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»**

Выполнил

студент группы ИСТ-19-2б

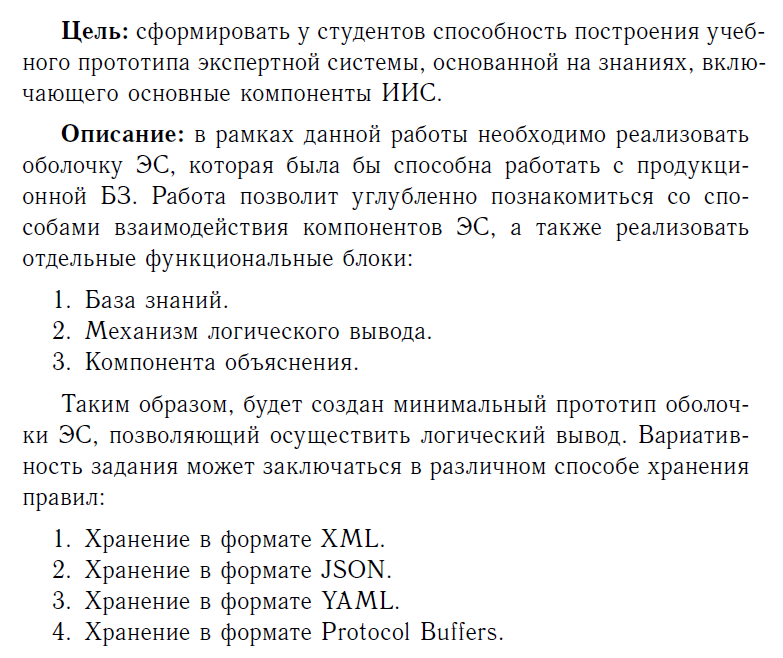
Лебедева А.А.

Проверил

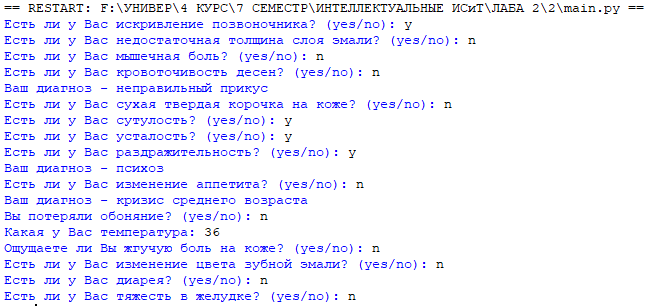
ассистент кафедры ВММБ

Нетбай Г.В.

Пермь, 2022



Пример запуска программы:



База знаний:

{

"ask-temp": {

"conditions": [ ["solution", "eq", "null"], "and", ["temp", "eq", "null"] ],

"actions": [ {"func": "ask\_int", "arg\_1": "temp", "arg\_2": "Какая у Вас температура: "} ]

},

"ask-heaviness-in-stomach": {

"conditions": [ ["solution", "eq", "null"], "and", ["stomach", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "stomach", "arg\_2": "Есть ли у Вас тяжесть в желудке? (yes/no): "} ]

},

"ask-diarrhea": {

"conditions": [ ["solution", "eq", "null"], "and", ["diarrhea", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "diarrhea", "arg\_2": "Есть ли у Вас диарея? (yes/no): "} ]

},

"ask-loss-of-sense-of-smell": {

"conditions": [ ["solution", "eq", "null"], "and", ["sense", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "sense", "arg\_2": "Вы потеряли обоняние? (yes/no): "} ]

},

"ask-dry-hard-crusts": {

"conditions": [ ["solution", "eq", "null"], "and", ["crusts", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "crusts", "arg\_2": "Есть ли у Вас сухая твердая корочка на коже? (yes/no): "} ]

},

"ask-burning-pain": {

"conditions": [ ["solution", "eq", "null"], "and", ["pain", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "pain", "arg\_2": "Ощущаете ли Вы жгучую боль на коже? (yes/no): "} ]

},

"ask-slouching": {

"conditions": [ ["solution", "eq", "null"], "and", ["slouching", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "slouching", "arg\_2": "Есть ли у Вас сутулость? (yes/no): "} ]

},

"ask-curvature-of-spine": {

"conditions": [ ["solution", "eq", "null"], "and", ["curvature", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "curvature", "arg\_2": "Есть ли у Вас искривление позвоночника? (yes/no): "} ]

},

"ask-bleeding-gums": {

"conditions": [ ["solution", "eq", "null"], "and", ["gums", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "gums", "arg\_2": "Есть ли у Вас кровоточивость десен? (yes/no): "} ]

},

"ask-changing-color-of-tooth-enamel": {

"conditions": [ ["solution", "eq", "null"], "and", ["changing-color", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "changing-color", "arg\_2": "Есть ли у Вас изменение цвета зубной эмали? (yes/no): "} ]

},

"ask-insufficient-thickness-of-enamel-layer": {

"conditions": [ ["solution", "eq", "null"], "and", ["insufficient-thickness", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "insufficient-thickness", "arg\_2": "Есть ли у Вас недостаточная толщина слоя эмали? (yes/no): "} ]

},

"ask-fatigue": {

"conditions": [ ["solution", "eq", "null"], "and", ["fatigue", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "fatigue", "arg\_2": "Есть ли у Вас усталость? (yes/no): "} ]

},

"ask-change-in-appetite": {

"conditions": [ ["solution", "eq", "null"], "and", ["appetite", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "appetite", "arg\_2": "Есть ли у Вас изменение аппетита? (yes/no): "} ]

},

"ask-muscle-pain": {

"conditions": [ ["solution", "eq", "null"], "and", ["muscle-pain", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "muscle-pain", "arg\_2": "Есть ли у Вас мышечная боль? (yes/no): "} ]

},

"ask-irritability": {

"conditions": [ ["solution", "eq", "null"], "and", ["irritability", "eq", "null"] ],

"actions": [ {"func": "yes\_or\_no", "arg\_1": "irritability", "arg\_2": "Есть ли у Вас раздражительность? (yes/no): "} ]

},

"RULE-1-diagnosis-of-pancreatitis": {

"conditions": [ ["solution", "eq", "null"], "and", ["temp", ">", 35], "and", ["temp", "<", 37], "and", ["stomach", "eq", "no"], "and", ["diarrhea", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "pancreatitis", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - панкреатит"}

]

},

"RULE-2-diagnosis-of-gastritis": {

"conditions": [ ["temp", ">", 35], "and", ["temp", "<", 37], "and", ["stomach", "eq", "yes"], "and", ["diarrhea", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "gastritis", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - гастрит"}

]

},

"RULE-3-diagnosis-of-covid": {

"conditions": [ ["temp", ">", 37], "and", ["temp", "<", 38], "and", ["sense", "eq", "yes"], "and", ["diarrhea", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "covid", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - covid-19"}

]

},

"RULE-4-diagnosis-of-flu": {

"conditions": [ ["temp", ">", 37], "and", ["temp", "<", 38], "and", ["sense", "eq", "no"], "and", ["diarrhea", "eq", "no"] ],

"actions": [

{"func": "assert", "arg\_1": "flu", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - грипп"}

]

},

"RULE-5-diagnosis-of-chemical-burn": {

"conditions": [ ["crusts ", "eq", "yes"], "and", ["pain", "eq", "no"] ],

"actions": [

{"func": "assert", "arg\_1": "chemical-burn", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - химический ожог"}

]

},

"RULE-6-diagnosis-of-thermal-burn": {

"conditions": [ ["crusts ", "eq", "no"], "and", ["pain", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "thermal-burn", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - термический ожога"}

]

},

"RULE-7-diagnosis-of-kyphosis": {

"conditions": [ ["slouching", "eq", "yes"], "and", ["curvature", "eq", "no"] ],

"actions": [

{"func": "assert", "arg\_1": "kyphosis", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - кифоз"}

]

},

"RULE-8-diagnosis-of-scoliosis": {

"conditions": [ ["slouching", "eq", "no"], "and", ["curvature", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "scoliosis", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - сколиоз"}

]

},

"RULE-9-diagnosis-of-plaque": {

"conditions": [ ["gums", "eq", "yes"], "and", ["changing-color", "eq", "yes"], "and", ["insufficient-thickness", "eq", "no"] ],

"actions": [

{"func": "assert", "arg\_1": "plaque", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - зубной налет"}

]

},

"RULE-10-diagnosis-of-gingivitis": {

"conditions": [ ["gums", "eq", "yes"], "and", ["changing-color", "eq", "no"] ],

"actions": [

{"func": "assert", "arg\_1": "gingivitis", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - гингивит"}

]

},

"RULE-11-diagnosis-of-hypoplasia": {

"conditions": [ ["gums", "eq", "no"], "and", ["changing-color", "eq", "yes"], "and", ["insufficient-thickness", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "hypoplasia", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - гипоплазия"}

]

},

"RULE-12-diagnosis-of-malocclusion": {

"conditions": [ ["gums", "eq", "no"], "or", ["slouching", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "malocclusion", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - неправильный прикус"}

]

},

"RULE-13-diagnosis-of-midlife-crisis": {

"conditions": [ ["fatigue", "eq", "yes"], "and", ["appetite", "eq", "no"], "and", ["muscle-pain", "eq", "no"] ],

"actions": [

{"func": "assert", "arg\_1": "midlife-crisis", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - кризис среднего возраста"}

]

},

"RULE-14-diagnosis-of-mild-neurosis": {

"conditions": [ ["fatigue", "eq", "yes"], "and", ["appetite", "eq", "no"], "and", ["muscle-pain", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "mild-neurosis", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - легкий невроз"}

]

},

"RULE-15-diagnosis-of-depression": {

"conditions": [ ["fatigue", "eq", "yes"], "and", ["appetite", "eq", "yes"], "or", ["irritability", "eq", "no"] ],

"actions": [

{"func": "assert", "arg\_1": "depression", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - депрессия"}

]

},

"RULE-16-diagnosis-of-psychosis": {

"conditions": [ ["fatigue", "eq", "no"], "and", ["appetite", "eq", "yes"], "or", ["irritability", "eq", "yes"] ],

"actions": [

{"func": "assert", "arg\_1": "psychosis", "arg\_2": "yes"},

{"func": "print", "arg\_1": "Ваш диагноз - психоз"}

]

}

}

Листинг кода main.py:

from code import interact

import json

from random import choice

import pathlib

class rule:

def \_\_init\_\_(self, id:str, conditions: list, actions: list) -> None:

self.id = id

self.conditions = conditions

self.actions = actions

self.activated = False

if any([action["func"]=="yes\_or\_no" or action["func"]=="ask\_int" for action in self.actions]):

self.requires\_input = True

else:

self.requires\_input = False

class machine:

def \_\_init\_\_(self, jsonfile: str) -> None:

self.raw\_data = {}

self.rules = {}

self.answered = {}

with open(jsonfile, "r", encoding='utf-8') as read\_rules:

self.raw\_data = json.load(read\_rules)

for id, info in self.raw\_data.items():

self.rules[id] = rule(id, info["conditions"], info["actions"])

for r in self.rules.values():

for condition in r.conditions:

if isinstance(condition, list):

self.answered[condition[0]] = None

def perform\_actions(self, rule\_id: str) -> None:

for action in self.rules[rule\_id].actions:

if action["func"] == "ask\_int":

self.answered[action["arg\_1"]] = self.ask\_int(action["arg\_2"])

elif action["func"] == "yes\_or\_no":

self.answered[action["arg\_1"]] = self.yes\_or\_no(action["arg\_2"])

elif action["func"] == "assert":

self.answered[action["arg\_1"]] = action["arg\_2"]

elif action["func"] == "print":

self.print(action["arg\_1"])

@staticmethod

def ask\_int(question: str) -> int:

while True:

answer = input(question)

try:

answer = int(answer)

return answer

except:

continue

@staticmethod

def yes\_or\_no(question: str) -> str:

while True:

answer = input(question)

if answer.lower() in ["yes", "y"]:

return "yes"

elif answer.lower() in ["no", "n"]:

return "no"

@staticmethod

def print(text: str) -> None:

print(text)

def interprete(self, rule\_id: str) -> tuple:

text = ""

for item in self.rules[rule\_id].conditions:

temp = ""

if isinstance(item, list):

temp += f"self.answered[\"{item[0]}\"]"

if item[1] == "eq":

temp += " == "

elif item[1] == ">":

temp += " > "

elif item[1] == "<":

temp += " < "

if item[2] == "null":

temp += "None"

elif isinstance(item[2], int):

temp += f"{item[2]}"

else:

temp += f"\"{item[2]}\""

elif isinstance(item, str):

temp = f" {item} "

text += temp

try:

ready = eval(text)

except:

ready = False

return text, ready

def run(self) -> None:

while True:

# asking for input

options = []

for rule in self.rules.values():

cond, ready = self.interprete(rule.id)

if ready == True and rule.activated == False:

options += [rule.id]

try:

chosen\_rule = choice(options)

self.perform\_actions(chosen\_rule)

self.rules[chosen\_rule].activated = True

except:

break

self.implications()

def implications(self):

# calculating implications

while True:

done = True

for rule in self.rules.values():

cond, ready = self.interprete(rule.id)

if ready == True and rule.activated == False and rule.requires\_input == False:

self.perform\_actions(rule.id)

rule.activated = True

done = False

break

if done == True:

break

def main():

DEBUG = False

jsonfile = pathlib.Path(\_\_file\_\_).parent.resolve()

jsonfile = str(jsonfile) + "/rules.json"

m = machine(jsonfile)

if DEBUG == False:

m.run()

elif DEBUG == True:

#m.perform\_actions("ask-if-read-sharp-objects")

m.perform\_actions("ask-temp")

m.implications()

if \_\_name\_\_ == "\_\_main\_\_":

main()