

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

***КРОССПЛАТФОРМЕННОЕ ПРОГРАММИРОВАНИЕ***

Лабораторный практикум  
для студентов специальности 1 – 40 02 01  
«Вычислительные машины, системы и сети»  
дневной формы обучения

*Общие требования ко всем лабораторным по Scala:*

- единый стиль кода:  
<http://docs.scala-lang.org/style/>
- весь реализованный функционал должен быть покрыт тестами при помощи библиотеки ScalaTest;
- весь ключевой функционал не работает с консолью напрямую: функция возвращает единичные значения или коллекции – никакого прямого вывода в консоль;
- пользуйтесь подвыражениями;
- запрещается использовать:
  - var
  - return
  - ;
  - if (cond) true else false

*Теорию, необходимую для выполнения работ можно найти в методичке 2-й ветки.*

## Лабораторная работа №5

### Функциональное программирование с использованием языка Scala. Хвостовая рекурсия

*Задание к лабораторной работе* - выполнить задания первой и второй частей согласно варианту:

- не используя хвостовую рекурсию;
- используя хвостовую рекурсию.

#### *Часть 1*

1. Вывести все целые значения между двумя числами.
2. Найти минимальное число в списке.
3. Найти среднее арифметическое.
4. Найти длину самого большого слова в списке.
5. Найти количество вхождений символа в строке.
6. Заменить все строчные буквы на прописные.
7. Разделить все чётные числа на два, а к нечётным прибавить один.
8. Вывести позицию первого элемента массива, равного введённому числу.
9. Возвести число в введённую степень.
10. Найти сумму всех чисел в списке.
11. Найти произведение двух векторов.
12. Найти второй элемент стека считая со дна (top -> [4,3,2,1] <- base).
13. Найти N-й элемент списка.
14. На основе переданного списка создать новый, содержащий только четные элементы.
15. Перевернуть число (123 -> 321).

#### *Часть 2*

1. Найти локальные экстремумы ряда.
2. Найти все простые числа в списке.
3. Инвертировать строку.
4. Перевести десятичное число в двоичную систему счисления.
5. Найти длину наибольшей последовательности одинаковых чисел идущих подряд.
6. Найти все числа в списке, которые меньше разности двух предыдущих.
7. Составить список, содержащий только положительные числа из исходного списка.
8. Найти комбинации двух положительных чисел, дающие в сумме введённое значение.
9. Найти все числа, квадрат которых попадает в заданный промежуток.
10. Найти сумму произведения двух функций на заданном целочисленном промежутке.
11. Найти определённый интеграл методом трапеций.
12. Найти индекс начала наибольшего интервала увеличивающейся последовательности.
13. Составить список на основе оригинального, удалив оттуда повторяющиеся символы.
14. Доказать бинарную проблему Гольдбаха (проблему Эйлера) - любое чётное число больше двух можно представить в виде суммы двух простых чисел (ввод: чётное число, вывод: два простых числа, сумма которых даёт переданное чётное число).
15. Сократить повторяющиеся элементы в списке

## Лабораторная работа №6

### Коллекции. Функции высших порядков

*Задание к лабораторной работе* - выполнить задания трех частей согласно варианту.

- в части 3 использовать fold;
- в части 4 использовать map/flatMap/filter/reduce.

#### *Часть 1*

1. Из двух несортированных коллекций сделать одну сортированную.
2. Найти среднее арифметическое для всех элементов коллекции.
3. Найти сумму всех положительных чисел в коллекции.
4. Коллекция представляет из себя номерной список студентов, отсортированных по алфавиту. Из двух таких списков сделать один.
5. Подсчитать количество совпадающих символов, стоящих на одинаковых позициях относительно первого элемента, в двух коллекциях.
6. Если первый символ после пробела строчная буква, заменить на прописную.
7. Найти длину самого длинного слова в коллекции.
8. Отсортировать в коллекции все слова по длине.
9. Найти в коллекции индексы всех отрицательных элементов.
10. Найти произведения чисел на главной и побочной диагоналях матрицы.
11. Сложить все элементы списка с использованием анонимной функции.
12. Выполнить переворот списка, каждый элемент умножить на два. Вывести третий элемент полученного списка.
13. Найти минимальный и максимальный элемент списка. Результат вернуть в виде кортежа.
14. Вставить элемент K в каждую N-ую позицию списка.
15. Поменять местами соседние элементы в списке ([0,1,2,3] -> [1,0,3,2]).
16. Из двух коллекций сделать одну, содержащую общие для двух коллекций элементы.
17. Из двух коллекций сделать одну, в которой будут только уникальные элементы.
18. Оставить в коллекции только простые числа.
19. Разделить коллекцию на две. В первой должны быть только положительные элементы, а во второй только отрицательные.
20. В коллекции хранится информация о футболистах и их голах. Создать две коллекции. В одной отсортировать футболистов по голам, а в другой в алфавитном порядке.
21. Существует набор точек. Найти максимальное расстояние между двумя точками среди точек коллекции.
22. Есть две коллекции с целочисленными элементами. Одну из них изменить таким образом, чтобы содержала поэлементную сумму, вторую – поэлементную разность.
23. Перевернуть все слова в коллекции, оставив неизменным их порядок.
24. Удалить из коллекции все повторяющиеся слова.
25. Найти в коллекции палиндромы.
26. Найти в коллекции символов все слова, состоящие только из цифр.
27. Отсортировать коллекцию пузырьком.
28. Если индекс простого числа, хранящегося в коллекции, чётный, умножить его на 10
29. Найти произведение двух матриц.
30. Транспонировать матрицу.

## Часть 2

1. Преобразовать список чисел в список строк.  
IN: List(1, -17)  
OUT: List("1", "-17")
2. Подсчитать длину списка.
3. Реализовать функцию, которая возвращает предпоследний элемент списка.
4. Реализовать функцию, которая принимает список и возвращает новый список с дублированными элементами первоначального.  
IN: List(1, 2, 3)  
OUT: List(1, 1, 2, 2, 3, 4)
5. Реализовать функцию 'flatten', которая конкатенирует список списков.  
IN: List(List(1, 2), List(3, 4))  
OUT: List(1, 2, 3, 4)
6. Реализовать функцию, которая подсчитывает сумму всех элементов в List[Int].
7. Реализовать функцию 'product', которая подсчитывает произведение всех элементов в List[Int].
8. Вернуть последний элемент списка.
9. Подсчитать среднее значение элементов в List[Double].
10. Реализовать функцию 'reverse', которая возвращает перевернутый список
11. Реализовать функцию, которая приводит список из строк в список чисел
12. Найти минимальное значение в списке
13. Реализовать функцию "toSet" которая принимает список, а возвращает Set содержащий уникальные элементы списка.
14. Проверить являются ли все числа в списке простыми
15. Реализовать функцию 'contains', принимающую List[Int] и элемент Int, проверяющую, входит ли переданный элемент в список.

## Часть 3

1. Удалить из списка все строки, не оканчивающиеся на "++".  
IN: List("D", "C++", "F#", "Scala")  
OUT: List("C++")
2. Из списка строк сгенерировать список символов этих строк.  
IN: List("1", "2-3", "4 5")  
OUT: List("1", "2", "-", "3", "4", " ", "5")
3. Преобразовать Map таким образом, чтобы ключи оригинального Map стали значениями преобразованного, и наоборот.  
IN: Map(k1->v1, k2->v2)  
OUT: Map(v1->k1, v2->k2)
4. На основе Map[String,Int] сгенерировать список, состоящий из тех ключей Map, значения которых являются четными.  
IN: Map("A"->2, "B"->3, "C"->5, "D"->6)  
OUT: List("A", "D")
5. Убрать из Map[Int,Int] ключи, которые не равны квадратному корню их значений.  
IN: Map(1->1, 2->4, 3->4, 4->5)  
OUT: Map(1->1, 2->4)
6. Убрать из Map некоторые ключи таким образом, чтобы результирующий Map содержал только уникальные значения.  
IN: Map(1->1, 2->1, 3->2)  
OUT: Map(1->1, 3->2)
7. На основе List[String] сгенерировать List[Int].  
IN: List("1", "01", "-17", "++")  
OUT: List(1, -17)

8. Разбить строку на подстроки с ',' в качестве разделителя.  
IN: "Trust me, I'm a programmer"  
OUT: List("Trust me", " I'm a programmer")
9. На основе List[String] сгенерировать строку, сконкатенированную через ' ' из тех подстрок List, которые начинаются с заглавной буквы.  
IN: List("Hey!", "I am a", "lazy but", "Clever", "Guy!")  
OUT: "Hey! I am a Clever Guy!"
10. Используя числа из List[Int] сгенерировать список кубов этих чисел.  
IN: List(1, 2, 3)  
OUT: List(1, 8, 27)
11. Найти строку с максимальной длиной в List[String].  
IN: List("I", "You", "We")  
OUT: "You"
12. Из данного List[Int] убрать все числа, не входящие в последовательность чисел Фибоначчи.  
IN: List(1, 2, 4, 5, 8, 13, 27, 55)  
OUT: List(1, 2, 5, 13, 55)
13. Найти сумму элементов трехмерной матрицы.  
IN: List(List(List(0, 1), List(2, 3)), List(List(4, 5), List(6, 7)))  
OUT: 28
14. Найти произведения элементов четных строк матрицы.  
IN: List(List(1, 2, 3), List(4, 5, 6), List(7, 8, 0))  
OUT: List(6, 0)
15. Отфильтровать последовательность, оставив только элементы типа String.  
IN: List("One", 2, 3, "Four", "Five")  
OUT: List("One", "Four", "Five")