# Санкт-Петербургский государственный политехнический университет

Кафедра “Прикладная математика и информатика”

Отчет по лабораторной работе 4 (вариант 10)

“Алгоритмы и структуры данных”

Студент группы № 5030102/20001

ФИО: Дрекалов Никита Сергеевич

Выполнил 15.11.2023

Оглавление

[Санкт-Петербургский государственный политехнический университет 1](#_Toc152252582)

[**Постановка задачи** 3](#_Toc152252583)

[**Описание алгоритма** 3](#_Toc152252584)

[Структура экспертной системы и поиска 3](#_Toc152252585)

[Добавление новых ответов 3](#_Toc152252586)

[**Текст программы** 4](#_Toc152252587)

[**Описание тестирования** 4](#_Toc152252588)

[Тест 1. Создание экспертной системы и её сохранение 5](#_Toc152252589)

[Тест 2. Загрузка системы из файла и проверка её работы 6](#_Toc152252590)

# Постановка задачи

В лабораторной работе требуется написать самообучающуюся экспертную систему: пользователь загадывает n, а программа должна с помощью простых вопросов (на которые ответ «Да/Нет») отгадать это n. При этом, если не получилось отгадать, следует предложить пользователю добавить информацию об n в экспертную систему.

Пользователь программы может делать следующие запросы:

* Запустить экспертную систему.
* Сохранить экспертную систему в файл.
* Загрузить экспертную систему из файла.
* Выйти из программы.
* Вывести список команд.

# Описание алгоритма

### Структура экспертной системы и поиска

1. Экспертная система представляет бинарное дерево, во внутренних узлах которого находятся вопросы, а в листьях – ответы.
2. При ответе «Да» будет происходить переход в левое поддерево, при «Нет» - в правое.

### Добавление новых ответов

1. Если экспертная система не знает ответ на вопрос (т. е. пользователь ответил «Нет», но у узла нет потомка справа), то пользователю будет предложено добавить свой ответ. Новый ответ будет в «Нет» потомке у узла вопроса.
2. Если экспертная система дала неправильный ответ (т. е. пользователь ответил «Нет» на элемент в листьях), то пользователю также будет предложено добавить свой ответ вместе с его уникальной характеристикой, но при этом уникальная характеристика будет на месте старого ответа, старый ответ – в «Нет» потомке характеристики, а новый ответ – в «Да» потомке (сделано это так для того, чтобы листья были все в среднем на одной глубине, другими словами, чтобы поддеревья «Да» тоже могли дальше «расти»).

# Текст программы

Проект и код можно найти и посмотреть тут: [***GitHub***](https://github.com/Forthey/algorithms_ds/tree/main/task4_10)

# Описание тестирования

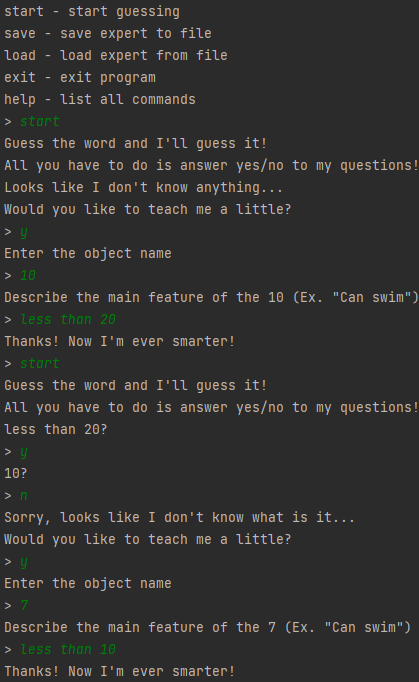
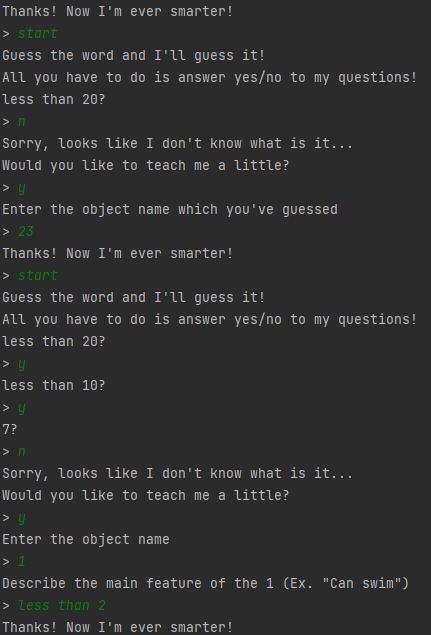
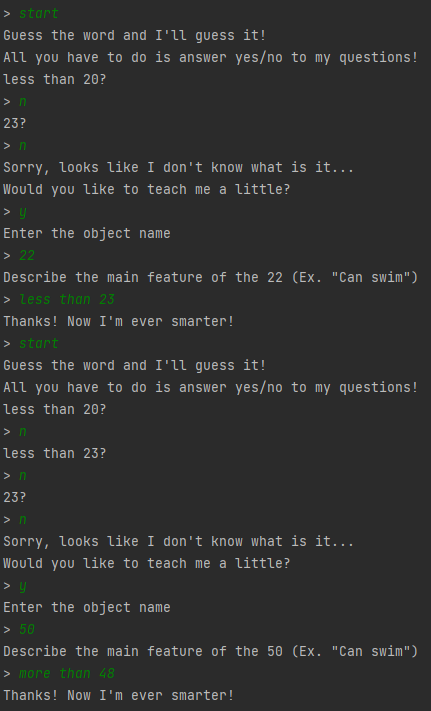
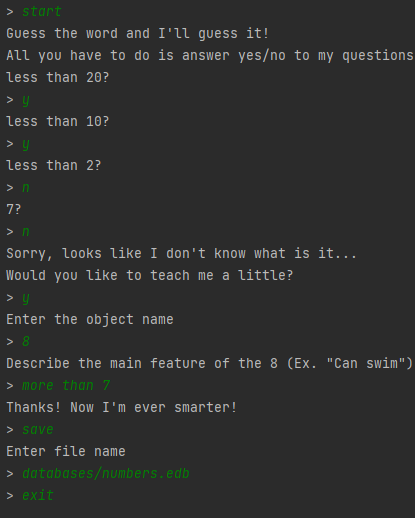
Тестирование будет происходить на экспертной системе, которая будет угадывать число (так как её просто делать…). Сначала будет в программе построена эта система, затем сохранена в файл. При следующем запуске система будет загружена из файла и будет произведено несколько тестовых угадываний, чтобы проверить, что система загрузилась и работает корректно.

Проверялось:

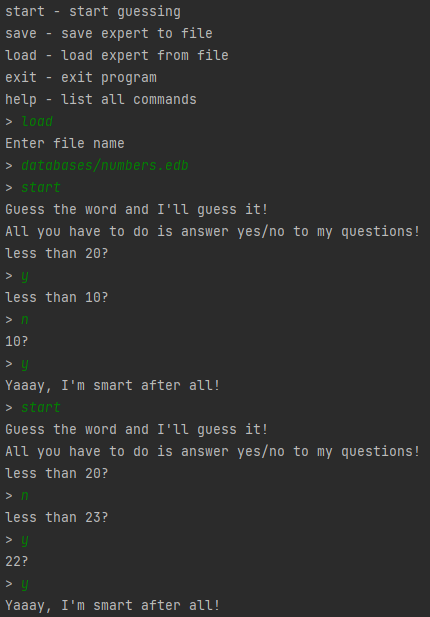
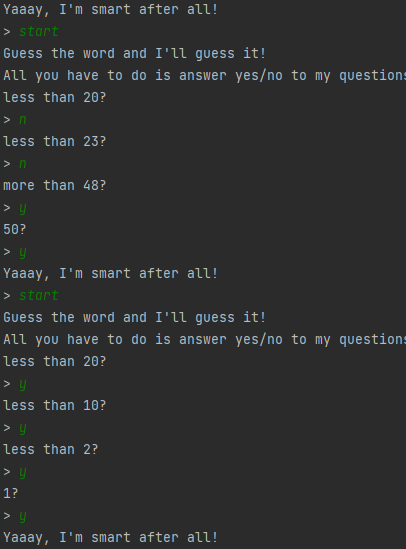
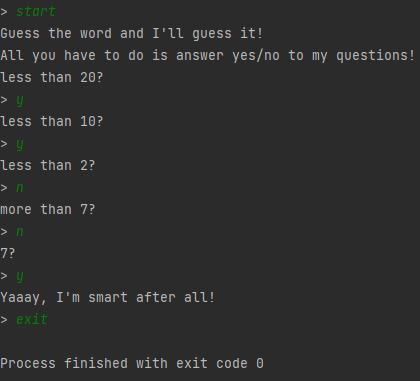
* Стабильность работы программы при одинаковых входных данных;
* Отсутствие “падений” и “зависаний”;
* Корректное выполнение всех заявленных процедур;
* Корректное завершение программы;

Приведем пример тестирования на скриншотах, данных ниже

### Тест 1. Создание экспертной системы и её сохранение

### Тест 2. Загрузка системы из файла и проверка её работы

При проведении тестирования такого рода никаких проблем обнаружено не было, что позволяет судить о корректности работы программы в целом.