

Липецкий государственный технический университет
Факультет автоматизации и информатики
Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №1
Вариант 8
по системам искусственного интеллекта
Разработка экспертной системы

Студент
Группа АИ-19-1

Первушин О. С.

Руководитель
Доцент

Кургасов В. В.

Липецк 2022 г.

Цель работы

Получение навыков проектирования и разработки экспертной системы на всех этапах ее создания.

Задание кафедры

Отработать этапы разработки экспертной системы для решения задачи (проблемы) выбора. Осуществить программную реализацию экспертной системы на любом языке программирования. Разрабатываемая экспертная система относится к классу поверхностных демонстрационных (учебных) систем. Поверхностные ЭС представляют знания в виде правил (условие – действие). Тема по варианту – выбор оружия для самообороны.

Ход работы

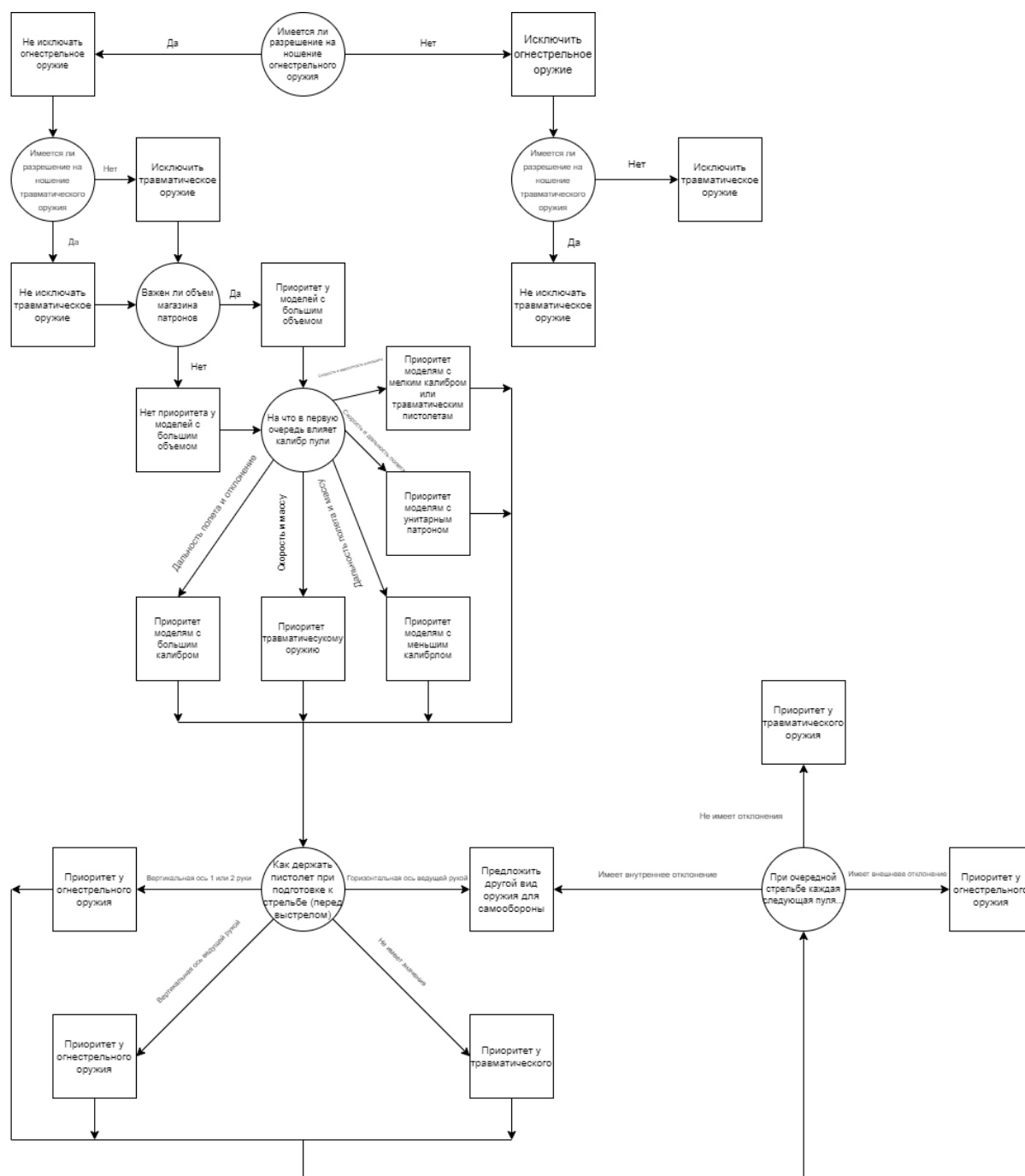


Рисунок 1 — Дерево решений

База знаний

ЕСЛИ наличие разрешение на ношение огнестрельного оружия = да, ТО не исключаем из списка огнестрельное оружие.

ЕСЛИ наличие разрешение на ношение огнестрельного оружия = нет, ТО исключаем из списка огнестрельное оружие.

ЕСЛИ наличие разрешение на ношение травматического оружия = да, ТО не исключаем из списка травматическое оружие.

ЕСЛИ наличие разрешение на ношение травматического оружия = нет, ТО исключаем из списка травматическое оружие.

ЕСЛИ наличие разрешение на ношение холодного оружия = да, ТО не исключаем из списка холодное оружие.

ЕСЛИ наличие разрешение на ношение холодного = нет, ТО исключаем из списка холодное оружие.

ЕСЛИ наличия боязни вида крови = нет, ТО не исключаем возможность ношения оружия для самообороны.

ЕСЛИ наличия боязни вида крови = да, ТО исключаем возможность ношения оружия для самообороны.

ЕСЛИ хватка пистолета = вертикальная ось, ТО не исключаем из списка стреляющее оружие

ЕСЛИ хватка пистолета = горизонтальная ось, ТО исключаем из списка стреляющее оружие

ЕСЛИ назначение гарды = ограничитель, ТО приоритет у кинжала видных ножей

ЕСЛИ назначение гарды = защитная функция, ТО приоритет у охотничьих ножей

ЕСЛИ назначение гарды = эстетическая роль, ТО приоритет у клиновидных ножей.

ЕСЛИ объем магазина патронов важен = да, ТО приоритет у моделей с большим объемом магазина

ЕСЛИ объем магазина патронов важен = нет, ТО нет приоритет у моделей с большим объемом магазина

ЕСЛИ влияние калибра пули = Дальность полета и отклонение, ТО приоритет у моделей с большим калибром

ЕСЛИ влияние калибра пули = Скорость и массу, ТО приоритет у травматического оружия

ЕСЛИ влияние калибра пули = Дальность полета и массу, ТО приоритет у моделей с меньшим калибром

ЕСЛИ влияние калибра пули = Скорость и дальность полета, ТО приоритет у моделей с унитарным калибром (7,62*25 для ТТ или .50 Magnum для Desert Eagle)

ЕСЛИ влияние калибра пули = Скорость и вероятность рикошета, ТО приоритет мелкокалиберным пистолетам или травматическим моделям

ЕСЛИ при очередной стрельбе каждая следующая пуля имеет отклонения = внешнее, ТО приоритет у огнестрельного оружия

ЕСЛИ при очередной стрельбе каждая следующая пуля имеет отклонения = внутреннее, ТО приоритет у холодного оружия

ЕСЛИ при очередной стрельбе каждая следующая пуля имеет отклонения = не имеет, ТО приоритет у травматического оружия

ЕСЛИ район проживания густонаселенный = да, ТО приоритет у моделей с малым калибром или травматических моделей

ЕСЛИ район проживания густонаселенный = нет, ТО нет приоритета у моделей с малым калибром или травматических моделей

Программная реализация

```
using System;
using System.Windows.Forms;

namespace SystemOfWeaponts
{
    internal static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MainForm());
        }
    }
}

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace SystemOfWeaponts
{
    public partial class MainForm : Form
    {
        private Weaponts weaponts;
        private Answers answers;
        private List<ComboBox> combos = new List<ComboBox>();
        private int[] selected = new int[10];
        public MainForm()
        {
            InitializeComponent();
        }
    }
}
```

```

        weaponts = new Weaponts();
        answers = new Answers();
        SetUp();
    }

    public void SetUp()
    {
        combos.Add(comboBox1);
        combos.Add(comboBox2);
        combos.Add(comboBox3);
        combos.Add(comboBox4);
        combos.Add(comboBox5);
        combos.Add(comboBox6);
        combos.Add(comboBox7);
        combos.Add(comboBox8);
        combos.Add(comboBox9);
        combos.Add(comboBox10);

        for (int i = 0; i < answers.answers.Count; i++)
        {
            combos[answers.answers[i].QuestID -
1].Items.Add(answers.answers[i].Text);
        }
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox1.SelectedIndex + 0;
        selected[0] = index;
    }

    private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox2.SelectedIndex + 5;
        selected[1] = index;
    }

    private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox3.SelectedIndex + 9;
        selected[2] = index;
    }

    private void comboBox4_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox4.SelectedIndex + 12;
        selected[3] = index;
    }

    private void comboBox5_SelectedIndexChanged(object sender, EventArgs e)
    {

```

```

        int index = comboBox5.SelectedIndex + 14;
        selected[4] = index;
    }

    private void comboBox6_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox6.SelectedIndex + 16;
        selected[5] = index; ;
    }

    private void comboBox7_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox7.SelectedIndex + 18;
        selected[6] = index;
    }

    private void comboBox8_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox8.SelectedIndex + 20;
        selected[7] = index; ;
    }

    private void comboBox9_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox9.SelectedIndex + 24;
        selected[8] = index;
    }

    private void comboBox10_SelectedIndexChanged(object sender, EventArgs e)
    {
        int index = comboBox10.SelectedIndex + 26;
        selected[9] = index;
    }

    private void OkButton_Click(object sender, EventArgs e)
    {
        for(int i = 0; i < selected.Length; i++)
        {
            answers.answers[selected[i]].Enter(weaponTs);
        }

        string result = "";

        foreach (var item in weaponTs.weaps)
        {
            result += item.ToString() + Environment.NewLine;
        }

        MessageBox.Show(result, "Результат");
    }

```



```

    }
}

using System;

namespace SystemOfWeaponts
{
    [Serializable]
    public class Weapont
    {
        public int ID { get; private set; }
        public string Name { get; private set; }
        public double Probability { get; private set; }

        public Weapont(int id, string name, double probability)
        {
            ID = id;
            Name = name;

            if (probability < 0 || probability > 1)
                throw new Exception("Probability is invalide (001)");

            Probability = probability;
        }

        public void ChangeProbability(double delta)
        {
            if(delta < -1 || delta > 1)
                throw new Exception("Change of Probability is invalid (002)");

            Probability += delta;

            if (Probability > 1)
                Probability = 1;
            if(Probability < 0)
                Probability = 0;
        }

        public override string ToString()
        {
            return Name + " - " + Probability * 100 + "%";
        }
    }
}

using System;
using System.Collections.Generic;

namespace SystemOfWeaponts
{

```

```

[Serializable]
public class Answer
{
    public int ID { get; private set; }
    public int QuestID {get; private set;}
    public string Text { get; private set; }
    public List<double> Effects { get; private set; }

    public Answer(int id, int questID, string text, List<double> effects)
    {
        ID = id;
        QuestID = questID;
        Text = text;
        Effects = effects;
    }

    public void Enter(Weaponts weaponts)
    {
        for (int i = 0; i < weaponts.weaps.Count; i++)
        {
            weaponts.weaps[i].ChangeProbability(Effects[i]);
        }
    }
}

using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.IO;

namespace SystemOfWeaponts
{
    [Serializable]
    public class Weaponts
    {
        public List<Weapon> weaps { get; private set; }

        public Weaponts()
        {
            string json;
            FileStream fileStream = new FileStream("weaponts.json",
FileMode.Open);
            StreamReader streamReader = new StreamReader(fileStream);
            json = streamReader.ReadToEnd();
            streamReader.Close();
            fileStream.Close();
            weaps = JsonConvert.DeserializeObject<List<Weapon>>(json);
        }
    }
}

```

```
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.IO;

namespace SystemOfWeaponts
{
    [Serializable]
    public class Answers
    {
        public List<Answer> answers { get; private set; }

        public Answers()
        {
            string json;
            FileStream fileStream = new FileStream("answers.json", FileMode.Open);
            StreamReader streamReader = new StreamReader(fileStream);
            json = streamReader.ReadToEnd();
            streamReader.Close();
            fileStream.Close();
            answers = JsonConvert.DeserializeObject<List<Answer>>(json);
        }
    }
}
```

MainForm

1. На что в первую очередь влияет калибр пули?

Дальность полета и отклонение

2. Как держать пистолет при подготовке к стрельбе (перед выстрелом)?

В вертикальной оси одной или двумя руками

3. При очередной стрельбе каждая следующая пуля...

Имеет внешнее отклонение от оси оружия

4. В каких районах вы чаще всего находитесь?

Густонаселенных, местах скопления людей

5. Имеете ли Вы разрешение на ношение огнестрельного оружия?

Да

6. Имеете ли Вы разрешение на травматическое оружие?

Нет

7. Важен ли для Вас объем магазина?

Да

8. Назначение гарды на ноже

Исполняет роль ограничителя

9. Боитесь ли Вы вида крови

Нет

10. Имеете ли разрешение на ношение холодного оружия

Нет

Ok

Рисунок 2 — Пример заполнения

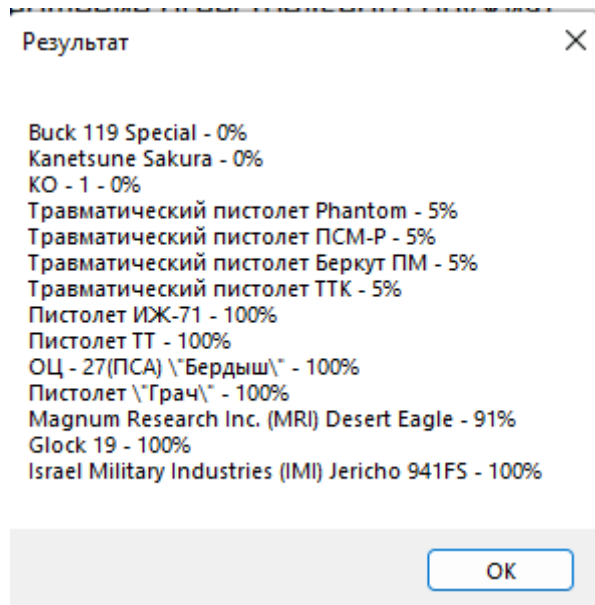


Рисунок 3 — Пример результата

Требования к системе

ОС: Windows 10x64/Windows 11

Процессор: 4-х ядерный intel core i3-8100 и старше

Наличие .NET Framework 2.0/3.5/4.0/4.8

Вывод

В ходе выполнения лабораторной работы получила навыки проектирования и разработки экспертной системы на всех этапах ее создания.