Министерство образования Республики Беларусь

Учреждение высшего образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе №1

на тему

**РАБОТА СО СПИСКАМИ И ФУНКЦИЯМИ**

|  |  |
| --- | --- |
| Студенты группы 150503:  Проверила: | Ходосевич М.А.  Семков А. Д.  Герман Ю. О. |

Минск 2023

Цель: изучить технику работы со списками и функциями в Scala.

1. **Краткие теоретические сведения:**

Примеры функций для работы со списками в Scala из методички:

В первом примере выполнения перебор всех элементов с помощью Map и умножение каждого элемента на 2. Кода данной программы приведен ниже, а выполнение данного кода на рисунке 1.

object Main22 {

def double(x: Int): Int = x \* 2

def main(args: Array[String]): Unit = {

val myList = List(1, 2, 3, 4, 5)

val doubledList = myList.map(double)

println(doubledList)

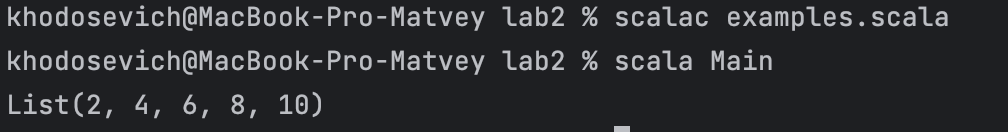


рисунок 1 - пример выполнения примера

Также продемонстрирован по элементный вывод List.

Пример выполнения данного кода приведен на рисунке 2.

object Main22 {

def double(x: Int): Int = x \* 2

def main(args: Array[String]): Unit = {

val myList = List(1, 2, 3, 4, 5)

val doubledList = myList.map(double)

println(myList.mkString(", "))

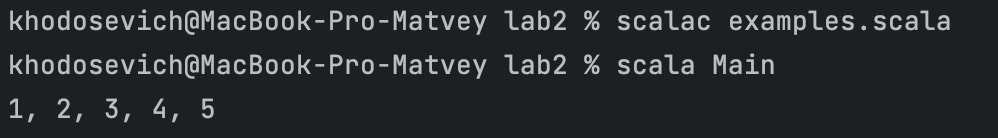


рисунок 2 - пример выполнения примера

Далее у нас рассматривается метод filter - эта функция отбирает элементы списка, удовлетворяющие заданному предикату.

object Main22 {

def isEven(x: Int): Boolean = x % 2 == 0

def main(args: Array[String]): Unit = {

val myList = List(1, 2, 3, 4, 5)

val filteredList = myList.filter(isEven)

println(filteredList.mkString(", "))

Пример выполнения данного кода приведен на рисунке 3.

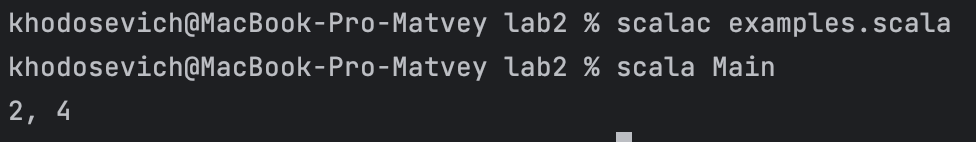


рисунок 3 - пример выполнения примера

FoldLeft – эта функция последовательно применяется к элементам списка слева направо, накапливая результат. Сумму элементов списка можно найти таким образом

object Main22 {

def main(args: Array[String]): Unit = {

val myList = List(1, 2, 3, 4, 5)

val sum = myList.foldLeft(0)((ac\_c, x) => ac\_c + x)

println(sum) // Output: 15

Здесь переменная ac\_c играет роль аккумулятора. Первоначально ей присваивается значение 0:

myList.foldLeft(0)

Пример выполнения данного кода приведен на рисунке 4.

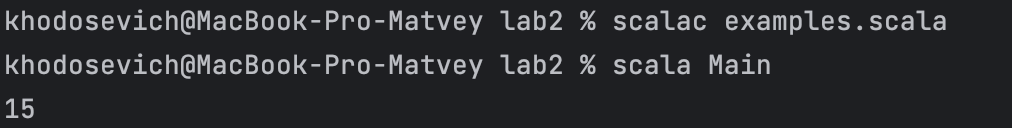


рисунок 4 - пример выполнения примера

Zip - эта функция объединяет два списка на примере словаря (dictionary) – ключ-значение.

val a = List(1, 2, 3)

val b = List("one", "two", "three")

val zipped = a.zip(b)

println(zipped)

Пример выполнения данного кода приведен на рисунке 5.

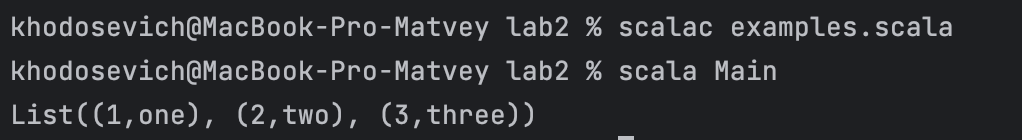


рисунок 5 - пример выполнения примера

Head and tail - эти функции возвращают голову и хвост списка соответственно.

val numbers = List(1, 2, 3, 4, 5)

val first = numbers.head

val rest = numbers.tail

Пример выполнения данного кода приведен на рисунке 6.

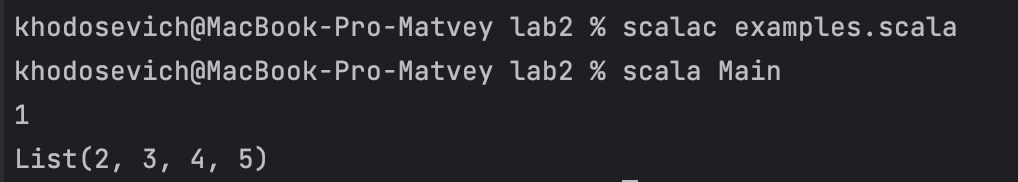


рисунок 6 - пример выполнения примера

Сумма квадратов элементов списка:

object Main22 {

def sumList(lst: List[Int]): Int = {

if (lst.isEmpty) 0

else lst.head\*lst.head + sumList(lst.tail)

}

def main(args: Array[String]): Unit = {

val myList = List(1, 2, 3, 4, 5)

val sumw = sumList(myList)

println(sumw)

Пример выполнения данного кода приведен на рисунке 7.

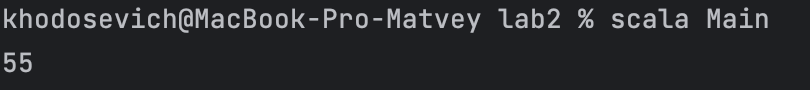


рисунок 7 - пример выполнения примера

Reverse – эта функция возвращает список в обратном порядке.

val numbers = List(1, 2, 3, 4, 5)

val reversed = numbers.reverse

Пример выполнения данного кода приведен на рисунке 8.

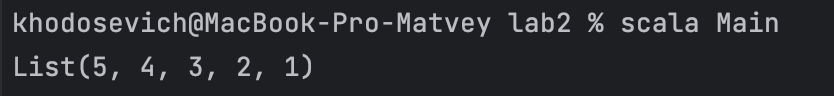


рисунок 8 - пример выполнения примера

**2. Ход работы:**

**Вариант 4.**

1. Написать функцию для подсчета суммы элементов списка, значение которых по модулю не превосходит 5. Список задать самостоятельно.
2. Написать функцию для подсчета суммы каждого второго элемента списка из 10 элементов. Список задать самостоятельно.
3. Написать функцию для отыскания индекса элемента списка, наименее отклоняющегося от среднего значения по списку. Список задать самостоятельно.
4. Написать функцию для проверки того, что список содержит квадрат одного из своих элементов. Список задать самостоятельно.
5. Написать функцию для проверки наличия трех разных элементов в списке. Список задать самостоятельно. Функция возвращает значение Да или Нет
6. **Вывод:**

В ходе выполнения данной лабораторной работы нами были проанализированы и изучены техники работы со списками и функциями в Scala.