Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**Отчёт по модулю №1**

по дисциплине

«Системы искусственного интеллекта»

*Выполнил:*

Студент группы P3333

Анисимов Максим Дмитриевич

*Преподаватель:*

Авдюшина Анна Евгеньевна



Санкт-Петербург, 2024

1. **Введение**

Данный модуль направлен на изучение языка программирования Prolog, концепции базы знаний, их создания и выполнения запросов. Этот раздел также включает в себя ознакомление с онтологией и инструментами разработки в Protege, разработкой системы поддержки принятия решений на основе базы знаний.

1. **Анализ требований**

Основные требования:

* Создание базы знаний в Prolog, имеющий факты и правила
* Создание запросов разной сложности к базе знаний
* Создание онтологической модели в Protege на основе базы знаний
* Разработка системы поддержки и принятия решения на основе базы знаний или онтологии

Требования к базе знаний и онтологии:

* База знаний должна включать не менее 20 фактов с одним аргументом, 10–15 фактов с двумя аргументами, которые дополняют и показывают связь с другими фактами и 5–7 правил
* В онтологии должны быть классы и свойства, соответствующие объектам и отношениям из базы знаний

1. **Изучение основных концепций и инструментов**

Основные концепции:

* База знаний – совокупность фактов с разными количествами аргументов (связанные между собой) и правил, которые описывают знания о какой-то определенной области
* Онтология – модель данных, показывающая объекты их свойства и их взаимоотношения

Инструменты:

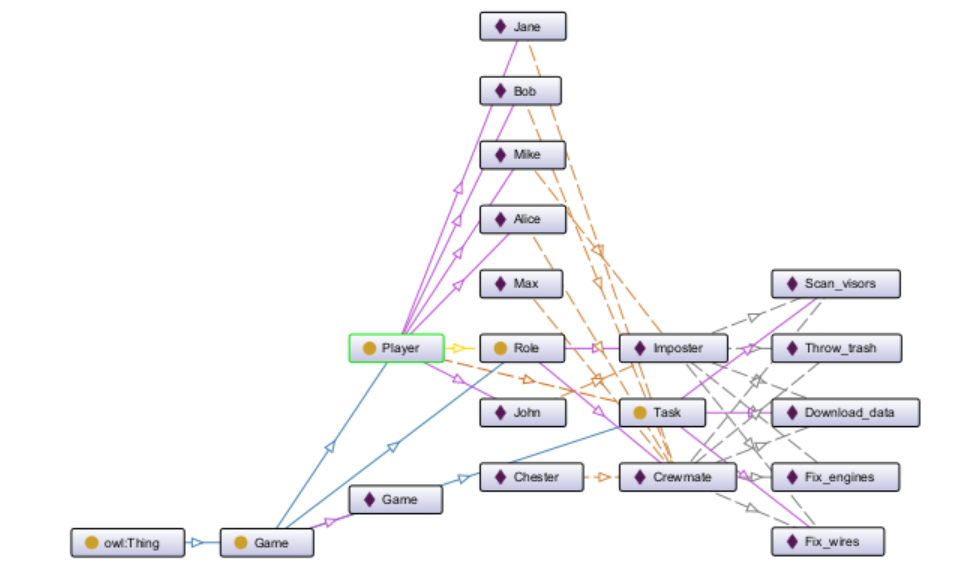
* Prolog – язык логического программирования, которые предоставляет возможности для написания баз знаний и для ввода запросов
* Protégé – приложение, в котором можно создавать онтологии

1. **Реализация системы искусственного интеллекта**

База знаний

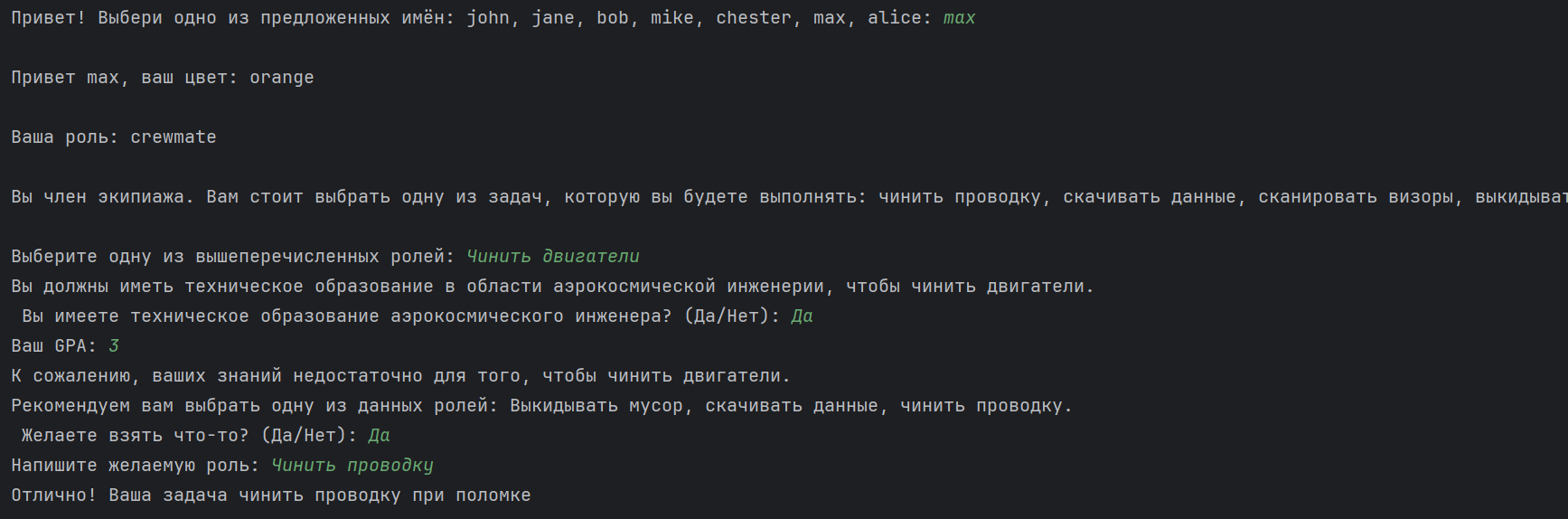
%факт об игре  
game(among\_us).  
  
%факты о жанре  
genre(mobile\_game).  
genre(social\_game).  
  
%факты о цвете персонажей  
color(red).  
color(blue).  
color(green).  
color(yellow).  
color(black).  
color(pink).  
color(orange).  
  
%факты о именах игроков  
nickname(john).  
nickname(jane).  
nickname(bob).  
nickname(alice).  
nickname(mike).  
nickname(chester).  
nickname(max).  
  
%факты о задачах экипажа  
task(fix\_wires).  
task(download\_data).  
task(scan\_visors).  
task(throw\_trash).  
task(fix\_engines).  
task(kill).  
  
%факты об именах игроков и их цветов  
player(john, red).  
player(jane, blue).  
player(bob, blue).  
player(alice, yellow).  
player(mike, black).  
player(max, orange).  
player(chester, pink).  
  
%факты о имени игрока и его роли  
role(john, impostor).  
role(jane, crewmate).  
role(bob, crewmate).  
role(alice, crewmate).  
role(mike, impostor).  
role(max, crewmate).  
role(chester, crewmate).  
  
%факты о задачах в зависимости от роли  
role\_task(crewmate, fix\_wires).  
role\_task(crewmate, download\_data).  
role\_task(crewmate, scan\_visors).  
role\_task(crewmate, throw\_trash).  
role\_task(crewmate, fix\_engines).  
role\_task(impostor, kill).  
  
%правила для членов экипажа  
is\_crewmate(Player):-  
 role(Player, crewmate).  
  
%правила для предателей  
is\_imposter(Player):-  
 player(Player, impostor).  
  
%правила для определения игроков и их возможности выполнять задачи  
can\_complete\_task(Player, Role):-  
 role\_task(Role, Task).  
  
%правила проверки двух игроков на то одинаковый ли у них цвет  
has\_same\_color(Player1, Player2):-  
 player(Player1, Color),  
 player(Player2, Color).  
  
%правила проверки двух игроков на то разный ли у них цвет  
has\_different\_color(Player1, Player2):-  
 player(Player1, Color1),  
 player(Player2, Color2),  
 Color1 \= Color2.  
  
%правила проверки двух игроков на то, входит ли их цвет в список предложенных  
has\_common\_color(Player1, Player2):-  
 player(Player1, Color),  
 player(Player2, Color),  
 (Color == red ; Color == blue ; Color == green).

Онтология



Рекомендательная система:

from pyswip import Prolog  
  
  
def switch(arg):  
 if arg == "чинить проводку" or arg == "Чинить проводку":  
 response = input("Вы должны хорошо знать теллекоммуникационные сети, а так же техническое образование, чтобы чинить проводку.\n Вы имеете техническое образование? (Да/Нет): ")  
 if response == "Да" or response == "да":  
 response2 = input("Ваш GPA: ")  
 if float(response2) < 3.0:  
 response3 = input("К сожалению, ваших знаний недостаточно для того, чтобы чинить проводку. Рекомендуем вам выбрать одну из данных ролей: Выкидывать мусор, скачивать данные.\n Желаете взять что-то? (Да/Нет): ")  
 if response3 == "Да" or response3 == "да":  
 response4 = input("Напишите желаемую роль: ")  
 if response4 == "скачивать данные" or arg == "Скачивать данные":  
 print("Отлично! Ваша задача скачивать данные, когда появится необходимость")  
 return False  
 elif response4 == "выкидывать мусор" or response4 == "Выкидывать мусор":  
 print("Отлично! Ваша задача очистить корабль от мусора")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif response3 == "Нет" or response3 == "нет":  
 print("К сожалению, вы не выбрали ни одной роли. Игра для вас недоступна")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif float(response2) >= 3.0 and float(response2) <= 4.0:  
 print("Отлично! Ваша задача чинить проводку при поломке")  
 return False  
 else:  
 print("Некорректный ввод. Повторите попытку")  
 elif response == "Нет" or response == "нет":  
 response5 = input("К сожалению, ваших знаний недостаточно для того, чтобы чинить проводку. Рекомендуем вам выбрать одну из данных ролей: Выкидывать мусор, скачивать данные. Желаете взять что-то? (Да/Нет)")  
 while (True):  
 if response5 == "Да" or response5 == "да":  
 response6 = input("Напишите желаемую роль: ")  
 while (True):  
 if response6 == "скачивать данные" or arg == "Скачивать данные":  
 print("Отлично! Ваша задача скачивать данные, когда появится необходимость")  
 return False  
 elif arg == "выкидывать мусор" or arg == "Выкидывать мусор":  
 print("Отлично! Ваша задача очистить корабль от мусора")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif response5 == "Нет" or response5 == "нет":  
 print("К сожалению, вы не выбрали ни одной роли. Игра для вас недоступна")  
 return False  
 else:  
 print("Некорректный ввод")  
 else:  
 print("Некорректный ввод. Повторите попытку")  
 elif arg == "скачивать данные" or arg == "Скачивать данные":  
 print("Отлично! Ваша задача скачивать данные, когда появится необходимость")  
 return False  
 elif arg == "сканировать визоры" or arg == "Сканировать визоры":  
 print("Отлично! Ваша задача сканировать других игроков в специальной комнате")  
 response = input("Вы должны хорошо знать устройство рентген-аппарата, а так же иметь техническое образование в области физики, чтобы сканировать визоров.\n Вы имеете техническое образование в области физики? (Да/Нет): ")  
 if response == "Да" or response == "да":  
 response2 = input("Ваш GPA: ")  
 if float(response2) < 3.0:  
 response3 = input(  
 "К сожалению, ваших знаний недостаточно для того, чтобы сканировать визоры. Рекомендуем вам выбрать одну из данных ролей: Выкидывать мусор, скачивать данные, чинить проводку.\n Желаете взять что-то? (Да/Нет): ")  
 if response3 == "Да" or response3 == "да":  
 response4 = input("Напишите желаемую роль: ")  
 if response4 == "скачивать данные" or arg == "Скачивать данные":  
 print("Отлично! Ваша задача скачивать данные, когда появится необходимость")  
 return False  
 elif response4 == "выкидывать мусор" or response4 == "Выкидывать мусор":  
 print("Отлично! Ваша задача очистить корабль от мусора")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif response3 == "Нет" or response3 == "нет":  
 print("К сожалению, вы не выбрали ни одной роли. Игра для вас недоступна")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif float(response2) >= 3.0 and float(response2) <= 4.0:  
 print("Отлично! Ваша задача чинить проводку при поломке")  
 return False  
 else:  
 print("Некорректный ввод. Повторите попытку")  
 elif response == "Нет" or response == "нет":  
 response5 = input(  
 "К сожалению, ваших знаний недостаточно для того, чтобы сканировать визоры. Рекомендуем вам выбрать одну из данных ролей: Выкидывать мусор, скачивать данные. Желаете взять что-то? (Да/Нет)")  
 while (True):  
 if response5 == "Да" or response5 == "да":  
 response6 = input("Напишите желаемую роль: ")  
 while (True):  
 if response6 == "скачивать данные" or response6 == "Скачивать данные":  
 print("Отлично! Ваша задача скачивать данные, когда появится необходимость")  
 return False  
 elif response6 == "выкидывать мусор" or response6 == "Выкидывать мусор":  
 print("Отлично! Ваша задача очистить корабль от мусора")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif response5 == "Нет" or response5 == "нет":  
 print("К сожалению, вы не выбрали ни одной роли. Игра для вас недоступна")  
 return False  
 else:  
 print("Некорректный ввод")  
 else:  
 print("Некорректный ввод. Повторите попытку")  
 elif arg == "выкидывать мусор" or arg == "Выкидывать мусор":  
 print("Отлично! Ваша задача очистить корабль от мусора")  
 return False  
 elif arg == "чинить двигатели" or arg == "Чинить двигатели":  
 response = input(  
 "Вы должны иметь техническое образование в области аэрокосмической инженерии, чтобы чинить двигатели.\n Вы имеете техническое образование аэрокосмического инженера? (Да/Нет): ")  
 if response == "Да" or response == "да":  
 response2 = input("Ваш GPA: ")  
 if float(response2) < 4.0:  
 response3 = input(  
 "К сожалению, ваших знаний недостаточно для того, чтобы чинить двигатели. Рекомендуем вам выбрать одну из данных ролей: Выкидывать мусор, скачивать данные, чинить проводку.\n Желаете взять что-то? (Да/Нет): ")  
 if response3 == "Да" or response3 == "да":  
 response4 = input("Напишите желаемую роль: ")  
 if response4 == "скачивать данные" or response4 == "Скачивать данные":  
 print("Отлично! Ваша задача скачивать данные, когда появится необходимость")  
 return False  
 elif response4 == "выкидывать мусор" or response4 == "Выкидывать мусор":  
 print("Отлично! Ваша задача очистить корабль от мусора")  
 return False  
 elif response4 == "чинить проводку" or response4 == "Чинить проводку":  
 print("Отлично! Ваша задача чинить проводку при поломке")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif response3 == "Нет" or response3 == "нет":  
 print("К сожалению, вы не выбрали ни одной роли. Игра для вас недоступна")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif float(response2) == 4.0:  
 print("Отлично! Ваша задача чинить поломку двигателя корабля")  
 return False  
 else:  
 print("Некорректный ввод. Повторите попытку")  
 elif response == "Нет" or response == "нет":  
 response5 = input(  
 "К сожалению, ваших знаний недостаточно для того, чтобы чинить двигатели. Рекомендуем вам выбрать одну из данных ролей: Выкидывать мусор, скачивать данные, чинить проводку. Желаете взять что-то? (Да/Нет)")  
 while (True):  
 if response5 == "Да" or response5 == "да":  
 response6 = input("Напишите желаемую роль: ")  
 while (True):  
 if response6 == "скачивать данные" or arg == "Скачивать данные":  
 print("Отлично! Ваша задача скачивать данные, когда появится необходимость")  
 return False  
 elif response6 == "выкидывать мусор" or response6 == "Выкидывать мусор":  
 print("Отлично! Ваша задача очистить корабль от мусора")  
 return False  
 elif response6 == "чинить проводку" or response6 == "Чинить проводку":  
 print("Отлично! Ваша задача чинить проводку при поломке")  
 return False  
 else:  
 print("Некорректный ввод")  
 elif response5 == "Нет" or response5 == "нет":  
 print("К сожалению, вы не выбрали ни одной роли. Игра для вас недоступна")  
 return False  
 else:  
 print("Некорректный ввод")  
 else:  
 print("Некорректный ввод. Повторите попытку")  
 else:  
 print("Некорректный ввод данных. Пожалуйста, повторите попытку")  
 return False  
  
  
def get\_player\_recommendations(player\_name):  
 prolog = Prolog()  
 prolog.consult("lab1.pl")  
 names = ["john", "jane", "bob", "mike", "chester", "max", "alice"]  
 if player\_name not in names:  
 print("Некорректный ввод")  
 return  
 color = list(prolog.query(f"player({player\_name}, Color)"))[0]['Color']  
 role = list(prolog.query(f"role({player\_name}, Role)"))[0]['Role']  
 incorrect\_data = True  
  
 print(f"\nПривет {player\_name}, ваш цвет: {color}\n\nВаша роль: {role}\n")  
  
 tasks = [task['Task'] for task in prolog.query(f"can\_complete\_task({player\_name}, Task)\n")]  
  
 while (incorrect\_data):  
 if len(tasks) == 1:  
 print("Вы предатель! Ваша задача незаметно убить весь экипаж")  
 break  
 elif tasks:  
 print(f"Вы член экипиажа. Вам стоит выбрать одну из задач, которую вы будете выполнять: чинить проводку, скачивать данные, сканировать визоры, выкидывать мусор, чинить двигатели\n")  
 player\_task = input("Выберите одну из вышеперечисленных ролей: ")  
  
 incorrect\_data = switch(player\_task)  
  
 if not incorrect\_data:  
 break  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 player\_name = input("Привет! Выбери одно из предложенных имён: john, jane, bob, mike, chester, max, alice: ")  
 get\_player\_recommendations(player\_name)

Тестирования рекомендательной системы:

**5. Оценка и интерпретация результатов:**

Примеры запросов к базе знаний:

* player(john, Color).
* is\_crewmate(Player), player(Player, Color), Color \= red.
* has\_common\_color(Player, john).
* Примеры запросов к онтологии:
* Player
* Player and has\_task value fix\_wires
* Player and has\_role value impostor

Все требования были выполнены

Данную систему можно улучшить таким образом, что через неё можно создать офлайн версию игры Among Us

1. **Заключение**

Преимуществом разработанной системы искусственного интеллекта является то она достаточно простая в использовании и даёт возможность усовершенствовать её в дальнейшем