Отчет по лабораторной работе №2

Компьютерный практикум по статистическому анализу данных

Амуничников Антон Игоревич

Содержание

# 1. Цель работы

Основная цель работы – изучить несколько структур данных, реализованных в Julia, научиться применять их и операции над ними для решения задач.

# 2. Задание

1. Используя Jupyter Lab, повторите примеры.
2. Выполните задания для самостоятельной работы.

# 3. Теоретическое введение

Julia – высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений [**julialang?**]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia [**juliadoc?**].

Рассмотрим несколько структур данных, реализованных в Julia. Несколько функций (методов), общих для всех структур данных:

* isempty() – проверяет, пуста ли структура данных;
* length() – возвращает длину структуры данных;
* in() – проверяет принадлежность элемента к структуре;
* unique() – возвращает коллекцию уникальных элементов структуры,
* reduce() – свёртывает структуру данных в соответствии с заданным бинарным оператором;
* maximum() (или minimum()) – возвращает наибольший (или наименьший) результат вызова функции для каждого элемента структуры данных.

# 4. Выполнение лабораторной работы

Для начала выполним примеры из раздела про кортежи ([рис. 1](#fig-001)). Кортеж (Tuple) – структура данных (контейнер) в виде неизменяемой индексируемой последовательности элементов какого-либо типа (элементы индексируются с единицы).

|  |
| --- |
| Рисунок 1: Примеры использования кортежей |

Теперь выполним примеры из раздела про словари ([рис. 2](#fig-002)). Словарь – неупорядоченный набор связанных между собой по ключу данных.

|  |
| --- |
| Рисунок 2: Примеры использования словарей |

Выполним примеры из раздела про множества ([рис. 3](#fig-003)). Множество, как структура данных в Julia, соответствует множеству, как математическому объекту, то есть является неупорядоченной совокупностью элементов какого-либо типа. Возможные операции над множествами: объединение, пересечение, разность; принадлежность элемента множеству.

|  |
| --- |
| Рисунок 3: Примеры использования множеств |

Выполним примеры из раздела про массивы ([рис. 4](#fig-004) - [рис. 8](#fig-008)). Массив — коллекция упорядоченных элементов, размещённая в многомерной сетке. Векторы и матрицы являются частными случаями массивов.

|  |
| --- |
| Рисунок 4: Примеры использования массивов |

|  |
| --- |
| Рисунок 5: Примеры использования массивов |

|  |
| --- |
| Рисунок 6: Примеры использования массивов |

|  |
| --- |
| Рисунок 7: Примеры использования массивов |

|  |
| --- |
| Рисунок 8: Примеры использования массивов |

Теперь перейдем к выполнению заданий.

**Задание №1**

Даны множества: . Найдем ([рис. 9](#fig-009)).

|  |
| --- |
| Рисунок 9: Задание №1. Работа с множествами |

**Задание №2**

Приведем свои примеры с выполнением операций над множествами элементов разных типов ([рис. 10](#fig-010)).

|  |
| --- |
| Рисунок 10: Задание №2. Примеры операций над множествами элементов разных типов |

**Задание №3**

Создадим массивы разными способами, используя циклы ([рис. 11](#fig-011) - [рис. 17](#fig-017)).

|  |
| --- |
| Рисунок 11: Задание №3. Работа с массивами |

|  |
| --- |
| Рисунок 12: Задание №3. Работа с массивами |

|  |
| --- |
| Рисунок 13: Задание №3. Работа с массивами |

|  |
| --- |
| Рисунок 14: Задание №3. Работа с массивами |

|  |
| --- |
| Рисунок 15: Задание №3. Работа с массивами |

|  |
| --- |
| Рисунок 16: Задание №3. Работа с массивами |

|  |
| --- |
| Рисунок 17: Задание №3. Работа с массивами |

**Задание №4**

Создадим массив squares, в котором будут храниться квадраты всех целых чисел от 1 до 100 ([рис. 18](#fig-018)).

|  |
| --- |
| Рисунок 18: Задание №4 |

**Задание №5**

Подключим пакет Primes (функции для вычисления простых чисел). Сгенерируем массив myprimes, в котором будут храниться первые 168 простых чисел. Определим 89-е наименьшее простое число. Получии срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа ([рис. 19](#fig-019)).

|  |
| --- |
| Рисунок 19: Задание №5. Работа с пакетом Primes |

**Задание №6**

Вычислим следующие выражения ([рис. 20](#fig-020)).

|  |
| --- |
| Рисунок 20: Задание №6 |

#Выводы

В результате выполнения данной лабораторной работы я изучил несколько структур дыннх, реализованных в Julia, Научился применять их и операции над ними для решения задач.