Государственное образовательное учреждение

высшего профессионального образования

«Донецкий национальный технический университет»

Кафедра АСУ

ОТЧЕТ

по лабораторной работе № 3

по дисциплине

«МЕТОДЫ И СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

По теме «Построение системы идентификации объекта по двум характеристическим признакам»

Выполнил:

студент группы ИС-17 (а)

Канатуш Сергей

Проверили:

Секирин А. И.

Новиков Д. Д.

Донецк – 2020

Цель: построить систему идентификации объекта по двум характеристическим признакам

Задание: построить обучающуюся систему идентификации объекта по двум характеристическим признакам.

Пояснения по ходу выполнения работы:

В ходе работы необходимо учитывать следующие особенности функционирования данной системы:

1. При каждом сеансе идентификации вопросы задаются пользователю в случайном порядке.
2. Окончание идентификации происходит при выполнении только двух условий:
   1. Получен второй положительный ответ.
   2. Исчерпаны все вопросы.

При этом в случае

**а)** объект или идентифицируется, или если в базе знаний нет идентифицируемого объекта, то пользователь должен ввести название этого объекта.

В случае

**б)** возможны два исхода:

1-й Если был получен всего лишь один положительный ответ.

2-й Если не было ни одного положительного ответа.

При этом пользователь вводит не только название объекта идентификации, но и необходимое количество новых идентифицирующих вопросов.

ЛИСТИНГ ПРОГРАММЫ:

**Работа с БД:**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.Windows.Forms;

namespace ii\_lab\_3

{

public class DBHandler

{

DBHelper dbHelper = new DBHelper();

public DBHandler()

{

dbHelper.Open();

}

~DBHandler()

{

dbHelper.Close();

}

public List<FeatureBean> Write(string table, string field)

{

DataTable dataTable = GetDataTable(table);

if (dataTable != null)

{

List<FeatureBean> list = new List<FeatureBean>();

for (int i = 0; i < dataTable.Rows.Count; i++)

{

list.Add(new FeatureBean (

dataTable.Rows[i].Field<int>("id"),

dataTable.Rows[i].Field<string>(field)));

}

return list;

}

return null;

}

public DataTable GetDataTable(String table)

{

try

{

DataTable dataTable = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM " + table, dbHelper.getMySqlConnection());

adapter.SelectCommand = command;

adapter.Fill(dataTable);

return dataTable;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

return null;

}

}

public string CheckObject(int[] arr\_otv)

{

DataTable dataTable = GetDataTable(DBHelper.TABLE\_DEPENDECES);

DataTable dataTableObjects = GetDataTable(DBHelper.TABLE\_OBJECTS);

for (int i = 0; i < dataTableObjects.Rows.Count; i++)

{

string filter = "(";

for (int j = 0; j < arr\_otv.Length; j++)

{

filter += (DBHelper.FIELD\_DEPENDECES\_FEATURE\_ID + " = " + arr\_otv[j]);

if (j != arr\_otv.Length - 1) filter += " OR ";

else filter += (") AND " + DBHelper.FIELD\_DEPENDECES\_OBJECT\_ID + " = " + dataTableObjects.Rows[i].Field<int>(DBHelper.FIELD\_OBJECTS\_ID));

}

DataRow[] dataRows = dataTable.Select(filter);

if (dataRows.Length >= 2)

{

return dataTableObjects.Rows[i].Field<string>(DBHelper.FIELD\_OBJECTS\_OBJECT);

}

}

return null;

}

public void Insert(string[] features, string \_object, int[] arr\_otv)

{

int obj\_id = Insert(DBHelper.TABLE\_OBJECTS, new string[] { DBHelper.FIELD\_OBJECTS\_OBJECT }, new string[] { \_object }, new MySqlDbType[] { MySqlDbType.VarChar });

int feat\_id, l = 0;

for (int i = 0; i < features.Length; i++)

{

if (features[i] != null) {

feat\_id = Insert(DBHelper.TABLE\_FEATURES, new string[] { DBHelper.FIELD\_FEATURES\_FEATURE }, new string[] { features[i] }, new MySqlDbType[] { MySqlDbType.VarChar });

} else {

feat\_id = arr\_otv[l];

l++;

}

Insert(DBHelper.TABLE\_DEPENDECES, new string[] { DBHelper.FIELD\_DEPENDECES\_OBJECT\_ID, DBHelper.FIELD\_DEPENDECES\_FEATURE\_ID },

new string[] { obj\_id.ToString(), feat\_id.ToString() }, new MySqlDbType[] { MySqlDbType.Int32, MySqlDbType.Int32 });

}

}

public int Insert(String table, String[] columns, String[] adds, MySqlDbType[] types)

{

try

{

string columns\_str = "";

string val\_str = "";

for (int i = 0; i < adds.Length; i++)

{

columns\_str += (columns[i]);

val\_str += ("@add" + i.ToString());

if (i != adds.Length - 1)

{

columns\_str += ", ";

val\_str += ", ";

}

}

string sql = "INSERT INTO " + table + " (" + columns\_str + ") VALUES (" + val\_str + ")";

MySqlCommand command = new MySqlCommand(sql, dbHelper.getMySqlConnection());

for (int i = 0; i < adds.Length; i++)

{

command.Parameters.Add("@add" + i.ToString(), types[i]).Value = adds[i];

}

command.ExecuteNonQuery();

return getLastId(table);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

return -1;

}

}

int getLastId(String table)

{

try

{

DataTable dataTable = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM " + table + " ORDER BY id DESC LIMIT 1", dbHelper.getMySqlConnection());

adapter.SelectCommand = command;

adapter.Fill(dataTable);

return dataTable.Rows[0].Field<int>("id");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

return -1;

}

}

}

}

**Основной класс:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ii\_lab\_3

{

public partial class Form1 : Form

{

DBHandler dbHandler = new DBHandler();

List<FeatureBean> features; // вопросы

Thread thread;

bool is\_yes = false;

public delegate void ChangeFeature(string text, bool isEnd);

public ChangeFeature changeFeatureDelegate;

private ManualResetEvent mre = new ManualResetEvent(false);

public Form1()

{

InitializeComponent();

changeFeatureDelegate = new ChangeFeature(ChangeFeatureOnForm);

}

public void ChangeFeatureOnForm(string text, bool isEnd)

{

feature\_label.Text = text;

if (isEnd)

{

this.yes.Enabled = false;

this.no.Enabled = false;

}

}

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

{

this.yes.Enabled = true;

this.no.Enabled = true;

thread = new Thread(RunSystem);

thread.Start();

}

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

{

thread.Abort();

thread = null;

this.yes.Enabled = false;

this.no.Enabled = false;

feature\_label.Text = "НАЖМИТЕ 'СТАРТ' ДЛЯ НАЧАЛА!";

}

public void RunSystem()

{

int k = 0, num\_feat = -1;

int[] arr\_otv = new int[2] { 0, 0 };

features = dbHandler.Write(DBHelper.TABLE\_FEATURES, DBHelper.FIELD\_FEATURES\_FEATURE);

Random rand = new Random();

while (k != 2 && this.features.Count > 0)

{

num\_feat = rand.Next(this.features.Count);

Invoke(changeFeatureDelegate, this.features[num\_feat].getFeature(), false);

mre.WaitOne();

if (is\_yes) {

arr\_otv[k] = features[num\_feat].getId();

k++;

}

features.RemoveAt(num\_feat);

mre = new ManualResetEvent(false);

}

if (k == 2)

{

string \_object = dbHandler.CheckObject(arr\_otv);

if (\_object != null) {

MessageBox.Show("Объект распознан: " + \_object, "Ответ");

} else {

AddObject(k, arr\_otv);

}

} else {

AddObject(k, arr\_otv);

}

Invoke(changeFeatureDelegate, "НАЖМИТЕ 'СТАРТ' ДЛЯ НАЧАЛА!", true);

thread.Abort();

}

void AddObject(int k, int[] arr\_otv)

{

MessageBox.Show("Объект не распознан, добавьте объект идентификации...", "Ответ");

Input inputForm = new Input(dbHandler, k);

inputForm.ShowDialog();

if (inputForm.getSuccess())

{

dbHandler.Insert(new string[] { inputForm.getFeature1(), inputForm.getFeature2() },

inputForm.getObject(), arr\_otv);

}

}

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

{

this.is\_yes = true;

mre.Set();

}

private void no\_Click(object sender, EventArgs e)

{

this.is\_yes = false;

mre.Set();

}

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

{

if (MessageBox.Show("Вы точно хотите выйти?", "Выход", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

thread.Abort();

Application.Exit();

}

}

}

}

**Класс добавления нового объекта и характеристик:**

using System;

using System.Windows.Forms;

using System.Data;

namespace ii\_lab\_3

{

public partial class Input : Form

{

DBHandler dbHandler;

int k;

string feature1;

string feature2;

string \_object;

bool success = false;

public Input(DBHandler dbHandler, int k)

{

InitializeComponent();

this.dbHandler = dbHandler;

if (k >= 1)

{

this.k = k;

textBox1.Enabled = false;

label1.Enabled = false;

if (k >= 2)

{

textBox2.Enabled = false;

label2.Enabled = false;

}

}

}

public string getFeature1() { return this.feature1; }

public string getFeature2() { return this.feature2; }

public string getObject() { return this.\_object; }

public bool getSuccess() { return this.success; }

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

{

this.Close();

}

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

{

if (String.Empty != textBoxObject.Text)

{

DataTable dataTable = dbHandler.GetDataTable(DBHelper.TABLE\_OBJECTS);

string object\_filter = DBHelper.FIELD\_OBJECTS\_OBJECT + " = '" + textBoxObject.Text + "'";

DataRow[] dataRows = dataTable.Select(object\_filter);

if (dataRows.Length > 0) {

MessageBox.Show("Такой объект уже существует в базе данных...", "Ошибка");

this.Close();

return;

} else {

this.\_object = textBoxObject.Text;

}

if (k <= 1) {

if (String.Empty != textBox2.Text) {

this.feature2 = textBox2.Text;

} else {

MessageBox.Show("Поле 2 заполнено...", "Ошибка");

this.Close();

return;

}

if (k == 0) {

if (String.Empty != textBox1.Text) {

this.feature1 = textBox1.Text;

} else {

MessageBox.Show("Поле 1 не заполнено...", "Ошибка");

this.Close();

return;

}

}

}

}

this.success = true;

this.Close();

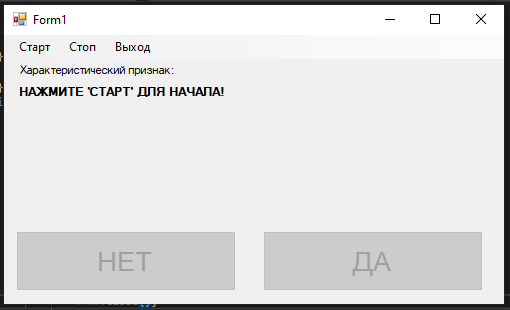
}

}

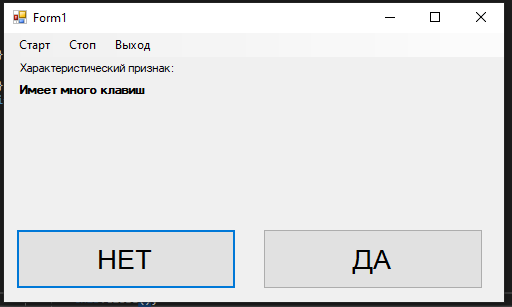
}

ПРИМЕР РАБОТЫ ПРОГРАММЫ:

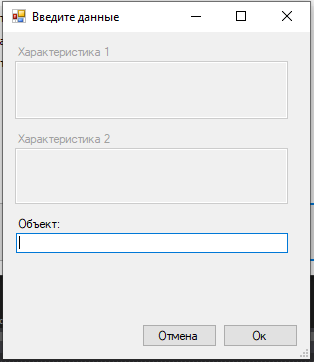
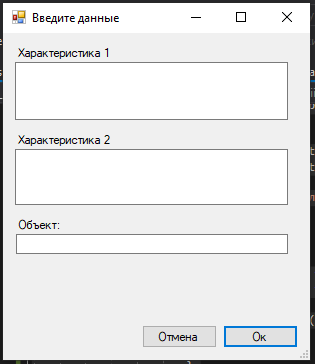
Главная форма:



Процесс ответа на вопросы:



Ввод нового объекта, в случае, когда k = 2 и k = 0:

Объект, подошедший под описание:

