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

студентка группы ИСТ-19-2б

Клименко А.А.

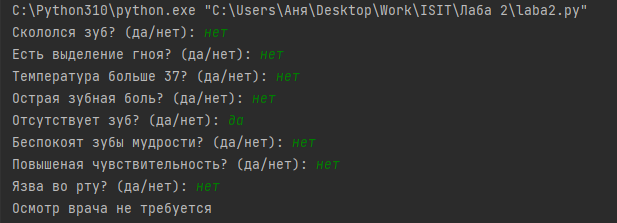
Проверил

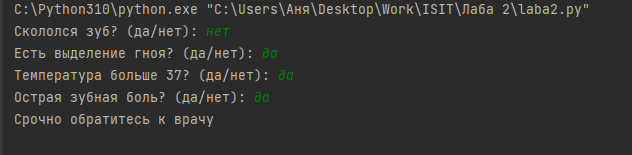
ассистент кафедры ВММБ

Нетбай Г.В.

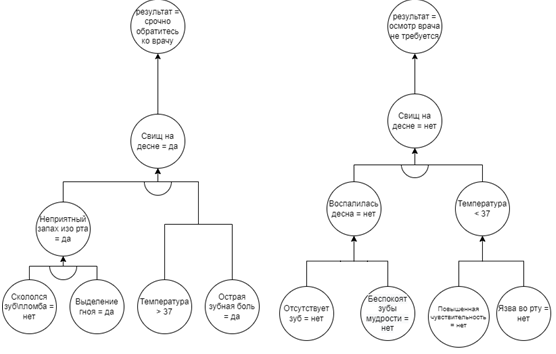
Пермь, 2022

Работа программы:





Граф



Скрипт правил в системе JSON

[

{

"rule":"q1",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["skol" , 0]}

]},

"right":{"op": "set", "arg": ["skol",{"op": "yes\_or\_no", "arg": "Скололся зуб?"}]}

},

{

"rule":"q2",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["gnoy" , 0]}

]},

"right":{"op": "set", "arg": ["gnoy",{"op": "yes\_or\_no", "arg": "Есть выделение гноя?"}]}

},

{

"rule":"a1",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["skol" , "нет"]},

{"op": "==", "arg": ["gnoy" , "да"]}

]},

"right":{"op": "set", "arg": ["a1", {"op": "found", "arg": ["zapah", "да"]}]}

}

,

{

"rule":"q3",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["temp" , 0]}

]},

"right":{"op": "set", "arg": ["temp",{"op": "yes\_or\_no", "arg": "Температура больше 37?"}]}

},

{

"rule":"q4",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["bol" , 0]}

]},

"right":{"op": "set", "arg": ["bol",{"op": "yes\_or\_no", "arg": "Острая зубная боль?"}]}

},

{

"rule":"a2",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["temp" , "да"]},

{"op": "==", "arg": ["bol" , "да"]},

{"op": "==", "arg": ["zapah", "да"]}

]},

"right":{"op": "set", "arg": ["a2", {"op": "found", "arg": ["svish", "да"]}]}

}

,

{

"rule":"a3",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["svish", "да"]}

]},

"right":{"op": "set", "arg": ["a3", {"op": "found", "arg": ["solution", 1]},

{"op": "print", "arg": "Срочно обратитесь к врачу"}]}

}

,

{

"rule":"q5",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["otsutstvuey\_zyb" , 0]}

]},

"right":{"op": "set", "arg": ["otsutstvuey\_zyb",{"op": "yes\_or\_no", "arg": "Отсутствует зуб?"}]}

},

{

"rule":"q6",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["zyb\_mydrosti" , 0]}

]},

"right":{"op": "set", "arg": ["zyb\_mydrosti",{"op": "yes\_or\_no", "arg": "Беспокоят зубы мудрости?"}]}

},

{

"rule":"a4",

"left": {"op": "or" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["otsutstvuey\_zyb", "нет"]},

{"op": "==", "arg": ["zyb\_mydrosti", "нет"]}

]},

"right":{"op": "set", "arg": ["a4",{"op": "found", "arg": ["vospalenie", "нет"]}]}

}

,

{

"rule":"q7",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["chuvstvitelnost" , 0]}

]},

"right":{"op": "set", "arg": ["chuvstvitelnost",{"op": "yes\_or\_no", "arg": "Повышеная чувствительность?"}]}

},

{

"rule":"q8",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["yazva" , 0]}

]},

"right":{"op": "set", "arg": ["yazva",{"op": "yes\_or\_no", "arg": "Язва во рту?"}]}

},

{

"rule":"a5",

"left": {"op": "or" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["chuvstvitelnost", "нет"]},

{"op": "==", "arg": ["yazva", "нет"]},

{"op": "==", "arg": ["temp", 0]}

]},

"right":{"op": "set", "arg": ["a5",{"op": "found", "arg": ["temp", "нет"]}]}

},

{

"rule":"a6",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["vospalenie", "нет"]},

{"op": "==", "arg": ["temp", "нет"]}

]},

"right":{"op": "set", "arg": ["a6",{"op": "found", "arg": ["svish", "нет"]}]}

},

{

"rule":"a7",

"left": {"op": "and" , "arg": [

{"op": "==", "arg": ["solution" , 0]},

{"op": "==", "arg": ["svish", "нет"]}

]},

"right":{"op": "set", "arg": ["a7",{"op": "found", "arg": ["solution", 1]},

{"op": "print", "arg": "Осмотр врача не требуется"}]}

}

]

Листинг программы:

import json  
with open("m.json", "r", encoding="utf-8") as read\_file:  
 data = json.load(read\_file)  
  
  
def sravn(arg):  
 x = arg[0]  
 if x in memory:  
 if (arg[1] == memory[x]):  
 return False  
 return True  
  
  
def op\_and(arg):  
 k = 0  
 for i in range(0, len(arg)):  
 if run(arg[i]) == False:  
 k += 1  
 if k == len(arg):  
 return True  
 if k > 0:  
 return False  
  
def op\_or(arg):  
 k = 0  
 for i in range(0, len(arg)):  
 if run(arg[i]) == False:  
 k += 1  
 if k > 1:  
 return True  
 else:  
 return False  
  
  
def pprint(arg):  
 arg = ''.join(arg)  
 print(arg)  
  
  
memory = {  
 "solution": 0,  
 "skol": 0,  
 "gnoy": 0,  
 "temp": 0,  
 "bol": 0,  
 "otsutstvuey\_zyb": 0,  
 "zyb\_mydrosti": 0,  
 "chuvstvitelnost": 0,  
 "yazva": 0,  
 "krovotochenie": 0,  
 "nalet": 0,  
 "karies": 0,  
 "zapah": 0,  
 "svish": 0,  
 "vospalenie": 0  
 }  
  
  
def get(arg):  
 return memory[arg[0]]  
  
  
def yes\_or\_no(arg):  
 arg = ''.join(arg)  
 answer = input(arg + ' (да/нет): ')  
 return answer  
  
  
def question\_weather(arg):  
 arg = ''.join(arg)  
 answer = int(input(arg + ' (числовое значение): '))  
 return answer  
  
  
def op\_set(arg):  
 x = arg[0]  
 for i in range(1, len(arg)):  
 res = run(arg[i])  
 if x in memory:  
 if (res != None):  
 memory[x] = res  
  
  
def found(arg):  
 x = arg[0]  
 if x in memory:  
 memory[x] = arg[1]  
  
  
func = {'==': sravn,  
 'print': pprint,  
 'get': get,  
 'yes\_or\_no': yes\_or\_no,  
 'question\_weather': question\_weather,  
 'and': op\_and,  
 'or': op\_or,  
 'set': op\_set,  
 'found': found  
 }  
  
  
def run(expression): # обработка вложенных  
 mas = [] # обработанные аргументы  
 op = expression.get('op')  
 func\_1 = func.get(op)  
 arg = expression.get('arg')  
 for i in range(len(arg)):  
 if isinstance(arg, dict):  
 mas.append(run(arg[i]))  
 else:  
 mas.append(arg[i])  
 return func\_1(mas)  
  
  
# Если левая часть правильная, то можем выполнить и правую.  
for rule in data:  
 if (run(rule['left']) == True):  
 run(rule['right'])