

Федеральное государственное автономное образовательное учреждение
высшего образования

«ОМСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Прикладная математика и фундаментальная информатика»

Домашнее задание

по дисциплине «Практикум по программированию»

на тему: «РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ ПРОДУКЦИОННОГО
ТИПА»

Вариант №14

Выполнил:

Студент гр. **ФИТ-212**

группа

Курпенов Куат Ибраимович

Фамилия И.О. студента

подпись

Принял:

Преподаватель:

Фамилия И.О. преподавателя

подпись

Омск 2022

Структура базы знаний

1. Необходимо ли системе запоминать прошлый опыт?
2. Должна ли система воспринимать эмоции людей?
3. Система должна сама формировать понятие о себе?

Пояснения:

1. Нет — реактивный ИИ
2. Да, нет — ИИ с ограничением по памяти
3. Да, да, нет — ИИ с теоретическими основами разума
4. Да, да, да — ИИ, обладающий самосознанием

Схема продукционных правил и заключений

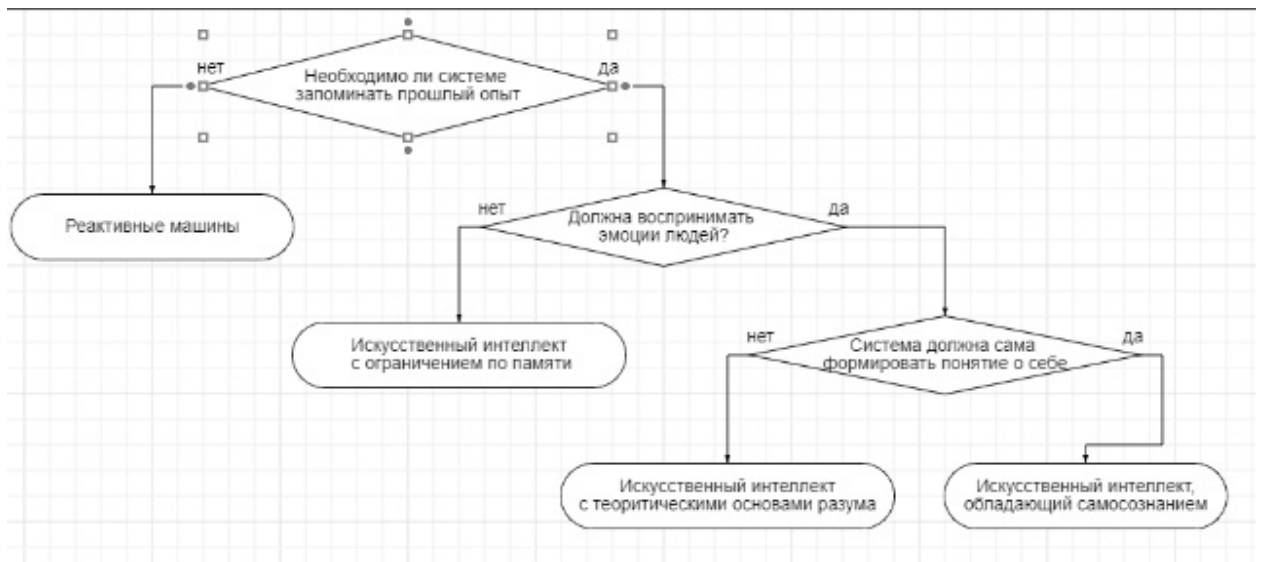


Рисунок 1 — дерево решений

Демонстрация работы

Регистрация	Вопросы
<p>Фамилия</p> <input type="text" value="Курпенов"/>	<p>Необходимо запоминать опыт?</p>
<p>Имя</p> <input type="text" value="Куат"/>	
<p>Отчество</p> <input type="text" value="Ибраимович"/>	
<p>Успешно сохранено</p> <input type="button" value="Сохранить"/>	<input type="button" value="Да"/>
	<input type="button" value="Скорее да"/>
	<input type="button" value="Не знаю"/>
	<input type="button" value="Скорее нет"/>
	<input type="button" value="Нет"/>

Рисунок 2 — вопросы после регистрации

Регистрация

Фамилия

Имя

Отчество

Сохранить

Вопросы

Да

Скорее да

Не знаю

Скорее нет

Нет

i

Данные успешно сохранены

OK

Код программы

```
1 #! /usr/bin/python3
2
3 import sys
4
5 from PySide2 import QtWidgets
6
7 from modules.ui.ui_main import Ui_main_window
8
9 # from modules.configurations import set_text
10
11 from modules.decision import save_user_info
12 from modules.decision import save_answer
13
14
15 class MainWindow(QtWidgets.QMainWindow, Ui_main_window):
16     def __init__(self):
17         super().__init__()
18
19         self.setupUi(self)
20
21         # set_text.setup(self)
22         self.label_save_status.setText("")
23
24         self.push_button_yes.setEnabled(False)
25         self.push_button_rather_yes.setEnabled(False)
26         self.push_button_ignorance.setEnabled(False)
27         self.push_button_rather_no.setEnabled(False)
28         self.push_button_no.setEnabled(False)
29
30         self.push_button_save_user_info.clicked.connect(lambda: save_user_info.save(self))
31
32         self.questions = [
33             "Необходимо запоминать опыт?",
34             "Должна ли воспринимать эмоции людей?",
35             "Должна ли система строить понятие о себе?"
36         ]
37         self.noq = 0
38
39         self.answers = []
40         self.answer = ""
41         self.push_button_yes.clicked.connect(
42             lambda: save_answer.save(self, 0, self.noq + 1))
43         self.push_button_rather_yes.clicked.connect(
44             lambda: save_answer.save(self, 1, self.noq + 1))
45         self.push_button_ignorance.clicked.connect(
46             lambda: save_answer.save(self, 2, self.noq + 1))
47         self.push_button_rather_no.clicked.connect(
48             lambda: save_answer.save(self, 3, self.noq + 1))
49         self.push_button_no.clicked.connect(
50             lambda: save_answer.save(self, 4, self.noq + 1))
51
52         self.show()
53
54
55 if __name__ == "__main__":
56     app = QtWidgets.QApplication(sys.argv)
57     window = MainWindow()
```

Рисунок 4 — ExpertSystem.py

```

1 import i18n
2
3
4 def setup(self):
5     i18n.load_path.append("resources/locale/")
6     i18n.set("locale", "ru")
7
8     self.setWindowTitle(i18n.t("locale.title"))
9
10    self.group_box_registration.setTitle(i18n.t("locale.registration"))
11    self.label_surname.setText(i18n.t("locale.surname"))
12    self.label_name.setText(i18n.t("locale.name"))
13    self.label_patronymic.setText(i18n.t("locale.patronymic"))
14    self.label_save_status.setText(i18n.t("locale.none"))
15    self.push_button_save_user_info.setText(i18n.t("locale.save_user_info"))
16
17    self.group_box_questions.setTitle(i18n.t("locale.questions"))
18    self.push_button_yes.setText(i18n.t("locale.yes"))
19    self.push_button_rather_yes.setText(i18n.t("locale.rather_yes"))
20    self.push_button_ignorance.setText(i18n.t("locale.ignorance"))
21    self.push_button_rather_no.setText(i18n.t("locale.rather_no"))
22    self.push_button_no.setText(i18n.t("locale.no"))
23

```

Рисунок 5 — set_text.py

```

1 from PySide2 import QtWidgets
2
3
4 def save(self) -> None:
5     self.surname = self.line_edit_surname.text()
6     self.name = self.line_edit_name.text()
7     self.patronymic = self.line_edit_patronymic.text()
8
9     if self.surname and self.name and self.patronymic:
10        self.label_save_status.setText("Успешно сохранено")
11
12        self.line_edit_surname.setEnabled(False)
13        self.line_edit_name.setEnabled(False)
14        self.line_edit_patronymic.setEnabled(False)
15
16        self.push_button_save_user_info.setEnabled(False)
17
18        self.text_edit_question.setText(self.questions[0])
19
20        self.push_button_yes.setEnabled(True)
21        self.push_button_rather_yes.setEnabled(True)
22        self.push_button_ignorance.setEnabled(True)
23        self.push_button_rather_no.setEnabled(True)
24        self.push_button_no.setEnabled(True)
25    else:
26        QtWidgets.QMessageBox.critical(self, "Ошибка", "Данные введены некорректно!")
27

```

Рисунок 6 — save_user_info.py

```

1 from modules.decision import clear_data
2
3
4 def save(self, number: int, noq: int) -> None:
5     self.answers.append(number)
6
7     if noq != len(self.questions):
8         self.text_edit_question.setText(self.questions[noq])
9         self.noq = noq
10    else:
11        clear_data.clear(self)
12

```

Рисунок 7 — save_answer.py

```

1 from PySide2 import QtWidgets
2
3 from modules.decision import save_to_docx
4 from modules.decision import save_to_xlsx
5
6
7 def clear(self) -> None:
8     if self.answers in [[0, 4, 2], [1, 4, 2]]:
9         self.answer = "Искусственный интеллект с ограничением по памяти"
10    elif self.answers in [[4, 2, 2], [3, 2, 2], [3, 2, 1], [3, 1, 2], [3, 1, 1]]:
11        self.answer = "Реагирующий ИИ"
12    elif self.answers in [[0, 0, 4], [0, 1, 4], [1, 0, 4], [0, 0, 3], [0, 1, 3], [1, 0, 3]]:
13        self.answer = "ИИ с теоретическими основами разума"
14    elif self.answers in [[0, 0, 0], [0, 0, 1], [0, 1, 0], [0, 1, 1], [1, 0, 0], [1, 0, 1], [1, 1, 0], [1, 1, 1]]:
15        self.answer = "ИИ, обладающий самосознанием"
16    else:
17        self.answer = "Ваши цели за пределами моего понимания"
18
19    self.line_edit_surname.setText("")
20    self.line_edit_name.setText("")
21    self.line_edit_patronymic.setText("")
22
23    self.label_save_status.setText("")
24
25    self.push_button_save_user_info.setEnabled(True)
26
27    self.text_edit_question.setText("")
28    self.noq = 0
29    self.answers = []
30
31    self.push_button_yes.setEnabled(False)
32    self.push_button_rather_yes.setEnabled(False)
33    self.push_button_ignorance.setEnabled(False)
34    self.push_button_rather_no.setEnabled(False)
35    self.push_button_no.setEnabled(False)
36
37    self.line_edit_surname.setEnabled(True)
38    self.line_edit_name.setEnabled(True)
39    self.line_edit_patronymic.setEnabled(True)
40
41    save_to_docx.save(self)
42    save_to_xlsx.save(self)
43
44    QtWidgets.QMessageBox.information(self, "Успешно", "Данные успешно сохранены")
45

```

Рисунок 8 — clear_data.py


```

1 import os
2
3 from docx import Document
4
5
6 def save(self) -> None:
7     wd = os.getcwd()
8
9     if not os.path.exists(f"{wd}/reports/"):
10         os.mkdir(f"{wd}/reports/")
11
12
13     if os.path.exists(f"{wd}/reports/Report_{self.surname}_{self.name}.docx"):
14         return
15
16     document = Document()
17
18     document.add_heading("Report", 0)
19     document.add_paragraph(f"{self.surname} {self.name} {self.patronymic}: {self.answer}")
20     document.save(f"reports/Report_{self.surname}_{self.name}.docx")
21

```

Рисунок 9 — save_to_docx.py

```

1 import os
2
3 import pylightxl
4
5
6 def save(self) -> None:
7     wb = os.getcwd()
8     line = 1
9
10     if not os.path.exists(f"{wb}/reports/"):
11         os.mkdir(f"{wb}/reports/")
12
13     if os.path.exists(f"{wb}/reports/Report.xlsx"):
14         db = pylightxl.readxl("reports/Report.xlsx")
15
16         while True:
17             output = db.ws(ws="Лист1").index(row=line, col=1)
18             if not output:
19                 break
20             line += 1
21     else:
22         db = pylightxl.Database()
23         db.add_ws(ws="Лист1")
24
25     db.ws(ws="Лист1").update_index(row=line, col=1, val=self.surname)
26     db.ws(ws="Лист1").update_index(row=line, col=2, val=self.name)
27     db.ws(ws="Лист1").update_index(row=line, col=3, val=self.patronymic)
28     db.ws(ws="Лист1").update_index(row=line, col=4, val=self.answer)
29
30     pylightxl.writexl(db=db, fn="reports/Report.xlsx")
31

```

Рисунок 10 — save_to_xlsx.py


```
1 ru:
2   title: "Экспертная система"
3
4   registration: "Регистрация"
5   surname: "Фамилия"
6   name: "Имя"
7   patronymic: "Отчество"
8   none: ""
9   success: "Успешно сохранено"
10  save_user_info: "Сохранить"
11
12  questions: "Вопросы"
13  yes: "Да"
14  rather_yes: "Скорее да"
15  ignorance: "Не знаю"
16  rather_no: "Скорее нет"
17  no: "Нет"
18
```

Рисунок 11 — locale.ru.yml

```
1 lxml==4.9.0
2 pylightxl==1.59
3 PySide2==5.15.2.1
4 python-docx==0.8.11
5 python-i18n==0.3.9
6 PyYAML==6.0
7 shiboken2==5.15.2.1
```

Рисунок 12 — requirements.txt

