка s , которые меньше заданного числа n
- б) Определите функцию $(f\ a\ n)$, которая от двух числовых аргументов строит следующий список:
 $[a, a*(a+1), a*(a+1)*(a+2), \dots, a*(a+1)*(a+2)*\dots*(a+n)]$.
14. а) Определите функцию $(f\ u\ v)$, которая возвращает список элементов, содержащихся либо в списке U , либо в списке V , но не одновременно в U и V . Встроенные функции работы с множествами не использовать!
- б) Напишите функцию $(p\ x\ y\ w)$, осуществляющую замену элементов списка x на соответствующие элементы списка y в списке w , например:
 $> p\ [1,2]\ [10,20]\ [1,6,4,2,7,8,1]$
 $[10,6,4,20,7,8,10]$
15. а) Написать функцию $(f\ x\ d\ n)$, формирующую геометрическую прогрессию длиной n , с начальным значением x , и знаменателем прогрессии d .
- б) Напишите функцию $(f\ p\ s)$, где s – двухуровневый список чисел, в котором каждый подсписок содержит ровно два числа: $[x,y]$, а p – предикат. Функция должна возвращать одноуровневый список чисел y , для которых выполняется условие $(p\ x)$ – истина.
Например: $f\ even\ [[1,12],[4,0],[23,23],[10,-3]] \rightarrow [0,-3]$.
16. а) Написать функцию $(f\ x\ func\ n)$, которая формирует список чисел длиной n , первый элемент которого равен x , а каждый последующий элемент списка равен значению функции $func$ с аргументом – предыдущим элементом списка. Например:
 $(f\ 2\ (\lambda x \rightarrow x+1)\ 5) \rightarrow [2, 3, 4, 5, 6]$

- б) Напишите функцию $(f p_1 p_2 s)$, где s – одноуровневый список чисел, p_1 и p_2 – предикаты. Функция должна формировать из s двухуровневый список $[[x_1, x_2, \dots], [y_1, y_2, \dots]]$, в котором элементы первого подсписка – числа, удовлетворяющие предикату p_1 , элементы второго подсписка – элементы, удовлетворяющие предикату p_2 . Если элемент исходного списка не удовлетворяет ни одному предикату, то он в результирующий список не включается.
17. а) Напишите функцию f и v , формирующую список всех элементов списка U , не содержащихся в списке V . Встроенные функции работы с множествами не использовать!
 б) Определить функцию с двумя аргументами A и B – числовыми списками. Функция должна проверять, выполняется ли следующее соотношение между элементами списков:
 $b_i = a_i + 1$; $a_{i+1} = 2 * b_i$
18. а) Определите функцию, единственным аргументом которой является список кортежей ($\langle \text{число} \rangle$, $\langle \text{строка} \rangle$). Функция должна строить новый список кортежей: если строка начинается на букву 'a', то число возводится в квадрат; если строка начинается на букву 'b', то из числа извлекается корень; остальные элементы списка остаются без изменения.
 б) Задан числовой список. Написать программу, подсчитывающую среднее значение элементов списка, за исключением максимального и минимального элементов.
19. а) Определите функцию $(f s)$, которая в двухуровневом списке s переставляет все четные элементы в начало списка, нечетные – в конец, и делает его одноуровневым. Например,
 $f [[1, 3, 7], [45, 2, 8], [8, 2]] \rightarrow [2, 8, 8, 2, 1, 3, 7, 45]$.
 б) Напишите функцию $(f p n)$, которая будет выдавать список из первых n чисел последовательности Фибоначчи, для которых выполняется условие p (p – заданный предикат).
20. а) Определите функцию $(f a n)$, которая от двух числовых аргументов строит следующий список:
 $[a, a*(a+1), a*(a+1)*(a+2), \dots, a*(a+1)*(a+2)*\dots*(a+n)]$.

б) Напишите функцию ($f\ s\ n1\ n2$), которая в двухуровневом списке чисел s перемножает только те числа, которые лежат в пределах от $n1$ до $n2$.

21. а) Определить функцию трех аргументов: a , b и c . Первым аргументом a является список строк, b и c – целые числа ($b < c$). Функция должна формировать список трех подмножеств данного множества, определяемых следующим образом:

- а. строки, длина которых меньше b ;
- б. строки, длина которых лежит в пределах от b до c ;
- в. строки, начинающиеся на букву 'd'.

Например:

```
> f ["q","sds","dddd","dd"] 2 4  
[["q"], ["sds","dd"], ["dddd","dd"]]
```

б) Определить функцию с одним аргументом – одноуровневым списком. Функция должна подсчитывать количество вхождений в список каждого элемента и выдавать результат в виде списка кортежей. Например: ['a', 'a', 'e', 't', 'a', 't'] ==> [('a', 3) ('e',1), ('t', 2)].

22. а) Задан список s кортежей вида ($\langle \text{число1} \rangle$, $\langle \text{число2} \rangle$), где число1 – целое, число2 – вещественное. Напишите функцию ($f\ s$), которая будет формировать список чисел2, выбранных из кортежей, в которых число1 является квадратом числа. Например: $f\ [(12, 3.5), (16, 1.75), (4, -1.2), (7, 5)] \rightarrow [1.75, -1.2]$

б) Дан список имен. Найти самое короткое имя. Если в списке несколько самых коротких имен – выбирать имя, которое лексикографически меньше других. Внимание: список не сортировать!

2.3 Лабораторная работа «Пользовательские типы. Классы»

Цель работы

Целью проведения данной работы является знакомство с пользовательскими типами в Хаскеле, а также классами и принципами их использования при написании новых функций классов.

Рекомендации по подготовке к работе

1. Определить тип(ы) данных, в соответствии с заданием. Совет: в некоторых случаях, когда форматы данных строго определены (особенно для типов данных с несколькими конструкторами), лучше не использовать полиморфизм для типов аргументов.
2. При необходимости объявить собственный класс типов.
3. Определить функции класса(ов) для созданных типов данных.

Варианты заданий

1. Определите типы геометрических фигур: треугольник, круг, прямоугольник, ромб. Напишите для них класс вычисления параметров геометрических фигур с двумя функциями: вычисление площади и вычисление периметра фигуры.
2. На предприятии все должности упорядочены по старшинству. Например (в вузе) преподавательские должности можно упорядочить следующим образом: ассистент, преподаватель, старший преподаватель, доцент, профессор, декан, ректор. Информация по персоналу включает в себя: ФИО, должность, оклад. Определите тип данных для представления информации по отдельным людям. Определите для этого типа функции '<', '<=' класса Ord, сравнивая людей по старшинству должностей. При равных должностях сравниваются оклады.
3. Определите тип геометрических фигур: треугольник, прямоугольник, ромб, задавая фигуры через координаты вершин. Определите для этого типа функцию сравнения '=' класса Eq, а также функции '<', '<=' класса Ord, сравнивая фигуры по площади.
4. Определите тип данных для представления не бинарного дерева, с поименованными вершинами (и промежуточные и листья должны быть поименованы), при этом имена вершин могут быть любого типа. Определите для этого типа функцию сравнения '=' класса Eq, где два дерева равны, если имеют одинаковую структуру.

5. После слияния разных БД выяснилось, что даты хранятся в разных форматах: списком - [день, месяц, год], где месяц – число; кортежем – (день, месяц, год), где месяц задан атомом типа May; кортежем (год, месяц, день), где месяц задан числом. Опишите тип данных с тремя конструкторами (для каждого формата). Определите для этого типа функцию сравнения '==' класса Eq, а также функции '<', '<=' класса Ord.
6. Определите тип геометрических фигур: треугольник, прямоугольник, ромб, задавая фигуры через координаты вершин. Определите для этого типа функцию сравнения '==' класса Eq. Помните, что фигура может располагаться на плоскости в любом месте, и повернута под любым углом.
7. После слияния разных БД выяснилось, что информация о людях представлена в разных форматах: 1) фамилия, имя, отчество, дата рождения - [день, месяц, год], где месяц – число; 2) имя, отчество, фамилия, дата рождения – кортеж – (день, месяц, год), где месяц задан названием месяца; 3) фамилия, имя, отчество, дата рождения – тип Data1 (Int, Month, Int), где месяц задан названием месяца. Опишите тип данных с тремя конструкторами (для каждого формата). Определите для этого типа функцию сравнения '==' класса Eq, позволяющую выявлять одинаковых людей.
8. Определите тип данных для хранения следующей информации по студентам: фамилия; информация по экзаменационной сессии – [(x1,y1), (x2,y2), ...], где xi – название предмета, yi – оценка. Определите для этого типа функцию сравнения '==' класса Eq, а также функции '<', '<=' класса Ord, сравнивая студентов по суммарной оценке. Но, если студент имеет двойку по какому-либо предмету, то он «меньше» любого другого студента. Например, Иванов, имеющий оценки 3,3,3, при сравнении с Сидоровым, имеющим оценки 5,5,2, будет «больше».
9. Определите тип данных для хранения следующей информации по спортсменам, соревнующимся по прыжкам в длину: фамилия; информация по прыжкам – [(x1,y1), (x2,y2), ...], xi – номер попытки, yi – длина прыжка. Всего спортсмен может сделать от одной до трех попыток. Определите для этого типа функцию сравнения '==' класса Eq, а также функции '<', '<=' класса Ord, сравнивая спортсменов по максимальной длине прыжка.
10. В документации фирмы массивы данных с ценами по разным отделам и показателям представлена в трех разных видах: одноуровневый список [Float]; двухуровневый список [[Float]]; список кортежей [(String,Float)], где String – наименование показателя. Опишите тип данных с тремя конструкторами (для

- каждого формата). Напишите для него класс вычислений с тремя функциями: сложить все значения; перемножить все значения; сложить квадраты значений.
11. Определите тип данных для хранения следующей информации по книгам: автор; название книги; издательство; год издания; рейтинг (в числовом выражении). Определите для этого типа функцию сравнения '==' класса Eq; совпадают автор и название книги; а также функции '<', '<=' класса Ord, сравнивая книги по рейтингу.
 12. Определите тип данных для представления не бинарного дерева, с поименованными вершинами (и промежуточные и листья должны быть поименованы), при этом имена вершин могут быть любого типа. Определите для этого типа функцию сравнения '==' класса Eq, а также функции '<', '<=' класса Ord, сравнивая деревья по количеству листьев.
 13. Определите тип данных для хранения следующей информации по футбольным командам: название команды; информация по результатам сыгранных игр – [(x1,y1), (x2,y2), ...], где каждый кортеж – это информация по отдельной игре и xi – количество забитых голов, yi – количество пропущенных голов. Определите для этого типа функцию сравнения '==' класса Eq, а также функции '<', '<=' класса Ord, сравнивая команды по количеству набранных очков (выигрыш=3, ничья=1), если у двух команд равное количество набранных очков, то дальше сравнивают разницу между количеством забитых и количеством пропущенных голов.
 14. Определите тип данных для хранения следующей информации по спортсменам: фамилия; список занимаемых спортсменом мест в различных соревнованиях – [Int]. Определите для этого типа функции '<', '<=' класса Ord, сравнивая спортсменов по набранным суммарным баллам, если 1-е место дает 20 баллов, 2-е – 19 баллов и т.д.
 15. Определите тип данных для представления не бинарного дерева, с поименованными вершинами (и промежуточные и листья должны быть поименованы), при этом имена вершин могут быть любого типа. Определите для этого типа функции сравнения '<', '<=' класса Ord, сравнивая деревья по высоте.
 16. Определите тип данных для хранения следующей информации по студенческим группам: группа; фамилия старосты; информация по студентам – [(x1,y1), (x2,y2), ...], где xi – фамилия студента, yi – список его оценок. Определите для этого типа функцию сравнения '==' класса Eq, а также функции '<', '<=' класса Ord, сравнивая группы по сумме средних баллов всех студентов группы. Причем,

- если сумма средних баллов в двух группах равная, то дополнительно сравниваются средние баллы старост.
17. Определите тип данных для хранения следующей информации по людям: фамилия, имя, отчество, дата рождения - [день, месяц, год], где месяц задан названием месяца. Определить для этого типа функции сравнения '<', '<=' класса Ord, сравнивая людей по возрасту.
 18. Разные преподаватели хранят информацию по рейтингу студентов в разных форматах: 1) Фамилия_студента [список баллов]; 2) Фамилия_студента [(x1,y1), (x2,y2), ...], где xi – название теста, yi – баллы за тест. Опишите тип данных с двумя конструкторами (для каждого формата). Напишите для него класс вычислений с двумя функциями: 1) определение суммарного рейтинга студента; 2) определение оценки за предмет исходя из суммарного рейтинга (81-100 баллов – отлично, 61-80 баллов – хорошо, 41-60 баллов – удовлетворительно, <41 балла - неудовлетворительно).
 19. Определите типы данных для хранения следующей информации по продуктам и их составляющих: Продукт – название (строка); рецептура [(x1,y1), (x2,y2), ...], где xi –название ингредиента, yi –требуемое количество. Ингредиент – название ингредиента, стоимость единицы. Определить для типа Продукт функции сравнения '<', '<=' класса Ord, сравнивая Продукты по стоимости.
 20. Определите тип данных для хранения следующей информации по дисциплинам в учебном плане: наименование дисциплины; семестр; количество часов: [<лекции>, <практики>, <лабораторные работы>, <самостоятельная работа>] (количество часов практик или лабораторных работ может равняться 0); форма отчетности (Экзамен/Зачет/Зачет_с_оценкой). При формировании учебного плана иногда требуется менять дисциплины местами (по семестрам), для этого должна быть возможность их сравнения на равенство. Определить для этого типа функцию сравнения '==' класса Eq: две дисциплины равны, если у них совпадают общее количество часов и форма отчетности.
 21. Работа турагентства. Определите тип данных для хранения следующей информации по турам: название тура; длительность (в днях); дата начала тура (тип даты определите сами); вид отдыха (пляжный/экскурсионный/активный); стоимость. Определить для этого типа функцию сравнения '==' класса Eq: равны одновременно - вид отдыха, длительность, месяц начала тура и стоимость; а также функции '<', '<=' класса Ord, сравнивая туры по стоимости.
 22. После слияния разных БД выяснилось, что информация о товарах на складе представлена в разных форматах: 1) наименование

товара, количество, цена единицы товара; 2) (<наименование_товара>, <количество>, <цена единицы товара >); 3) наименование товара, [<количество>, <цена единицы товара>]. Опишите тип данных с тремя конструкторами (для каждого формата). Напишите для него класс вычислений с двумя функциями: вычислить общую стоимость всего товара на складе; вычислить суммарную стоимость конкретного товара (по наименованию).

2.4 Лабораторная работа «Графы и деревья»

Цель работы

Целью данной работы является работа с графами, представленными различными способами и написание функций для работы с ними.

Рекомендации по подготовке к работе

В процессе выполнения лабораторной работы необходимо написать программу определяющую для графа (дерева) заданную характеристику. При необходимости используйте локальные или вспомогательные функции.

Решение различных задач обработки графов напрямую зависит от способа их представления. Так, для поиска пути на графе удобнее использовать вектора смежности. Если же мы решаем задачу построения остовного дерева, граф удобнее представлять в виде списка ребер и множества узлов графа.

Представление деревьев зависит от вида дерева: обычное или бинарное, и от вида решаемой задачи.

Продумайте алгоритм, который позволит решить поставленную задачу. При необходимости вы можете воспользоваться существующими алгоритмами просмотра, изменения, корректировки графов и деревьев.

Если в вашем варианте не оговорен способ представления графа (дерева), вы можете выбрать тот способ представления, который будет более удобен для решения поставленной задачи.

Варианты заданий

23. Дан граф. Написать функцию, которая находит в графе максимальный цикл и выдает его в виде списка вершин. Если в графе нет циклов, функция должна возвращать nil.
24. Написать функцию, на вход которой подается граф, определяющую, связан ли граф.
25. Задан граф, у которого для каждой дуги задана ее длина: $((a \ b \ 12) \ (s \ d \ 3) \ \dots)$. Написать функцию, определяющую кратчайший путь между указанными двумя вершинами.
26. Определить функция, на вход которой подается граф. Функция должна определять, является ли он гамильтоновым, и если да, то найти гамильтонов цикл. Цикл в графе называется гамильтоновым, если он содержит все вершины графа ровно по одному разу; граф с таким циклом называется гамильтоновым.
27. Написать функцию, на вход которой подается неориентированный граф в виде $(a \ (b) \ b \ (a \ c \ d) \ c \ (b \ e \ f \ g) \ \dots)$. Функция должна преобразовывать граф в вид: $((a \ b) \ (b \ c) \ (b \ d) \ \dots)$.
28. Написать функцию, на вход которой подается граф в виде $(a \ (b) \ b \ (c \ d) \ c \ (e \ f \ g) \ \dots)$. Функция должна проверять, является ли граф ориентированным.
29. Определите функцию, аргументом которой является дерево. Функция должна вернуть ветвь с максимальным количеством листьев.
30. Определить функцию, которая ищет заданную вершину в дереве и возвращает список, содержащий предка искомой вершины и ее потомков: $(\text{предок} \ (\text{потомок1} \ \text{потомок2} \ \dots))$.
31. Написать функцию, которая проверяет, является ли заданный граф деревом (имеет ли циклы)?
32. На вход функции подается дерево. Функция должна добавить к каждой вершине информацию: глубину ветки, исходящей из данной вершины.
Например, дерево вида $(1 \ (2 \ (3 \ (4)))(5)) \ (6 \ (7)) \)$ должно преобразоваться в $((1 \ 3) \ ((2 \ 2) \ ((3 \ 1) \ ((4 \ 0)))(5 \ 0))) \ ((6 \ 1) \ ((7 \ 0)))$
33. Стяжение ветви. Определить функцию, аргументами которой является дерево и две его вершины. Функция должна склеивать два заданных узла, если они соседние и выдавать NIL в противном случае.
34. Напишите программу, определяющую эйлеровый путь, начинающийся с заданной вершины неориентированного графа. Путь называется эйлеровым, если проходит все ребра графа по одному разу. Если такой путь не существует – программа должна возвращать NIL.

35. Напишите программу, определяющую компоненты связности данного графа. Результатом работы программы должен быть список списков (компонента связности – список вершин).
36. Напишите программу, на вход которой подаются два дерева. Программа должна проверять, изоморфны ли данные деревья (Два дерева называются изоморфными, если можно отобразить одно из них в другое, изменением порядка ветвей в поддеревьях).
37. Напишите программу, которая будет строить список смежностей по представлению графа, заданному в виде матрицы смежностей.
38. Определить функцию, аргументом которой является дерево, подсчитывающую количество листьев в дереве.
39. Написать функцию, на вход которой подается дерево, определяющую максимальную глубину дерева.
40. Задан граф, у которого для каждой дуги задана ее длина: $((a\ b\ 12)$ $(s\ d\ 3)$...). Определить функцию, которая находит две самые удаленные друг от друга вершины.
41. Определить функцию, которая ищет середину кратчайшего пути между двумя заданными вершинами. Функция должна возвращать: атом-имя вершины (если между исходными вершинами нечетное количество вершин), список из двух вершин (если четное) и NIL, если заданные вершины – соседи.
42. Напишите программу, формирующую для ориентированного графа G его транзитивное замыкание G^* . Дуга $(a\ b)$ принадлежит G^* , если существует ориентированный путь из a в b .
43. Определите для бинарного дерева отношение `ordered(Tree)`, выполненное, если дерево `Tree` является упорядоченным деревом целых чисел, т. е. число, стоящее в любой вершине дерева, больше любого элемента в левом поддереве и меньше любого элемента в правом поддереве. Например, дерево `tree(nil, 5, tree(nil, 6, tree(tree(nil, 8, nil), 10, nil)))` является упорядоченным.
44. На вход функции подается ориентированный граф. Функция должна определять, существует ли путь из A в B . Если путь существует, выдать его в качестве результата работы, если пути нет, выдать пустой список.
45. Написать функцию, подсчитывающую количество циклов в графе.

2. 5 Лабораторная работа «Работа с файлами»

Цель работы

Целью данной работы является получение навыков работы с файлами: хранение, добавление, изменение информации, представленной в виде структур/записей.

Рекомендации по подготовке к работе

В процессе выполнения лабораторной работы необходимо написать программу работы с файлами, выполняющую заданные действия. Исходный файл с данными можно создавать любым способом. Помните, что структуры/записи в файле должны записываться с полным указанием имен полей, порядок атрибутов должен соответствовать их порядку в описании типа.

Если вы сомневаетесь, в каком виде данные должны храниться в файле, то запишите в файл хотя бы одну запись с помощью программы (организуя для этого диалог добавления записи в файл), а остальные записи можно уже будет вносить в файл вручную (по образцу и подобию).

Порядок выполнения работы:

- 1) Создайте файл с набором структур/записей, содержание которых определено в вашем задании. Набор данных должен быть достаточным, чтобы проверить работу вашей программы!
- 2) Напишите программу, выполняющую заданные действия с набором фактов. Допустимо использовать любые встроенные функции.
- 3) Ввод-вывод необходимой информации организовать в виде диалога, а по необходимости – через контекстное меню.

Варианты заданий.

1. В файле хранится следующая информация о полетах: <рейс>, <пункт вылета>, <пункт прилета>, <время вылета>, <время прилета>. По заданным X=«пункт вылета» и Y= «пункт прилета» сформировать список возможных рейсов и отсортировать его по времени перелета.
2. Результаты соревнований хранятся в виде базы фактов: матч(<команда1>,N1,<команда2>,N2), где Ni – количество голов, забитых командой i. Необходимо сформировать рейтинговую таблицу команд, если победа приносит команде 3 очка, а ничья – 1

- очко. В случае одинакового количества очков, рейтинг выше у той команды, которая забила больше голов.
3. В файле хранится следующая информация о животных: животное(<название>,<ареал_обитания>,<популяция>). Написать программу, которая бы позволяло править данные файла, вводя количество родившихся или умерших животных, а также формировать список вымирающих животных по заданной границе минимальной популяции.
 4. В файле хранится база фактов о людях в виде предиката: person(<фамилия>,<имя>,<пол>,<дата_рождения>). Сформировать и сохранить в другом файле список семей: семья (фамилия, имя_мужа, имя_жены, список_детей). Будем считать, что в список детей попадают люди возрастом менее 18 лет.
 5. В файле хранится информация о рейтинге спортсменов в виде: спортсмен(<фамилия>,<количество_набранных_баллов>). Написать программу, позволяющую вводить результаты соревнований в виде: спортсмен – занятое место. Корректировать файл рейтинга спортсменов, учитывая, что за 20-е место дается 1 балл, за 19-е – 2 балла, ..., за 1-е место – 20 баллов.
 6. В файле хранится информация расписания экзаменов в виде: экзамен(<группа>,<предмет>,<преподаватель>,<дата>). Написать программу, позволяющую выбирать информацию по экзаменам для группы/преподавателя, сортируя ее по дате.
 7. В файле хранится информация о музыкальных инструментах в следующем виде: инструмент(<название>,<группа>,<жанр>). Под группами инструментов подразумеваются: духовые, струнные, клавишные, ударные и т.д. «Жанр» - здесь хранится информация в виде списка, в каких жанрах музыки (поп, рок, классика и т.п.) может быть использован данный инструмент. Написать программу, формирующую: 1) список инструментов, используемых в заданном жанре; 2) какие группы инструментов используются в заданном жанре.
 8. Рейтинг группы по дисциплине хранится в файле в следующем виде: рейтинг(<фамилия>,[B1,B2,...]). Здесь Bi – баллы, набранные студентом. Написать программу, позволяющую добавлять баллы в таблицу рейтинга, а также сортировать группу по суммарному рейтингу.
 9. В файле хранится база фактов о полетах в следующем виде: fly(<рейс>,<пункт_вылета>,<пункт_прилета>,<L>). Здесь L – список дней, по которым летает данный рейс, например, L=[ежедневно], L=[пон,среда,птн]. Для заданных пункта отправления, пункта назначения и дня вылета выдать все возможные перелеты.

10. В файле хранятся результаты сессии группы в следующем виде: студент(<фамилия>,[Экз1,Оценка1],...[ЭкзN,ОценкаN]). Написать программу, выдающую список отличников, хорошистов, студентов с тройками и должников (оценку по не сданному экзамену можно отображать как «неуд»).
11. В БД фактов (в файлах) хранятся данные по преподавателям и группам:
лектор (<фамилия>,<дисциплина>) и студенты(<дисциплина>,<группа>). Написать программу, позволяющую: 1) для заданной группы формировать список преподавателей, ведущих у нее занятия; 2) изменять в предикате «лектор» информацию по преподавателю или по названию дисциплины. Учесть, что при изменении названия дисциплины необходимо корректировать и файл с информацией по группам.
12. В файле хранится информация о музыкальных альбомах в следующем виде:
музыка(<исполнитель/группа>,<альбом>,<время_звучания>,<год_выпуска>). Написать программу, позволяющую добавлять новые факты, а также формировать список альбомов по заданному исполнителю, сортируя по году выпуска.
13. В файлах хранится информация об актерах и фильмах в следующем виде: актер(<фамилия>,<название_фильма>), фильм(<название_фильма>,<жанр>,<год_выпуска>). Написать программу, формирующую список актеров по жанрам (если актер снимался в фильмах разных жанров, то определять его жанр исходя из максимального количества ролей в фильмах одного жанра).
14. В файле хранится информация о приеме врачей в поликлинике в следующем виде: врач(<фамилия_врача>,<фамилия_пациента>,t). Здесь t – время начала приема (под прием отводится полчаса). Написать программу, позволяющую записаться новому пациенту на прием. Для этого сначала вывести на экран свободные часы по запрашиваемому врачу, ввести фамилию пациента и выбранное им время, добавить новый факт в БД.
15. В файле хранится информация о книгах в следующем виде: книга(<автор>,<название>,<год_издания>,<цена>). Написать программу, позволяющую формировать список книг заданного автора, сортируя их по году выпуска и цене одновременно.
16. В файле хранится информация о соревнованиях по прыжкам в длину: спортсмен(<фамилия>,<попытка>,<длина_прыжка>). У каждого спортсмена есть три попытки. Написать программу, формирующую список результатов – спортсмен + лучшая попытка (максимальная длина), сортируя его по длине прыжка.

17. В файле хранится информация о блюдах в следующем виде: блюдо(<название>, <список ингредиентов>), а также информация об ингредиентах: ингредиент(<название>, <категория>). В качестве категорий рассмотрите: вегетарианский, диетический, мясной.
- Написать программу, позволяющую выбирать список блюд по заданной категории. Будем считать, что, если в списке ингредиентов есть хоть один мясной – блюдо мясное; блюдо вегетарианское – если нет мясных ингредиентов; блюдо диетическое – если все ингредиенты диетические.
18. В файле хранится информация по лекарствам в следующем виде: Лекарство(<наименование>,<список заболеваний>, <список противопоказаний>).
- Написать программу, позволяющую формировать список лекарств для заданного заболевания, с учетом заданных противопоказаний.
19. В файле хранится информация по материалам для изготовления мебели в следующем виде: Материал(<наименование>,<цена>).
- Написать программу, производящую расчет стоимости мебели исходя из заданного списка:
[а(материал1, количество1), а(материал2,количество2),...].
- Список пар материал-количество можно вводить по очереди.
20. В файле хранится информация о блюдах в следующем виде: блюдо(<наименование>,<раздел><категория >, <цена>). Написать программу, позволяющую формировать список блюд из указанного раздела (супы, салаты, вторые, десерт) и заданной категории (вегетарианское, диетическое, мясное), сортируя их цене. Раздел и категория могут учитываться одновременно и по отдельности.
21. В файле хранится информация о книгах в следующем виде: книга(<ФИО автора>,<название книги>, <рейтинг>).
- Написать программу, формирующую список книг по заданному автору (позволять вводить только фамилию – осуществить поиск подстроки). Список сортировать по рейтингу.
22. В файле хранится информация о продукции в виде следующих двух предикатов: Материал(<наименование>,<стоимость единицы>).
- Продукция(<наименование>,<список структур str>), где str(<материал>,<количество/объем>).
- Написать программу, подсчитывающую сумму заказа, в котором указывается наименование продукта и его количество.

23. В файле хранится информацию по турам в следующем виде:
тур(<название>,<страна>,<категория>,<дата начала>,<дата окончания>).
Написать программу, которая бы выдавала информацию по турам по заданной стране и в указанном периоде. Например, на запрос тура в Италию с 1.06.22 по 1.07.22 может быть выдана информация:
«Отдых в Италии» пляжный 5.06.22 15.06.22
«Италия историческая» экскурсионный 10.06.22 17.06.22

3 Методические указания для организации самостоятельной работы

3.1 Общие положения

Цель самостоятельной работы по дисциплине – проработка лекционного материала, самостоятельное изучение некоторых разделов курса, подготовка к лабораторным работам, тестам и контрольной работе.

Самостоятельная работа студента по дисциплине «Функциональное программирование» включает следующие виды его активности:

1. проработка лекционного материала;
2. изучение тем теоретической части дисциплины, вынесенных для самостоятельной проработки;
3. подготовка к лабораторным работам;
4. подготовка к тестам;
5. подготовка к контрольной работе;
6. подготовка к экзамену.

3.2 Проработка лекционного материала

При проработке лекционного материала необходимо:

- а) отработать прослушанную лекцию (прочитать конспект, прочитать учебное пособие и сопоставить записи с конспектом,

просмотреть слайды) и восполнить пробелы, если они имелись (например, если вы что-то не поняли или не успели записать);

б) перед каждой последующей лекцией прочитать предыдущую, дабы не тратилось много времени на восстановление контекста изучения дисциплины при продолжающейся теме.

Данный вид деятельности ориентирован как на закрепление материала, так и на подготовку к тестовым заданиям и контрольным работам.

Содержание лекций:

Концепция и особенности функционального программирования. Свойства функциональных языков. Основные особенности Хаскелл, достоинства языка.

Элементарные понятия языка. Понятие функции, префиксная нотация. Вычисление лямбда-выражений. Определение функций. Базовые функции языка, предикаты.

Понятие рекурсии. Правила записи рекурсивной функции. Терминальная ветвь, рекурсивная ветвь. Прямая и косвенная рекурсия. Рекурсия с несколькими терминальными ветвями, рекурсивными ветвями.

Внутреннее представление списков. Вспомогательные функции над списками.

Функции высших порядков. Различие между данными и функциями. Функционалы.

Работа с графами и деревьями: представление, обработка, поиск пути на графе.

3.3 Самостоятельное изучение тем теоретической части курса

3.3.1 Обработка и хранение знаний: свойства символов, ассоциативные списки.

Для обработки и хранения знаний существует два типа списков: списки свойств и ассоциативные списки.

С символом можно связывать, не только значение, но и информацию, называемую списком свойств (property list).

Списки свойств широко используются для представления смысловой информации, когда с атомом необходимо связать список определенных свойств и связанных с ними значений. Существуют специальные функции языка, позволяющие выполнять различные

действия с такими списками: присвоить, прочитать, заменить, удалить свойство. Кроме этого, всегда можно просмотреть информацию о списке свойств.

Ассоциативный список или просто а-список (a-list) есть основанная на списках и точечных парах структура данных, описывающая связи наборов данных и ключевых полей, для работы с которой существуют готовые функции.

Ассоциативные списки применяются при работе с динамическими базами данных в оперативной памяти. Ассоциативный список можно рассматривать как отображение множества ключей на множество соответствующих им объектов. Для работы с такими списками существуют функции создания ассоциативного списка и поиска элементов в ассоциативном списке. К сожалению, не существует готовой функции, позволяющей изменить значение ключа в таком списке. При необходимости функцию изменения значений в ассоциативном списке необходимо писать самостоятельно.

Рекомендуемые источники

Салмина Н. Ю., Функциональное программирование и интеллектуальные системы: учебное пособие [Электронный ресурс] / Салмина Н. Ю. — Томск: ТУСУР, 2016 . — 100 с. — Режим доступа: <https://edu.tusur.ru/publications/63572>).