**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

Лабораторная работа №1

по дисциплине

«Системы искусственного интеллекта»

Разработка экспертной системы

Студент Корноухов Н. И.

Группа АС-19

Руководитель

доцент Кургасов В. В.

Липецк 2022 г.

Цель работы

Создание экспертной системы для решения проблемы выбора ноутбука.



Рисунок 1 – Дерево решений

Таблица 1 – Имена переменных

|  |  |  |
| --- | --- | --- |
| Имя переменной | Условия | Вершина(ы) |
| Activity | Планируемая деятельность | 1 |
| HomeUsage | Предполагается домашнее использование или в дороге? | 2 |
| MacUser | Вы владелец устройств от Apple? | 3, 10 |
| TargetDevice | Предложенное устройство | 4, 7, 8, 11, 12, 18, 21, 24, 25, 27, 30, 31, 34, 37, 38 |
| NoMac | Не владелец Apple ус-в | 5 |
| NightUsage | Устройство будет использоваться ночью? | 6 |
| Compact | Необходимо легкое, портативное ус-во, с емким аккумулятором | 9 |
| Powerful | Необходимо мощное устройство | 13 |
| UsageType | Как именно будет использоваться устройство? | 14 |
| JobWithGraphics | Работа связана с графикой? | 15 |
| TrueColorMatrix | Необходима матрица с широким цветовым охватом | 16 |
| JobWith3D | Вы занимаетесь 3D-моделированием, созданием сложных сцен? | 17 |
| NoMoreVram | Нет необходимости в большой объеме VRAM | 19 |
| JobWithPhotography | Вы фотограф? | 20 |
| NoCardReader | Кардридер не обязателен | 22 |
| JobWithStylus | Вы художник? | 23 |
| JobWithCode | Вы занимаетесь разработкой ПО? | 26 |
| NeedGoodCpuAndRam | Необходим хороший CPU И большой объем RAM | 28 |
| DevForMac | Разработка под устройства Apple? | 29 |
| QuicklyMatrix | Необходима высокоскоростная матрица | 32 |
| GameType | В каких видеоиграх планируется использовать ус-во? | 33 |
| FpsPriority | Количество кадров в секунду в приоритете | 35 |
| CyberSport | Вы предпочитаете FPS киберспорт? | 36 |

Таблица 2 – База знаний

|  |  |  |
| --- | --- | --- |
| № | Правило | Путь |
| 1 | ЕСЛИ Activity = Повседневная И HomeUsage = Домашнее И MacUser = Да, ТО TargetDevice = Apple MacBook Air (M1-7C) | 4, 3, 2, 1 |
| 2 | ЕСЛИ Activity = Повседневная И HomeUsage = Домашнее И MacUser = Да, ТО NoMac = Да | 5, 3, 2, 1 |
| 3 | ЕСЛИ NoMac = Да И NightUsage = Да, ТО TargetDevice = ASUS A516JA-BQ1918 | 8, 6, 5 |
| 4 | ЕСЛИ NoMac = Да И NightUsage = Нет, ТО TargetDevice = Honor MagicBook X15 BBR-WAH9 | 7, 6, 5 |
| 5 | ЕСЛИ Activity = Повседневная И HomeUsage = В дороге, ТО Compact = Да | 9, 2, 1 |
| 6 | ЕСЛИ Compact = Да И MacUser = Да, ТО TargetDevice = Apple MacBook Air (M1-7C) | 11, 10, 9 |
| 7 | ЕСЛИ Compact = Да И MacUser = Нет, ТО TargetDevice = Honor MagicBook X14 NBR-WAI | 12, 10, 9 |
| 8 | ЕСЛИ Activity = Ресурсоемкая, ТО Powerful = Да | 13, 1 |
| 9 | ЕСЛИ Powerful = Да И UsageType = Проф. деятельность И JobWithGraphics = Да, ТО TrueColorMatrix = Да | 16, 15, 14, 13 |
| 10 | ЕСЛИ TrueColorMatrix = Да И JobWith3D = Да, ТО TargetDevice = ASUS ProArt Studiobook Pro 16 OLED | 18, 17, 16 |
| 11 | ЕСЛИ TrueColorMatrix = Да И JobWith3D = Нет, ТО NoMoreVram = Да | 19, 17, 16 |
| 12 | ЕСЛИ NoMoreVram = Да И JobWithPhotography = Да, ТО TargetDevice = Apple MacBook Pro 14 Late 2021 | 21, 20, 19 |
| 14 | ЕСЛИ NoMoreVram = Да И JobWithPhotography = Нет, ТО NoCardReader = Да | 22, 20, 19 |
| 15 | ЕСЛИ NoCardReader = Да И JobWithStylus = Да, ТО TargetDevice = Lenovo Yoga 9 15IMH5 | 24, 23, 22 |
| 16 | ЕСЛИ NoCardReader = Да И JobWithStylus = Нет, ТО TargetDevice = Apple MacBook Pro 14 Late 2021 | 25, 23, 22 |
| 17 | ЕСЛИ Powerful = Да И UsageType = Проф. деятельность И JobWithGraphics = Нет И JobWithCode = Нет, ТО TargetDevice = Xiaomi RedmiBook 14 | 27, 26, 15, 14, 13 |
| 18 | ЕСЛИ Powerful = Да И UsageType = Проф. деятельность И JobWithGraphics = Нет И JobWithCode = Да, ТО NeedGoodCpuAndRam = Да | 28, 26, 15, 14, 13 |
| 19 | ЕСЛИ NeedGoodCpuAndRam = Да И DevForMac = Да, ТО TargetDevice = Apple MacBook Pro 14 Late 2021 | 30, 29, 28 |
| 20 | ЕСЛИ NeedGoodCpuAndRam = Да И DevForMac = Нет, ТО TargetDevice = Honor MagicBook 16 | 31, 29, 28 |
| 21 | ЕСЛИ Powerful = Да И UsageType = Видеоигры, ТО QuicklyMatrix = Да | 32, 14, 13 |
| 22 | ЕСЛИ QuicklyMatrix = Да И GameType = Крупные проекты, ТО TargetDevice = ASUS TUF Gaming A15 FA507RR-HQ007 | 34, 33, 32 |
| 23 | ЕСЛИ QuicklyMatrix = Да И GameType = Киберспорт/Инди, ТО FpsPriority = Да | 35, 33, 32 |
| 24 | ЕСЛИ FpsPriority = Да И CyberSport = Да, ТО TargetDevice = ASUS TUF Gaming A15 FA507RM-HF043 | 37, 36, 35 |
| 25 | ЕСЛИ FpsPriority = Да И CyberSport = Нет, ТО TargetDevice = ASUS TUF Gaming A15 FA507RE-HN021 | 38, 36, 35 |

Таблицы структур данных

Таблица 3 – Структура класса Answer

|  |  |  |
| --- | --- | --- |
| Тип переменной | Название переменной | Описание переменной |
| string | text | Текст ответа |
| int | nextQuestion | Индекс класса следующего вопроса |
| int | targetId | Индекс класса результата |

Таблица 4 – Структура класса Quest

|  |  |  |
| --- | --- | --- |
| Тип переменной | Название переменной | Описание переменной |
| string | text | Текст вопроса |
| Answer[] | answers | Массив ответов на данный вопрос |

Таблица 5 – Структура класса Target

|  |  |  |
| --- | --- | --- |
| Тип переменной | Название переменной | Описание переменной |
| string | name | Название и модель устройства |
| Bitmap | img | Изображение устройства |
| string | link | Ссылка на устройство |

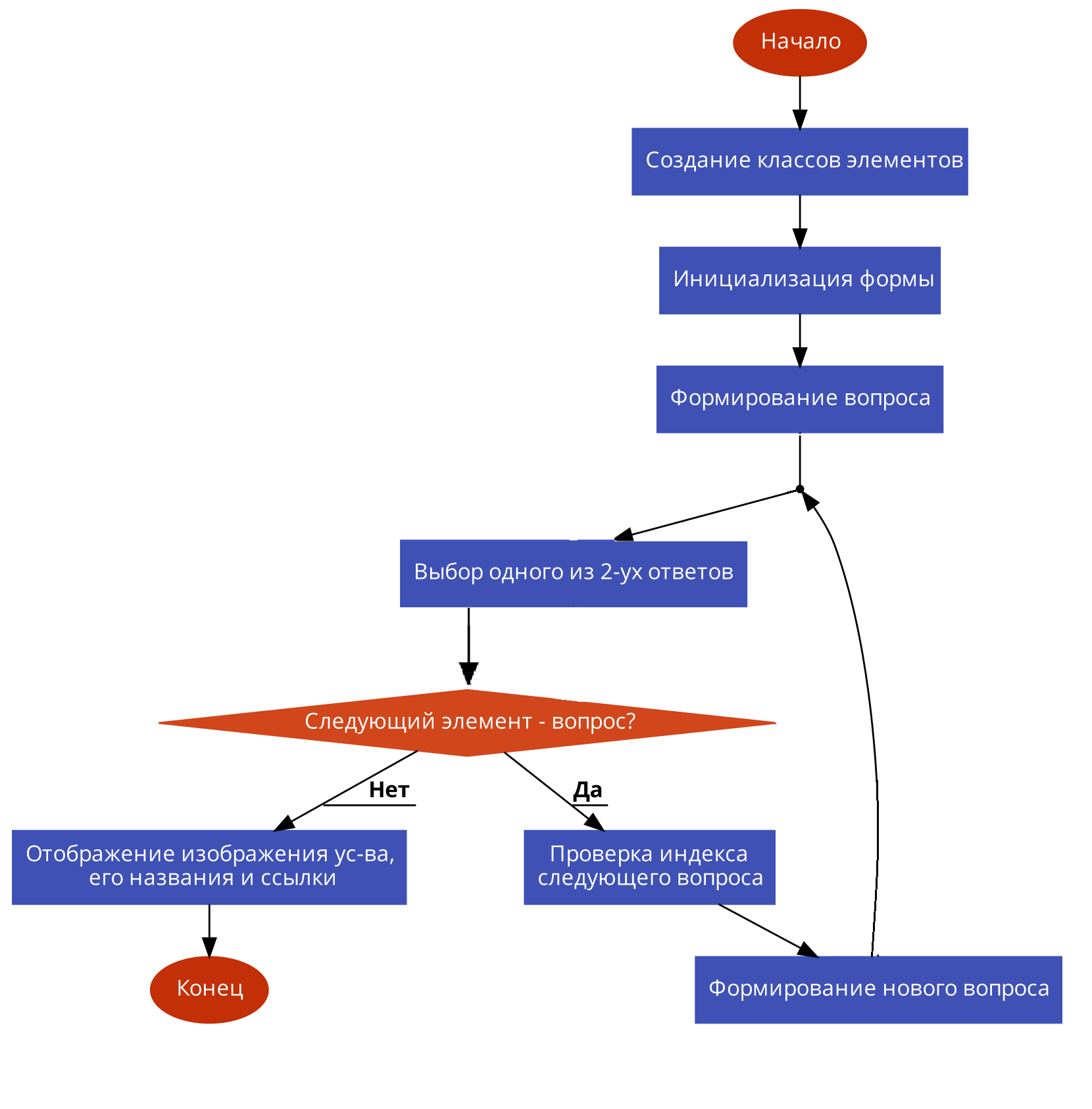


Рисунок 2 – Блок схема алгоритма программы

Программная реализация ЭС

Программа представляет собой интерфейс, состоящий из вопроса, ответов на него, истории ответов и кнопки начала подбора с начала. При выборе одного из ответов, подгружается новый вопрос и ответы, а также заносится запись в историю. Если ответ ведет к итоговому результату, то подгружается изображение устройства, его название и ссылка, для оценки его характеристик и стоимости.

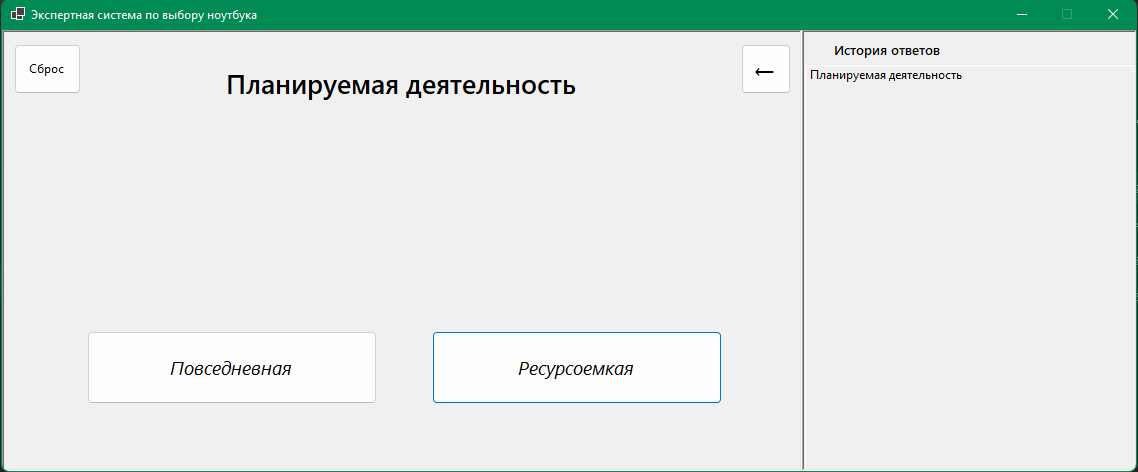


Рисунок 3 – Интерфейс программы



Рисунок 4 – Подобранное устройство

Код программы:

using AI\_Lab1.Properties;

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

namespace AI\_Lab1

{

public partial class MainForm : Form

{

public MainForm()

{

fillingData();

InitializeComponent();

label\_question.Text = questions[currentQuestId].text;

button\_answer1.Text = questions[currentQuestId].answers[0].text;

button\_answer2.Text = questions[currentQuestId].answers[1].text;

textBox\_history.Text += questions[currentQuestId].text;

}

List<Quest> questions = new List<Quest>();

List<Target> targets = new List<Target>();

bool historyPanelEnabled = true;

int currentQuestId = 0;

public class Answer

{

public string text = "";

public int nextQuestion = 0;

public int targetId = -1;

}

public class Quest

{

public string text = "";

public Answer[] answers = new Answer[2];

}

public class Target

{

public string name = "";

public Bitmap img;

public string link = "";

}

private void buttonToggleHistory\_Click(object sender, EventArgs e)

{

if (historyPanelEnabled)

{

panel\_history.Visible = false;

panel\_history.Enabled = false;

button\_toggleHistory.Text = "→ ";

historyPanelEnabled = false;

panel\_history.Width = 0;

this.Width = 815;

}

else

{

panel\_history.Visible = true;

panel\_history.Enabled = true;

button\_toggleHistory.Text = "← ";

historyPanelEnabled = true;

panel\_history.Width = 336;

this.Width = 1150;

}

}

public void fillingData()

{

string[] questionsArray = Resources.Questions.Split(Environment.NewLine);

Quest quest = new Quest();

quest.answers[0] = new Answer();

quest.answers[1] = new Answer();

questions.Add(quest);

int propPos = -1;

foreach (string line in questionsArray)

{

switch (propPos)

{

case 0:

questions[questions.Count - 1].text = line;

propPos++;

break;

case 1:

questions[questions.Count - 1].answers[0].text = line;

propPos++;

break;

case 2:

questions[questions.Count - 1].answers[0].nextQuestion = Convert.ToInt32(line);

propPos++;

break;

case 3:

questions[questions.Count - 1].answers[0].targetId = Convert.ToInt32(line);

propPos++;

break;

case 4:

questions[questions.Count - 1].answers[1].text = line;

propPos++;

break;

case 5:

questions[questions.Count - 1].answers[1].nextQuestion = Convert.ToInt32(line);

propPos++;

break;

case 6:

questions[questions.Count - 1].answers[1].targetId = Convert.ToInt32(line);

propPos++;

break;

default:

propPos++;

break;

}

if (line == "")

{

quest = new Quest();

quest.answers[0] = new Answer();

quest.answers[1] = new Answer();

questions.Add(quest);

propPos = -1;

}

}

questionsArray = Resources.Target.Split(Environment.NewLine);

Target target = new Target();

targets.Add(target);

propPos = -1;

foreach (string line in questionsArray)

{

switch (propPos)

{

case 0:

targets[targets.Count - 1].name = line;

propPos++;

break;

case 1:

targets[targets.Count - 1].link = line;

propPos++;

break;

default:

propPos++;

break;

}

if (line == "")

{

targets[targets.Count - 1].img = Resources.ResourceManager.GetObject((targets.Count - 1).ToString()) as Bitmap;

targets.Add(target = new Target());

propPos = -1;

}

}

targets[targets.Count - 1].img = Resources.ResourceManager.GetObject((targets.Count - 1).ToString()) as Bitmap;

}

private void button\_answer\_Click(object sender, EventArgs e)

{

if ((sender as Button).Text == questions[currentQuestId].answers[0].text)

{

textBox\_history.Text += Environment.NewLine + questions[currentQuestId].answers[0].text + Environment.NewLine;

if (questions[currentQuestId].answers[0].nextQuestion != -1)

currentQuestId = questions[currentQuestId].answers[0].nextQuestion;

else

{

pictureBox\_targetImg.Image = targets[questions[currentQuestId].answers[0].targetId].img;

label\_targetName.Text = targets[questions[currentQuestId].answers[0].targetId].name;

linkLabel\_target.Text = targets[questions[currentQuestId].answers[0].targetId].link;

panel\_target.Enabled = true;

panel\_target.Visible = true;

return;

}

}

else

{

textBox\_history.Text += Environment.NewLine + questions[currentQuestId].answers[1].text + Environment.NewLine;

if (questions[currentQuestId].answers[1].nextQuestion != -1)

currentQuestId = questions[currentQuestId].answers[1].nextQuestion;

else

{

pictureBox\_targetImg.Image = targets[questions[currentQuestId].answers[1].targetId].img;

label\_targetName.Text = targets[questions[currentQuestId].answers[1].targetId].name;

linkLabel\_target.Text = targets[questions[currentQuestId].answers[1].targetId].link;

panel\_target.Enabled = true;

panel\_target.Visible = true;

return;

}

}

label\_question.Text = questions[currentQuestId].text;

button\_answer1.Text = questions[currentQuestId].answers[0].text;

button\_answer2.Text = questions[currentQuestId].answers[1].text;

textBox\_history.Text += Environment.NewLine + questions[currentQuestId].text;

}

private void linkLabel\_target\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

Clipboard.SetText(linkLabel\_target.Text);

}

public void questionPanelClear()

{

currentQuestId = 0;

label\_question.Text = questions[currentQuestId].text;

button\_answer1.Text = questions[currentQuestId].answers[0].text;

button\_answer2.Text = questions[currentQuestId].answers[1].text;

textBox\_history.Clear();

textBox\_history.Text += questions[currentQuestId].text;

}

private void button\_reset1\_Click(object sender, EventArgs e)

{

questionPanelClear();

}

private void button\_reset2\_Click(object sender, EventArgs e)

{

questionPanelClear();

panel\_target.Visible = false;

panel\_target.Enabled = false;

}

}

}