ФГАОУ ВО «КФУ им. В.И. Вернадского»
физико-технический институт (структурное подразделение)

Кафедра компьютерной инженерии и моделирования

Порозов Кирилл Сергеевич

отчет по практической работе №4
по дисциплине **«ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ»**

Направление подготовки:
09.03.04 "Программная инженерия"

Оценка -

Симферополь, 2022

Практическая работа №4. Тема: "Логическое программирование"

Цель работы: Изучить парадигму логического программирования, преимущества и недостатки парадигмы, научиться создавать простейшие приложения на языке высокого уровня, реализующие логический подход к созданию программ.

Описание ключевых понятий:

(при необходимости)

Перед выполнением практической работы изучена следующая литература:

1. Презентация лектора курса: «Логическое программирование»
2. Прослушана видеолекция и посещены практические занятия Милюкова Виктора Васильевича.
3. Прочитаны материалы статей: AI with Python – My Blog (foobrdigital.com) ; logpy/logpy: Logic Programming in Python (github.com)

Написана программа на языке Python, реализующая алгоритм:
Пример использования логической парадигмы для решения задачи поиска нужного параметра, зависящего от других условий (Задачка Эйнштейна про 5 домов):

```
Решаемая задача:
5 разных человек в 5 разных домах разного цвета, курят 5 разных марок сигарет,
выращивают 5 разных видов животных, пьют 5 разных видов напитков.
Вопрос: кто выращивает хомяка?


Норвежец живет в первом доме.
Англичанин живет в красном доме.
Зеленый дом находится левее белого.
Датчанин пьет чай.
Тот, кто курит Rothmans, живет рядом с тем, кто выращивает кошек.
Тот, кто живет в желтом доме, курит Dunhill.
Немец курит Marlboro.
Тот, кто живет в центре, пьет молоко.
Сосед того, кто курит Rothmans, пьет воду.
Тот, кто курит Pall Mall, выращивает птиц.
Швед выращивает собак.
Норвежец живет рядом с синим домом.
Тот, кто выращивает лошадей, живет в синем доме.
Тот, кто курит Philip Morris, пьет пиво.
В зеленом доме пьют кофе.
"""
from kanren import *
from kanren.core import lall

# Функция left и next определяет отношение положения домов,
# чей дом слева или справа(следующий) по отношению к другому дому
def left(q, p, list):
    return membero((q,p), zip(list, list[1:]))
def next(q, p, list):
    return conde([left(q, p, list)], [left(p, q, list)])

houses = var()
# lall - представляет конъюнктуру описанных в функции параметров.
# Учитаны должны быть все условия, поэтому используем lall
rules_zebraproblem = lall(
    # У нас 5 домов. Объявляем 5 переменных, которые должны быть подобраны в houses
    (eq, (var(), var(), var(), var(), var()), houses),
```

```
    # Вводим в программу все вышеописанные условия

    # membero - мы спрашиваем элемент, описанный 5-ю переменными, в нашем случае -
    # это нация, сигареты, питье, животное, цвет дома. По этим параметрам логические
    # функции самостоятельно подбирают ответы, сопоставляя соответствия.
    (membero('Англичанин', var(), var(), var(), 'красный'), houses),
    (membero('Швед', var(), var(), 'собака', var()), houses),
    (membero('Датчанин', var(), 'чай', var(), var()), houses),
    (left(var(), var(), var(), var(), 'зелёный'),
    (var(), var(), var(), var(), 'белый'), houses),
    (membero(var(), var(), 'кофе', var(), 'зелёный'), houses),
    (membero(var(), 'Pall Mall', var(), 'птички', var()), houses),
    (membero(var(), 'Dunhill', var(), var(), 'жёлтый'), houses),
    (eq(var(), var(), (var(), var(), 'молоко', var(), var()), var(), var()), houses),
    (eq(('Норвежец', var(), var(), var(), var()), var(), var(), var(), var()), houses),
    (next(var(), 'Blend', var(), var(), var()),
    (var(), var(), var(), 'кошки', var()), houses),
    (next(var(), 'Dunhill', var(), var(), var()),
    (var(), var(), var(), 'Лошадь', var()), houses),
    (membero(var(), 'Blue Master', 'пиво', var(), var()), houses),
    (membero('Немец', 'Prince', var(), var(), var()), houses),
    (next('Норвежец', var(), var(), var(), var()),
    (var(), var(), var(), var(), 'синий'), houses),
    (next(var(), 'Blend', var(), var(), var()),
    (var(), var(), 'вода', var(), var()), houses),
    (membero(var(), var(), var(), 'хомяк', var()), houses)
)

# Мы запрашиваем список домов со всеми характеристиками у нашей логической функции,
# в которой прописаны правила
solutions = run(0, houses, rules_zebraproblem)

# Сама суть программы - никаких данных про хомяка нет.
# Программа самостоятельно генерирует ответ и к нужной записи добавляет хомяка

output_zebra = [house for house in solutions[0] if 'хомяк' in house][0][0]
print('\n'+ output_zebra + ' владелец хомяка.')
```

```
main (3)
Немец владелец хомяка.
```

Более подробно – на странице задания на гитхабе:
[Prog_Paradigms/4 задание - логическое at main · JustBlood/Prog_Paradigms (github.com)](https://github.com/JustBlood/Prog_Paradigms/tree/main/4%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5%20-%20%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5)

Проект представлен преподавателю в электронной форме, продемонстрирована работоспособность программы, разъяснены детали программного кода.

Вопросы, заданные преподавателем: