

**Министерство образования Республики Беларусь**

**Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»**

---

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ**

**Кафедра интеллектуальных информационных технологий**

**Отчёт по лабораторной работе №1  
по курсу «ЕЯзИИС» на тему:  
«Разработка автоматизированной системы формирования  
словаря естественного языка»**

Выполнил студент группы 921703:      Кравцов Михаил Сергеевич

Проверил:      Крапивин Юрий Борисович

**Минск 20.04.2022**

1. Условие .....	3
1.1. Цель .....	3
1.2. Постановка задачи.....	3
1.3. Задание .....	3
2. Схема программы .....	4
3. Оценка быстродействия .....	5
4. Используемые библиотеки и разработанные классы .....	6
Выводы.....	7

# **1. Условие**

## **1.1. Цель**

Освоить принципы разработки прикладных сервисных программ для решения задачи автоматического лексического и лексико-грамматического анализа текста естественного языка.

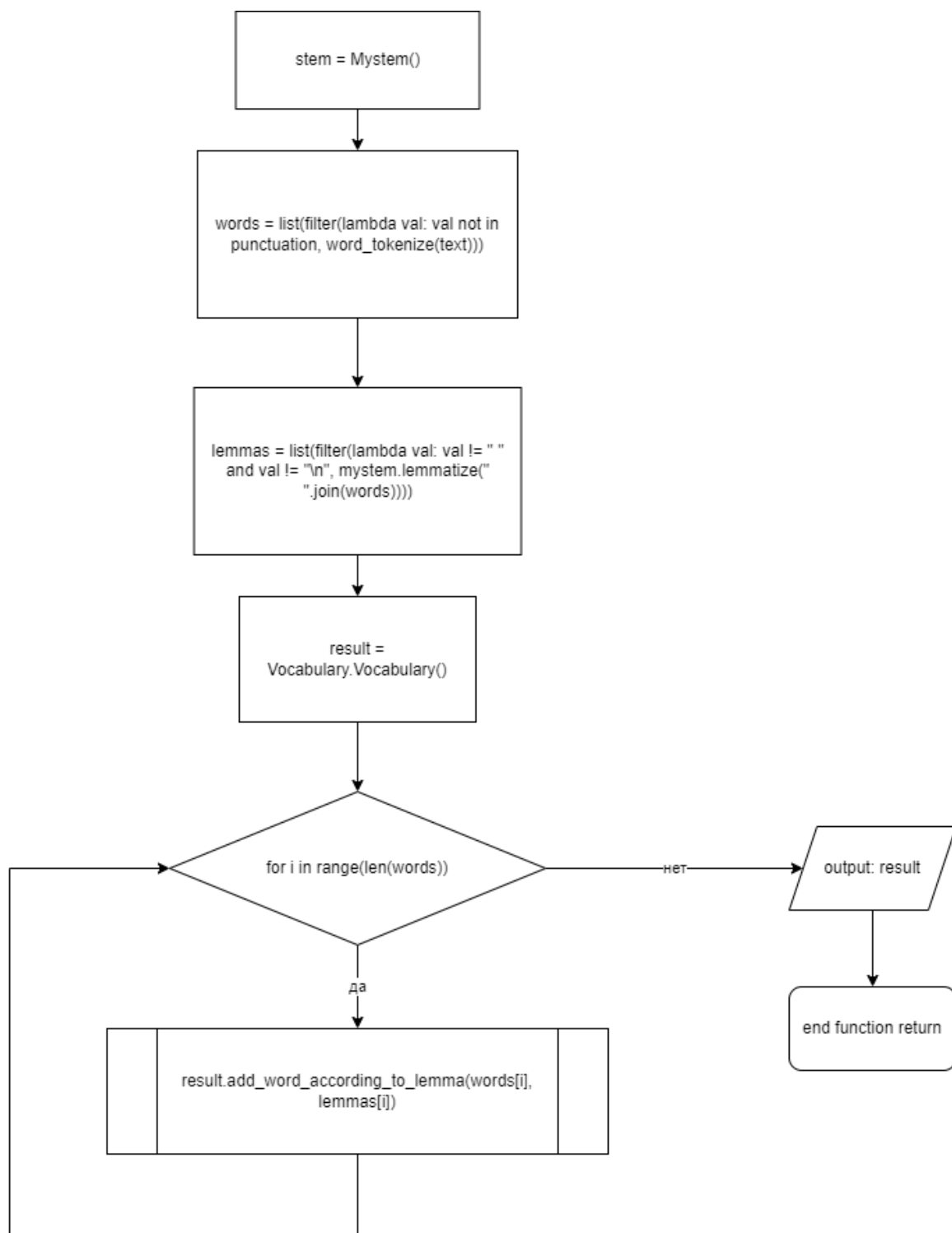
## **1.2. Постановка задачи**

1. Познакомиться с назначением, структурой и функциональностью, предоставляемой базовым ЛП для решения задачи автоматического лексического и лексико-грамматического анализа ТЕЯ.
2. Закрепить навыки программирования при решении задач автоматической обработки ТЕЯ.

## **1.3. Задание**

**Задание 1.** Список слов, упорядоченный по алфавиту и включающий как лексемы, так и словоформы, с указанием частоты встречаемости каждой из форм. Для словоформ пользователю должна быть предоставлена возможность вводить дополнительную морфологическую информацию, а именно, отнесение слова к соответствующей части речи, указание рода, числа, падежа и т.п. При этом морфологическая информация может быть оформлена как отдельная неформатированная запись, т.е. это просто текст, который пользователь может оформлять произвольным образом.

## 2. Схема программы



### 3. Оценка быстродействия

Быстродействие программы, как известно, зависит от сложности применённого алгоритма. Поэтому подсчитаем сложность алгоритма используя нотацию  $O()$ .

Изначально создаем объект класса `Mystem` библиотеки `rumystem3`, который в последствии будет применен. Далее составляем множество слов и множество лексем с использованием `nltk.word_tokenizer` объекта `mystem.lemmatize`. После отфильтровываем ненужные значения (пунктуацию, пробелы).

Далее идет цикл где все лексем и словоформы добавляются в словарь (разработанный класс) с помощью метода `add_word_according_to_lemma`.

Сложность составляет  $O(n*m*p)$ , где  $n$  - кол-во входных токенов в тексте,  $m$  — кол-во подходящих токенов,  $p$  - кол-во лексем.

#### **4. Используемые библиотеки и разработанные классы**

При выполнении поставленных задач были использованы следующие библиотеки:

- 1) Tkinter
- 2) re
- 3) nltk
- 4) docx
- 5) pickle
- 6) string
- 7) pymystem3

Были разработаны классы Vocabulary, App, Controller, Table, View, также функции для обработки engine.

Код можно посмотреть по ссылке - [https://github.com/mixa1334/laba1\\_eyazis\\_6sem](https://github.com/mixa1334/laba1_eyazis_6sem)

## **Выводы**

В данной лабораторной работе было создано и протестировано приложение для морфологического анализа текста естественного языка (русский). Приложение получилось простое в использовании и довольно функциональным, что дает ему преимущества.