

Министерство образования Республики Беларусь

Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

Кафедра интеллектуальных информационных технологий

Отчет

По дисциплине: Проектирование программного  
обеспечения интеллектуальных систем  
Лабораторная работа №1

Выполнили:  
Смольник В.А.  
гр. 121701

Проверил:  
Бутрин С.В.

Минск 2022

**Условие:** Древовидное представление и интерпретация теоретико-множественных выражений

При выборе этого варианта студенту необходимо разработать объектную модель для представления теоретико-множественных выражений в виде дерева и реализовать программу-калькулятор,

которая осуществляет вычисления выражений такого типа.

Рассмотрение этого варианта начнем с примера работы программы (бирюзовым цветом выделен ввод от пользователя, черным цветом – вывод программы, символ «?» обозначает нажатие клавиши Enter):

```
? {a, b, c} ?  
= {a, b, c}  
? {a, b, c} + {a, b, d} ?  
= {a, b, c, d}  
? {a, b, c} * {a, b, d} ?  
= {a, b}  
? B = A = {a, b, c} ?  
= {a, b, c}  
? B - A ?  
= {}  
? {a} + {} * ( ( {} + {a, {a, b}} ) ) ?  
= {a, {a, b}}
```

Из приведенной сессии работы можно сформулировать возможности калькулятора:

- Вычисление теоретико-множественных выражений с использованием операций объединения (+), пересечения (\*), разности множества, симметрической разности (-) множеств. При этом приоритет этих

операций должен задаваться при помощи скобок.

- Возможность использования заранее определенных функций обработки множества (size({}) - размер множества без вложений, rpsr({}) - размер множества со вложениями, A[i] - получить подмножество множества).

- Возможность установки результата вычисления выражения в качестве значения переменных.

- Указания места ошибки и сообщения об ошибке при некорректном вводе.

Реализация:

В программе я создал 4 класса: Read\_a\_Line (прочитать строку и определить, к какому классу будет в дальнейшем относиться строка). Sets\_and\_operations (в котором описывается основная логика парсинга строки, представления строки с множеством в виде множества (класс Set), а также составление списка с приоритетами выполнения операций над множествами), класс Constants (отвечает за создание констант:  $A = \{a, b\}$ , а также их использование в операциях над множествами). В классе Another\_operations определяются такие операции, как  $A[i]$  - получить элемент/подмножество множества,  $size(A)$  - получить размер множества без учета подмножеств,  $psp(A)$  - получить размер множества с учетом размеров всех подмножеств.

class Set - класс множества, который имеет два поля: elements (элементы множества) и subsets (подмножества). Также переопределены методы сложения, умножения, вычитания и деления объектов множества Set.

Программа обернута в обработчик ошибок try/except, помимо этого на протяжении всего кода идет валидация данных. В случае какой-либо ошибки или неправильно введенной строки, вследствие чего выполняются неверные операции, ошибка выбрасывается на внешний обработчик ошибок, где и обрабатывается.

Сохраненные данные хранятся в json формате.