Федеральное государственное автономное учебное учреждение высшего образования «Национальный исследовательский университет ИТМО»

«Мегафакультет компьютерных технологий и управления Факультет программной инженерии и компьютерной техники

**Отчет**

по Модулю №1

«Базы знаний и онтологии»

по дисциплине

«Системы искусственного интеллекта

Группа: P3318

Студент: Рыскаль Н.А.

Преподаватель: Авдюшина А.Н.

Санкт-Петербург, 2024

# Введение

Цель исследования состоит в разработке и изучении трех ключевых компонентов в области искусственного интеллекта: базы знаний, онтологии и системы поддержки принятия решений на основе этих знаний. Результат работы- программа, которая помогает подобрать лидера для игры компьютерной игры Civilization VI.

Значимость первой лабораторной работы определяется созданием базы знаний и онтологии классов лидеров с помощью Prolog и Protege. База знаний может использоваться для разработки системы поддержки принятия решения.

Значимость второй лабораторной работы определяется разработкой программы, которая будет использовать базу знаний для предоставления рекомендаций на основе введенных пользователем данных

# Анализ требований

## Требования к базе знаний и онтологии

Требования к базе знаний и онтологиям определяются необходимостью хранения корректной, структурированной информации для заданной предметной области. Необходимо продумать четкую иерархию сущностей. Говоря о базе знаний, нужно выделить четкие факты, предикаты, правила. Работая с онтологией, необходимо четко определить сущности, свойства.

## Требования к системе поддержи принятия решений

1. Модульность и гибкость - система должна позволять расширять предметную область и добавлять новые опции с минимальным количеством вносимых изменений
2. Точность - система должна предоставлять пользователю ответ, полностью со ответствующий его запросу;
3. Полнота функционала- система должна поддерживать ввод запросов любой сложности в рамках предметной области;
4. Удобство использования- система должна предоставлять пользователю чет кие и понятные инструкции по пользованию;
5. Устойчивость к ошибкам ввода- система должна выводить корректные сообщения об ошибках ввода.

# Изучение основных концепций и инструментов

Базы знаний и онтологии — это два понятия в области искусственного интеллекта и компьютерных наук. Обе концепции используются для описания знаний и информации, но имеют разные подходы к этому.

Базы знаний — это системы, которые хранят информацию и знания, которые могут быть использованы для решения задач. Они обычно содержат факты, правила, процедуры и другие элементы, необходимые для решения задач. Базы знаний используются в различных областях, таких как медицина, финансы, право и т.д. Основными принципами баз знаний являются декомпозиция знаний на факты, правила и процедуры; построение моделей знаний на основе логических и математических принципов; использование методов интеллектуального анализа данных для решения задач.

Онтологии — это формальные модели, которые описывают концептуальные категории и отношения между ними. Они используются для описания знаний в языке, который может быть понят людьми и машинами. Онтологии используются для создания более точных и эффективных систем поиска, классификации и анализа данных Основными принципами онтологий являются описание концептуальных категорий и отношений между ними; использование формальных языков для описания знаний; создание стандартных моделей для обмена и интеграции данных. Обе концепции имеют свои преимущества и недостатки, и их выбор зависит от конкретных задач и потребностей. Базы знаний используются для решения конкретных задач, а онтологии- для описания общих концептуальных категорий и создания стандартных моделей для обмена данными

## Логический язык программирования Prolog

Prolog — это язык программирования общего назначения. Он обладает возможностями для создания эффективных алгоритмов и моделей машинного обучения. Некоторые из наиболее полезных функций и возможностей Prolog:

1. Логический вывод: Prolog поддерживает логические операции, такие как AND, OR, NOT, EQUALS и LT, GT, LE, BETWEEN и IN, которые могут использоваться для выполнения различных типов логических операций. Это позволяет создавать эффективные модели логического вывода и обработки естественно го языка.
2. Моделирование. Prolog предоставляет мощные инструменты для создания и манипулирования моделями машинного обучения. Он поддерживает различные типы моделей, включая линейную регрессию, деревья решений, нейронные сети и многое другое. Prolog также поддерживает возможность создания собственных моделей с помощью пакетов, таких как PL/SQL и Symbolic Expression Language (SEL).
3. Алгоритмическое программирование. Prolog поддерживает различные типы алгоритмического программирования, такие как задачи коммивояжера, сортировка слиянием и быстрая сортировка. Это позволяет создавать эффективные алгоритмы для решения задач, которые требуют сложных структур данных и вычислительных ресурсов.
4. Интеграция с другими языками программирования. Prolog поддерживает интеграцию с различными языками программирования, такими как Python, Java и Go. Это позволяет разработчикам интегрировать свои модели машинного обучения и алгоритмы в другие проекты и среды разработки. Prolog предлагает мощный и гибкий инструмент для разработки систем искусственного интеллекта. Он обеспечивает высокую производительность, эффективность и удобство использования, а также поддерживает широкий спектр функциональности, которая может быть использована для создания эффективных моделей машинного обучения и обработки естественного языка.

# Реализация системы искусственного интеллекта (системы поддержки принятия решений)

## Разработанная база знаний:

/\*

leader - факт описывающий лидеров представленных в игре

\*/

leader('Abraham Lincoln').

leader('Teddy Roosevelt').

leader('Saladin').

leader('Basil II').

leader('Theodora').

leader('Qin Shi Huang').

leader('Wu Zetian').

leader('Yongle').

leader('Cleopatra').

leader('Ramses II').

leader('Elizabeth I').

leader('Victoria').

leader('Frederick Barbarossa').

leader('Ludwig II').

leader('Peter').

leader('Tomyris').

leader('Philip II').

leader('Gilgamesh').

leader('Kristina').

leader('Shaka').

/\*

civilization - факт описывающий во главе какой нации стоит каждый из лидеров

\*/

civilization('Abraham Lincoln', 'American').

civilization('Teddy Roosevelt', 'American').

civilization('Saladin', 'Arabian').

civilization('Basil II', 'Byzantine').

civilization('Theodora', 'Byzantine').

civilization('Qin Shi Huang', 'Chinese').

civilization('Wu Zetian', 'Chinese').

civilization('Yongle', 'Chinese').

civilization('Cleopatra', 'Egyptian').

civilization('Ramses II', 'Egyptian').

civilization('Elizabeth I', 'English').

civilization('Victoria', 'English').

civilization('Frederick Barbarossa', 'German').

civilization('Ludwig II', 'German').

civilization('Peter', 'Russian').

civilization('Tomyris', 'Scythian').

civilization('Philip II', 'Spanish').

civilization('Gilgamesh', 'Sumerian').

civilization('Kristina', 'Swedish').

civilization('Shaka', 'Zulu').

/\*

location - факт описывающий на каком материке расположина нация

\*/

location('American', 'America').

location('Arabian', 'Asia').

location('Byzantine', 'Asia').

location('Chinese', 'Asia').

location('Egyptian', 'Africa').

location('English', 'Europe').

location('German', 'Europe').

location('Russian', 'Europe').

location('Scythian', 'Asia').

location('Spanish', 'Europe').

location('Sumerian', 'Asia').

location('Swedish', 'Europe').

location('Zulu', 'Africa').

/\*

strength - факт описывающий аспекты лидера, где 1 - он плох, 5 - крутой

задется как strength(leader, аспек, сила аспекта)

\*/

strength('Abraham Lincoln', 'Science', 3).

strength('Abraham Lincoln', 'Diplomacy', 4).

strength('Abraham Lincoln', 'War', 5).

strength('Abraham Lincoln', 'Culture', 2).

strength('Abraham Lincoln', 'Faith', 1).

strength('Teddy Roosevelt', 'Science', 4).

strength('Teddy Roosevelt', 'Diplomacy', 5).

strength('Teddy Roosevelt', 'War', 3).

strength('Teddy Roosevelt', 'Culture', 2).

strength('Teddy Roosevelt', 'Faith', 2).

strength('Saladin', 'Science', 2).

strength('Saladin', 'Diplomacy', 3).

strength('Saladin', 'War', 5).

strength('Saladin', 'Culture', 3).

strength('Saladin', 'Faith', 5).

strength('Basil II', 'Science', 2).

strength('Basil II', 'Diplomacy', 3).

strength('Basil II', 'War', 5).

strength('Basil II', 'Culture', 4).

strength('Basil II', 'Faith', 4).

strength('Theodora', 'Science', 3).

strength('Theodora', 'Diplomacy', 4).

strength('Theodora', 'War', 4).

strength('Theodora', 'Culture', 5).

strength('Theodora', 'Faith', 5).

strength('Qin Shi Huang', 'Science', 5).

strength('Qin Shi Huang', 'Diplomacy', 2).

strength('Qin Shi Huang', 'War', 4).

strength('Qin Shi Huang', 'Culture', 3).

strength('Qin Shi Huang', 'Faith', 1).

strength('Wu Zetian', 'Science', 4).

strength('Wu Zetian', 'Diplomacy', 3).

strength('Wu Zetian', 'War', 4).

strength('Wu Zetian', 'Culture', 5).

strength('Wu Zetian', 'Faith', 2).

strength('Yongle', 'Science', 5).

strength('Yongle', 'Diplomacy', 4).

strength('Yongle', 'War', 3).

strength('Yongle', 'Culture', 4).

strength('Yongle', 'Faith', 2).

strength('Cleopatra', 'Science', 3).

strength('Cleopatra', 'Diplomacy', 4).

strength('Cleopatra', 'War', 3).

strength('Cleopatra', 'Culture', 5).

strength('Cleopatra', 'Faith', 3).

strength('Ramses II', 'Science', 2).

strength('Ramses II', 'Diplomacy', 3).

strength('Ramses II', 'War', 5).

strength('Ramses II', 'Culture', 4).

strength('Ramses II', 'Faith', 4).

strength('Elizabeth I', 'Science', 3).

strength('Elizabeth I', 'Diplomacy', 5).

strength('Elizabeth I', 'War', 4).

strength('Elizabeth I', 'Culture', 4).

strength('Elizabeth I', 'Faith', 2).

strength('Victoria', 'Science', 4).

strength('Victoria', 'Diplomacy', 5).

strength('Victoria', 'War', 3).

strength('Victoria', 'Culture', 3).

strength('Victoria', 'Faith', 2).

strength('Frederick Barbarossa', 'Science', 3).

strength('Frederick Barbarossa', 'Diplomacy', 2).

strength('Frederick Barbarossa', 'War', 5).

strength('Frederick Barbarossa', 'Culture', 3).

strength('Frederick Barbarossa', 'Faith', 1).

strength('Ludwig II', 'Science', 4).

strength('Ludwig II', 'Diplomacy', 3).

strength('Ludwig II', 'War', 3).

strength('Ludwig II', 'Culture', 5).

strength('Ludwig II', 'Faith', 2).

strength('Peter', 'Science', 5).

strength('Peter', 'Diplomacy', 4).

strength('Peter', 'War', 3).

strength('Peter', 'Culture', 4).

strength('Peter', 'Faith', 2).

strength('Tomyris', 'Science', 2).

strength('Tomyris', 'Diplomacy', 2).

strength('Tomyris', 'War', 5).

strength('Tomyris', 'Culture', 3).

strength('Tomyris', 'Faith', 1).

strength('Philip II', 'Science', 3).

strength('Philip II', 'Diplomacy', 4).

strength('Philip II', 'War', 4).

strength('Philip II', 'Culture', 3).

strength('Philip II', 'Faith', 5).

strength('Gilgamesh', 'Science', 5).

strength('Gilgamesh', 'Diplomacy', 3).

strength('Gilgamesh', 'War', 5).

strength('Gilgamesh', 'Culture', 3).

strength('Gilgamesh', 'Faith', 2).

strength('Kristina', 'Science', 4).

strength('Kristina', 'Diplomacy', 5).

strength('Kristina', 'War', 2).

strength('Kristina', 'Culture', 5).

strength('Kristina', 'Faith', 3).

strength('Shaka', 'Science', 2).

strength('Shaka', 'Diplomacy', 2).

strength('Shaka', 'War', 5).

strength('Shaka', 'Culture', 2).

strength('Shaka', 'Faith', 1).

/\*

difficulty - факт описывающий сложность лидера

difficulty\_value - это факт описывающий сложность, вторым термом вводится значение сложности, где 1 - легкий, 3-сложный

\*/

difficulty('Abraham Lincoln', 'Medium').

difficulty('Teddy Roosevelt', 'Easy').

difficulty('Saladin', 'Hard').

difficulty('Basil II', 'Hard').

difficulty('Theodora', 'Medium').

difficulty('Qin Shi Huang', 'Easy').

difficulty('Wu Zetian', 'Medium').

difficulty('Yongle', 'Medium').

difficulty('Cleopatra', 'Medium').

difficulty('Ramses II', 'Hard').

difficulty('Elizabeth I', 'Medium').

difficulty('Victoria', 'Easy').

difficulty('Frederick Barbarossa', 'Hard').

difficulty('Ludwig II', 'Medium').

difficulty('Peter', 'Medium').

difficulty('Tomyris', 'Hard').

difficulty('Philip II', 'Easy').

difficulty('Gilgamesh', 'Medium').

difficulty('Kristina', 'Easy').

difficulty('Shaka', 'Hard').

difficulty\_value('Easy', 1).

difficulty\_value('Medium', 2).

difficulty\_value('Hard', 3).

/\*

поиск континета по лидеру

\*/

leader\_location(Leader, Location) :-

civilization(Leader, Nation),

location(Nation, Location).

/\*

поиск воинствующих лидеров

\*/

strong\_war\_leader(Leader) :-

strength(Leader, 'War', Value),

Value >= 4.

/\*

поиск научных лидеров, с маленькой верой

\*/

science\_focused\_leader(Leader) :-

strength(Leader, 'Science', ScienceValue),

strength(Leader, 'Faith', FaithValue),

ScienceValue >= 4,

FaithValue < 3.

/\*

поиск культурных лидеров

\*/

cultural\_leader(Leader) :-

strength(Leader, 'Culture', Value),

Value >= 4.

/\*

поиск дипломатичных лидеров

\*/

diplomatic\_leader(Leader) :-

strength(Leader, 'Diplomacy', Value),

Value >= 4.

/\*

поиск универсальных лидеров

\*/

versatile\_leader(Leader) :-

strength(Leader, 'Science', S),

strength(Leader, 'Diplomacy', D),

strength(Leader, 'War', W),

strength(Leader, 'Culture', C),

strength(Leader, 'Faith', F),

S >= 3, D >= 3, W >= 3, C >= 3, F >= 3.

/\*

поиск самых легких универсальных лидеров

\*/

easiest\_versatile\_leader(Leader) :-

versatile\_leader(Leader),

difficulty(Leader, Difficulty),

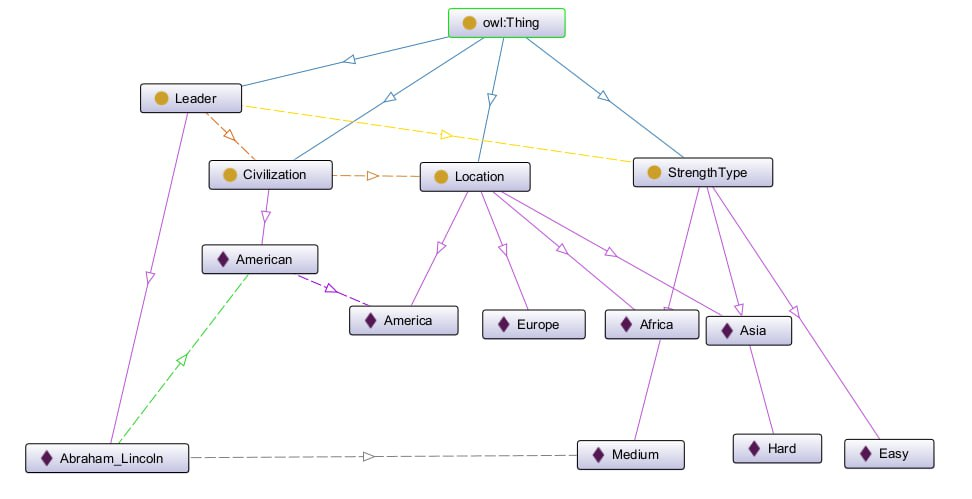
difficulty\_value(Difficulty, Value),

\+ (versatile\_leader(OtherLeader),

difficulty(OtherLeader, OtherDifficulty),

difficulty\_value(OtherDifficulty, OtherValue),

OtherValue < Value).



Для разработки системы поддержки принятия решений был выбран язык программирования Python и библиотека **pyswip** (репозиторий на GitHub: [ссылка](https://github.com/bullerun/AI_System/tree/main/lab2)).

Исходный код разработанной программы: [ссылка](https://github.com/bullerun/AI_System/blob/main/lab2/main.py)