Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Пермский национальный исследовательский

политехнический университет» (ПНИПУ)

Кафедра вычислительной математики и механики

**Лабораторная работа № 2**

**по дисциплине: «Интеллектуальные ИСИТ»**

Выполнила

студент группы ИСТ-19-1б

Лопина М.С.

Проверил

ассистент кафедры ВММБ

Нетбай Г.В.

Пермь, 2022

**Цель**: сформировать у студентов способность построения учебного прототипа экспертной системы, основанной на знаниях, включающего основные компоненты ИИС.

**Описание**: в рамках данной работы необходимо реализовать оболочку ЭС, которая была бы способна работать с продукционной БЗ. Работа позволит углубленно познакомиться со способами взаимодействия компонентов ЭС, а также реализовать отдельные функциональные блоки.

Результат:

Предметная область – выбор подарка, сделанного своими руками.

Пример работы программы с результатом –игровая (рис.1) и авторская (рис.2) игрушка:

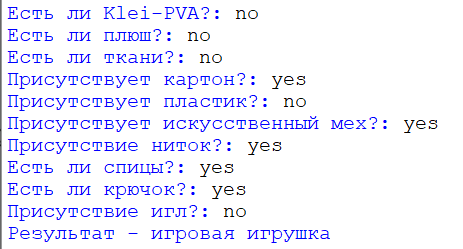


Рис.1

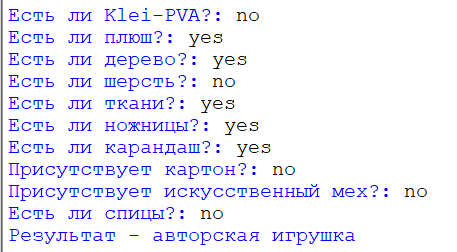


Рис.2

Скрипт правил в формате JSON:

[

{

"rule\_name" : "ask\_Klei-PVA",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Klei-PVA", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Klei-PVA",

{ "func": "yes\_or\_no", "arg": "Есть ли Klei-PVA?: "} ]

}

},

{

"rule\_name" : "ask\_Kraski",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Klei-PVA", "yes" ] } ,

{ "func": "eq", "arg": [ "Kraski", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Kraski",

{ "func": "yes\_or\_no", "arg": "Есть ли краски?: "} ]

}

},

{

"rule\_name" : "ask\_Plush",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Plush", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Plush",

{ "func": "yes\_or\_no", "arg": "Есть ли плюш?: "} ]

}

},

{

"rule\_name" : "ask\_Derevo",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Plush", "yes" ] },

{ "func": "eq", "arg": [ "Derevo", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Derevo",

{ "func": "yes\_or\_no", "arg": "Есть ли дерево?: "} ]

}

},

{

"rule\_name" : "ask\_Sherst",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Plush", "yes" ] },

{ "func": "eq", "arg": [ "Derevo", "yes" ] },

{ "func": "eq", "arg": [ "Sherst", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Sherst",

{ "func": "yes\_or\_no", "arg": "Есть ли шерсть?: "} ]

}

},

{

"rule\_name" : "ask\_Tkani",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Tkani", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Tkani",

{ "func": "yes\_or\_no", "arg": "Есть ли ткани?: "} ]

}

},

{

"rule\_name" : "ask\_Nognitci",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Tkani", "yes" ] },

{ "func": "eq", "arg": [ "Nognitci", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Nognitci",

{ "func": "yes\_or\_no", "arg": "Есть ли ножницы?: "} ]

}

},

{

"rule\_name" : "ask\_Karandash",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Tkani", "yes" ] },

{ "func": "eq", "arg": [ "Nognitci", "yes" ] },

{ "func": "eq", "arg": [ "Karandash", 0 ] }]

},

"actions": { "func": "set", "arg":

[ "Karandash",

{ "func": "yes\_or\_no", "arg": "Есть ли карандаш?: "} ]

}

},

{

"rule\_name" : "ask\_Prisytstvuet-karton",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Prisytstvuet-karton", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Prisytstvuet-karton",

{ "func": "yes\_or\_no", "arg": "Присутствует картон?: "} ]

}

},

{

"rule\_name" : "ask\_Prisytstvuet-plastic",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Prisytstvuet-karton", "yes" ] },

{ "func": "eq", "arg": [ "Prisytstvuet-plastic", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Prisytstvuet-plastic",

{ "func": "yes\_or\_no", "arg": "Присутствует пластик?: "} ]

}

},

{

"rule\_name" : "ask\_Iskustvenui-mex",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Iskustvenui-mex", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Iskustvenui-mex",

{ "func": "yes\_or\_no", "arg": "Присутствует искусственный мех?: "} ]

}

},

{

"rule\_name" : "ask\_Prisytstvuet-nitok",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Iskustvenui-mex", "yes" ] },

{ "func": "eq", "arg": [ "Prisytstvuet-nitok", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Prisytstvuet-nitok",

{ "func": "yes\_or\_no", "arg": "Присутствие ниток?: "} ]

}

},

{

"rule\_name" : "ask\_Spicu",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Spicu", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Spicu",

{ "func": "yes\_or\_no", "arg": "Есть ли спицы?: "} ]

}

},

{

"rule\_name" : "ask\_kruchok",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Spicu", "yes"] },

{ "func": "eq", "arg": [ "kruchok", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "kruchok",

{ "func": "yes\_or\_no", "arg": "Есть ли крючок?: "} ]

}

},

{

"rule\_name" : "ask\_Prisytstvuet-igl",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Spicu", "yes"] },

{ "func": "eq", "arg": [ "kruchok", "yes" ] },

{ "func": "eq", "arg": [ "Prisytstvuet-igl", 0 ] } ]

},

"actions": { "func": "set", "arg":

[ "Prisytstvuet-igl",

{ "func": "yes\_or\_no", "arg": "Присутствие игл?: "} ]

}

},

{

"rule\_name" : "answer\_0",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Klei-PVA", "yes" ] },

{ "func": "eq", "arg": [ "Kraski", "yes" ] },

{ "func": "eq", "arg": [ "Plush", "no" ]},

{ "func": "eq", "arg": [ "Tkani", "no" ]}]

},

"actions": { "func": "and", "arg":

[ { "func": "set", "arg": [ "Solution", 1] },

{ "func": "print", "arg": "Результат - авторская игрушка" } ]

}

},

{

"rule\_name" : "answer\_1",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Klei-PVA", "no" ]},

{ "func": "eq", "arg": [ "Sherst", "no" ]},

{ "func": "eq", "arg": [ "Prisytstvuet-karton", "no" ] },

{ "func": "eq", "arg": [ "Karandash", "yes" ]}]

},

"actions": { "func": "and", "arg":

[ { "func": "set", "arg": [ "Solution", 1] },

{ "func": "print", "arg": "Результат - авторская игрушка" } ]

}

},

{

"rule\_name" : "answer\_2",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Klei-PVA", "no" ]},

{ "func": "eq", "arg": [ "Prisytstvuet-plastic", "no" ]},

{ "func": "eq", "arg": [ "Sherst", "no" ]},

{ "func": "eq", "arg": [ "Karandash", "yes" ]}]

},

"actions": { "func": "and", "arg":

[ { "func": "set", "arg": [ "Solution", 1] },

{ "func": "print", "arg": "Художественная и интерьерная игрушка" } ]

}

},

{

"rule\_name" : "answer\_3",

"conditions": { "func": "and", "arg":

[ { "func": "eq", "arg": [ "Solution", 0 ] },

{ "func": "eq", "arg": [ "Prisytstvuet-plastic", "no" ]},

{ "func": "eq", "arg": [ "Prisytstvuet-nitok", "yes" ] },

{ "func": "eq", "arg": [ "Prisytstvuet-igl", "no" ] }]

},

"actions": { "func": "and", "arg":

[ { "func": "set", "arg": [ "Solution", 1] },

{ "func": "print", "arg": "Результат - игровая игрушка" } ]

}

}

]

Код программы:

import json

def andParams(\*params):

return all([el == True for el in params])

def eq(param, value):

if not(param in fact):

fact[param] = 0

return fact[param] == value

def set(param, value):

fact[param] = value

return fact[param]

def yes\_or\_no(question):

answer = ""

while(answer != "yes" and answer != "no"):

answer = input(question)

return answer

def ask\_question(question, \*values):

answer = ""

while all([str(el) != answer for el in values]):

answer = input(question)

return answer

def printResult(text):

print(text)

functions = {"and": andParams, "eq": eq, "set": set, "yes\_or\_no": yes\_or\_no, "ask\_question": ask\_question, "print": printResult}

fact = {"Solution": 0}

def ruleManager(part):

final\_arguments = []

if(isinstance(part["arg"], str)):

final\_arguments.append(part["arg"])

else:

for arg in part["arg"]:

if not(isinstance(arg, dict)):

final\_arguments.append(arg)

else:

final\_arguments.append(ruleManager(arg))

return functions.get(part["func"])(\*final\_arguments)

def ruleHandler():

i = 0;

while(fact["Solution"] == 0 and i < len(data)):

if (ruleManager(data[i]["conditions"])):

ruleManager(data[i]["actions"])

i += 1

with open("labb.json", "r", encoding="utf-8") as rules:

data = json.load(rules)

ruleHandler()