МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра ВСТ

Отчёт

по лабораторной работе №2

Реализация дерева вопрос-ответ через структуры данных.

по дисциплине

Основы теории интеллектуальных вычислительных систем

РУКОВОДИТЕЛЬ:

Мисевич П.В.

СТУДЕНТЫ:

Шатов Д.В.

Папанов Р.В.

19-ПО

Работа защищена

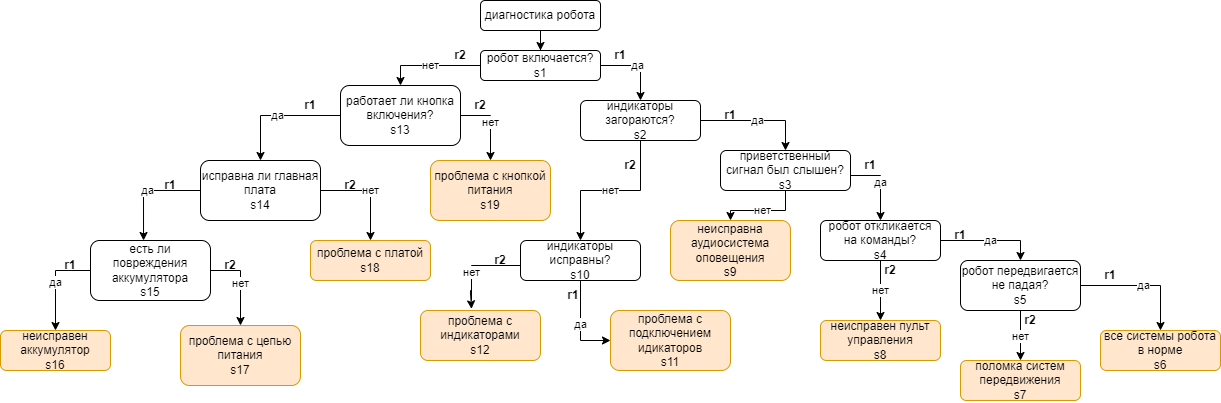
С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

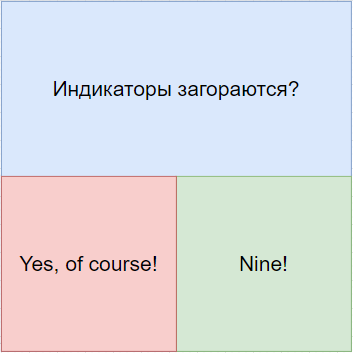
2023

**Цель:** Реализовать дерево вопрос-ответ с помощью структур данных.

**Тема:** Дерево проверки работоспособности систем робота.



**Пример 1:**



**Пример 2:**



**Код программы:**

from tkinter import \*

rules = {

        "S1": ["Робот включается?", {"R1": "S2", "R2": "S13"}],

        "S2": ["Индикаторы загораются?", {"R1": "S3", "R2": "S10"}],

        "S3": ["Приветственный сигнал был слышен?", {"R1": "S4", "R2": "S9"}],

        "S4": ["Робот откликается на команды?", {"R1": "S5", "R2": "S8"}],

        "S5": ["Робот передвигается не падая?", {"R1": "S6", "R2": "S7"}],

        "S6": ["Все системы робота в норме", None],

        "S7": ["Поломка системы передвижения", None],

        "S8": ["Неисправен пульт управления", None],

        "S9": ["Неисправна система оповещения", None],

        "S10": ["Индикаторы исправны?", {"R1": "S11", "R2": "S12"}],

        "S11": ["Проблема с подключением индикаторов", None],

        "S12": ["Проблема с индикаторами", None],

        "S13": ["Работает ли кнопка включения?", {"R1": "S14", "R2": "S19"}],

        "S14": ["Исправна ли главная плата?", {"R1": "S15", "R2": "S18"}],

        "S15": ["Есть ли повреждения аккумулятора?", {"R1": "S16", "R2": "S17"}],

        "S16": ["Неисправен аккумулятор", None],

        "S17": ["Проблема с цепью питания", None],

        "S18": ["Проблема с платой", None],  
 "S19": ["Проблема с кнопкой питания", None],

        }

def main():

    log\_out = Logical\_output(rules, "S1")

    windows = Tk()

    windows.geometry("500x500")

    windows.title("Диагностика систем робота")

    b\_yes = Button(text="Yes, of course!", font=('Arial',28),

                fg="black", bg="blue", command=log\_out.check\_yes)

    b\_yes.place(x=0,y=250,width=250,height=250)

    b\_no = Button(text="Nine", font=('Arial',28),

                fg="black", bg="brown", command=log\_out.check\_no)

    b\_no.place(x=250,y=250,width=250,height=250)

    global label\_question

    label\_question = Message(text=log\_out.rules[log\_out.current\_state][0],

                    font=("Arial",24), width=300)

    label\_question.place(x=100,y=0,width=300,height=200)

    windows.mainloop()

class Logical\_output(object):

    def \_\_init\_\_(self, rules, start):

        self.rules = rules

        self.current\_state = start

    def check(self):

        return self.rules[self.current\_state][1] is None

    def check\_yes(self):

        if not self.check():

            self.current\_state = self.rules[self.current\_state][1]["R1"]

            label\_question["text"] = self.rules[self.current\_state][0]

        else:

            text = f" {self.rules[self.current\_state][0]}!"

            label\_question["text"] = text

    def check\_no(self):

        if not self.check():

            self.current\_state = self.rules[self.current\_state][1]["R2"]

            label\_question["text"] = self.rules[self.current\_state][0]

        else:

            text = f" {self.rules[self.current\_state][0]}!"

            label\_question["text"] = text

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Вывод:** в ходе выполнения данной лабораторной работы было реализовано дерево вопрос-ответ c помощью структур данных. Данный подход намного более расширяем, чем реализация, представленная в первой лабораторной.