Министерство образования и науки Российской Федерации

Федеральное Государственное Бюджетное Образовательное Учреждение

Высшего Профессионального Образования

Тамбовский Государственный Технический Университет

Кафедра

Лабораторная работа №1

дисциплина «Базы знаний»

Вариант 4

Выполнил студент гр. –41 qwinmen

Проверила: И. Л.

Тамбов

**Задание:**

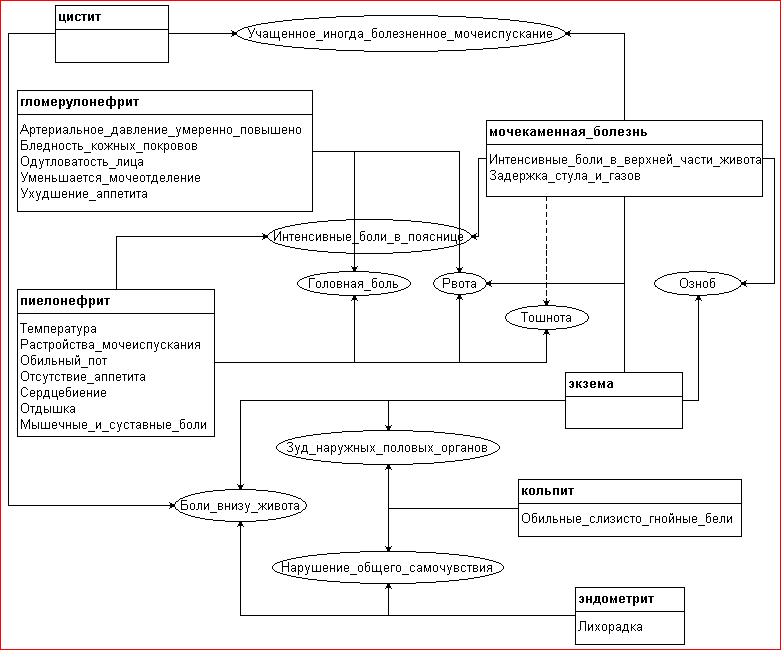
Разработать экспертную систему, реализующую алгоритм обратной цепочки рассуждений на основе разработанной базы знаний. Предусмотреть вывод на экран системы объяснений.

Рисунок Схема совпавших симптомов

**Решение:**

На рисунке 1 приведена схема Варианта 4, в которой у каждой болезни мочеполовой системы перечисляются свойственные только ей симптомы. Отдельно вынесены только те признаки, проявление коих может проявлятся в нескольких заболеваниях одновременно.

Правила, собирающие все симптомы воедино, следующие:

**ЕСЛИ** Артериальное\_давление\_умеренно\_повышено = да И Бледность\_кожных\_покровов = да И Головная\_боль = да И Одутловатость\_лица = да И Рвота = да И Уменьшается\_мочеотделение = да И Ухудшение\_аппетита = да ТО болезнь = гломерулонефрит

**ЕСЛИ** Обильные\_слизисто-гнойные\_бели = да И Зуд\_наружных\_половых\_органов = да И Нарушение\_общего\_самочувствия = да ТО болезнь = кольпит

**ЕСЛИ** Интенсивные\_боли\_в\_пояснице = да И Интенсивные\_боли\_в\_верхней\_части\_живота = да И Тошнота = да И Задержка\_стула\_и\_газов = да И Рвота = да И Учащенное,иногда\_болезненное\_мочеиспускание = да ТО болезнь = мочекаменная\_болезнь

**ЕСЛИ** Температура = да И Растройства\_мочеиспускания = да И Интенсивные\_боли\_в\_пояснице = да И Озноб = да И Обильный\_пот = да И Головная\_боль = да И Тошнота = да И Рвота = да И Отсутствие\_аппетита = да И Сурдцебиение = да И Отдышка = да И Мышечные\_и\_суставные\_боли = да ТО болезнь = пиелонефрит

**ЕСЛИ** Боли\_внизу\_живота = да И Учащенное,иногда\_болезненное\_мочеиспускание = да ТО болезнь = цистит

**ЕСЛИ** Нарушение\_общего\_самочувствия = да И Боли\_внизу\_живота = да И Лихорадка = да ТО болезнь = эндометрит

Работа обратной цепочки сводится к следующему алгоритму:

Изначально имеем связки «болезнь, симптомы» (в программе это список из «key, value» пар), каждый симптом может быть как подтвержденным – true, так и опровергнутым – false (по умолчанию). Проявление набора симптомов у пациента формирует определенный перечень, его мы выставляем в «активное» состояние. И если не сразу, то чере какое-то время элементы списка совпадут с эталон-набором из базы, что укажет на название болезни, либо, на её отсутствие, однако данный результат заведомо провальный и решается правками базы (или вынесением вероятностного вердикта по количеству совпадений).

Рисунок Пример связи

Заключение: алгоритм малоэффективен.

Код:

//Вариант 4

namespace Lekari

{

class Program

{

private static Bolesni \_clBolesni;

static void Main(string[] args)

{

{//Note: Формируем базу знаний из известных данной болезни симптомов:

ЕСЛИ(clBaseKnow.SimptomMPS.Артериальное\_давление\_умеренно\_повышено,

clBaseKnow.SimptomMPS.Бледность\_кожных\_покровов, clBaseKnow.SimptomMPS.Головная\_боль,

clBaseKnow.SimptomMPS.Одутловатость\_лица, clBaseKnow.SimptomMPS.Рвота,

clBaseKnow.SimptomMPS.Уменьшается\_мочеотделение, clBaseKnow.SimptomMPS.Ухудшение\_аппетита);

ЕСЛИ(clBaseKnow.SimptomMPS.Обильные\_слизисто\_гнойные\_бели,

clBaseKnow.SimptomMPS.Зуд\_наружных\_половых\_органов, clBaseKnow.SimptomMPS.Нарушение\_общего\_самочувствия);

ЕСЛИ(clBaseKnow.SimptomMPS.Интенсивные\_боли\_в\_пояснице,

clBaseKnow.SimptomMPS.Интенсивные\_боли\_в\_верхней\_части\_живота, clBaseKnow.SimptomMPS.Тошнота,

clBaseKnow.SimptomMPS.Задержка\_стула\_и\_газов, clBaseKnow.SimptomMPS.Рвота,

clBaseKnow.SimptomMPS.Учащенное\_иногда\_болезненное\_мочеиспускание);

ЕСЛИ(clBaseKnow.SimptomMPS.Температура, clBaseKnow.SimptomMPS.Растройства\_мочеиспускания,

clBaseKnow.SimptomMPS.Интенсивные\_боли\_в\_пояснице, clBaseKnow.SimptomMPS.Озноб,

clBaseKnow.SimptomMPS.Обильный\_пот, clBaseKnow.SimptomMPS.Головная\_боль, clBaseKnow.SimptomMPS.Тошнота,

clBaseKnow.SimptomMPS.Рвота, clBaseKnow.SimptomMPS.Отсутствие\_аппетита,

clBaseKnow.SimptomMPS.Сурдцебиение, clBaseKnow.SimptomMPS.Отдышка,

clBaseKnow.SimptomMPS.Мышечные\_и\_суставные\_боли);

ЕСЛИ(clBaseKnow.SimptomMPS.Боли\_внизу\_живота,

clBaseKnow.SimptomMPS.Учащенное\_иногда\_болезненное\_мочеиспускание);

ЕСЛИ(clBaseKnow.SimptomMPS.Нарушение\_общего\_самочувствия, clBaseKnow.SimptomMPS.Боли\_внизу\_живота,

clBaseKnow.SimptomMPS.Лихорадка);

ЕСЛИ(clBaseKnow.SimptomMPS.Озноб, clBaseKnow.SimptomMPS.Боли\_внизу\_живота,

clBaseKnow.SimptomMPS.Рвота, clBaseKnow.SimptomMPS.Зуд\_наружных\_половых\_органов);

}

{

for (var i = clBaseKnow.Bolesni.start + 1; i < clBaseKnow.Bolesni.end; i++)

{

var outres = 0;

if (int.TryParse(i.ToString(), out outres)) continue;

var bolesn = i.ToString();//дергаем болезнь из списка

var d = \_clBolesni.GetSimptomesByName((clBaseKnow.Bolesni)Enum.Parse(typeof(clBaseKnow.Bolesni), bolesn));

var activStatesFl = new List<bool>();

foreach (var mps in d)//вывод симптомов для болезни

{

var numBlsn = "";

if (!\_clBolesni.isActivateSimptom(mps.Key))

{

Console.Write("Симптом " + mps + " есть? y/n \r\n");

numBlsn = Console.ReadLine();

}

else numBlsn = "y";

activStatesFl.Add(numBlsn == "y" ? true : false);

}

Console.WriteLine();

//выставить симптомы

var res = \_clBolesni.SetActivSimptom((clBaseKnow.Bolesni)Enum.Parse(typeof(clBaseKnow.Bolesni), bolesn), activStatesFl);

if (res)

{

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("Болезнь {0} подтверждена!", (clBaseKnow.Bolesni)Enum.Parse(typeof(clBaseKnow.Bolesni), bolesn));

foreach (var keyVal in \_clBolesni.GetSimptomesByName(

(clBaseKnow.Bolesni) Enum.Parse(typeof (clBaseKnow.Bolesni), bolesn)))

Console.WriteLine("Симптомы {0}", keyVal);

break;

}

}

}

Console.ReadKey();

}

private static clBaseKnow.Bolesni ЕСЛИ(params clBaseKnow.SimptomMPS []obj)

{

var sum = obj.Sum(simptomMps => (int) simptomMps);

for (var i = clBaseKnow.Bolesni.start; i <= clBaseKnow.Bolesni.end; i++)

if ((int)i == sum)

{

//заполняем базу связкой Болезнь-Симптомы

\_clBolesni = new Bolesni(i, obj);

return i;

}

return clBaseKnow.Bolesni.start;

}

}

}

namespace Lekari

{

class clBaseKnow

{

/// <summary>

/// Симптомы мочеполовой системы

/// </summary>

public enum SimptomMPS : int

{//---------------1=MPS--------------------------

Артериальное\_давление\_умеренно\_повышено = 1,

Бледность\_кожных\_покровов = 2,

Боли\_внизу\_живота = 3,

Головная\_боль = 4,

Задержка\_стула\_и\_газов = 5,

Зуд\_наружных\_половых\_органов = 6,

Интенсивные\_боли\_в\_верхней\_части\_живота = 7,

Интенсивные\_боли\_в\_пояснице = 8,

Лихорадка = 9,

Мышечные\_и\_суставные\_боли = 10,

Нарушение\_общего\_самочувствия = 11,

Обильный\_пот = 12,

Обильные\_слизисто\_гнойные\_бели = 13,

Одутловатость\_лица = 14,

Озноб = 15,

Отдышка = 16,

Отсутствие\_аппетита = 17,

Растройства\_мочеиспускания = 18,

Рвота = 19,

Сурдцебиение = 20,

Температура = 21,

Тошнота = 22,

Уменьшается\_мочеотделение = 23,

Ухудшение\_аппетита = 24,

Учащенное\_иногда\_болезненное\_мочеиспускание = 25,

//---------------2=--------------------------

}

/// <summary>

/// Болезнь = раздел\_болезни

/// </summary>

public enum Bolesni : int

{

start = 0,

end = 200,

//----------1=MPS-----------------------

экзема = 43,

эндометрит = 23,

цистит = 28,

кольпит = 30,

мочекаменная\_болезнь = 86,

гломерулонефрит = 87,

пиелонефрит = 182,

//----------2=-----------------------

}

}

internal class Bolesni

{

private clBaseKnow.Bolesni \_name;

private Dictionary<clBaseKnow.SimptomMPS, bool > \_simptoms = new Dictionary<clBaseKnow.SimptomMPS, bool>();

private static Dictionary<clBaseKnow.Bolesni, Dictionary<clBaseKnow.SimptomMPS, bool>> \_keyValueArray =

new Dictionary<clBaseKnow.Bolesni, Dictionary<clBaseKnow.SimptomMPS, bool>>();

/// <summary>

/// Внести болезнь и симптомы в картотеку

/// </summary>

/// <param name="name"></param>

/// <param name="obj"></param>

public Bolesni(clBaseKnow.Bolesni name, params clBaseKnow.SimptomMPS []obj)

{

\_name = name;

foreach (var simptomMps in obj)

\_simptoms.Add(simptomMps, false);

\_keyValueArray.Add(\_name, \_simptoms);

}

/// <summary>

/// Вернуть список симптомов по названию болезни

/// </summary>

/// <param name="name"></param>

/// <returns></returns>

public Dictionary<clBaseKnow.SimptomMPS, bool> GetSimptomesByName(clBaseKnow.Bolesni name)

{

try

{

return \_keyValueArray[name];

}

catch (KeyNotFoundException e)

{

throw new Exception("Болезни "+name+" нет в базе!");

return null;

}

}

/// <summary>

/// По имени запросить симптомы и изменить статус активности

/// </summary>

/// <param name="name"></param>

/// <param name="flagActiv"></param>

/// <returns>Вернет true если все симптомы в true</returns>

public bool SetActivSimptom(clBaseKnow.Bolesni name, List<bool> flagActiv)

{

var state = true;

var resTmp = \_keyValueArray[name];

var dic = new Dictionary<clBaseKnow.SimptomMPS, bool>();

var i = 0;

foreach (KeyValuePair<clBaseKnow.SimptomMPS, bool> keyValuePair in resTmp)

{

dic.Add(keyValuePair.Key, flagActiv[i]);

if (!flagActiv[i])

state = false;

//иначе флаг Активен и поместить в стек

else StackActivSimptoms(keyValuePair.Key);

i++;

}

\_keyValueArray[name] = dic;

return state;

}

private static Stack<clBaseKnow.SimptomMPS> \_stackSimptomsActiv = new Stack<clBaseKnow.SimptomMPS>();

/// <summary>

/// Положить активный симпт в стек

/// </summary>

/// <param name="simptomMps"></param>

private void StackActivSimptoms(clBaseKnow.SimptomMPS simptomMps)

{

if (!\_stackSimptomsActiv.Contains(simptomMps))

\_stackSimptomsActiv.Push(simptomMps);

}

/// <summary>

/// Глянуть в стек по симптому, если там такой есть, то вернуть TRUE

/// </summary>

/// <param name="simptomMps"></param>

/// <returns>Вернет Да, если там такой есть</returns>

public bool isActivateSimptom(clBaseKnow.SimptomMPS simptomMps)

{

return \_stackSimptomsActiv.Contains(simptomMps);

}

}

}

Пример

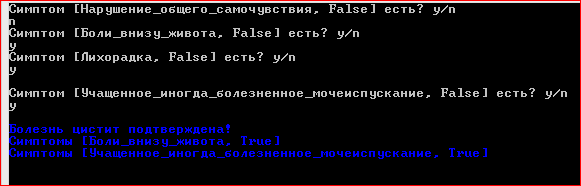


Рисунок Положительный результат