Министерство науки и образования Российской Федерации

ФГБОУ ВО «Пермский Национальный Исследовательский Политехнический Университет»

Факультет прикладной математики и механики

Лабораторные работы   
по дисциплине

Интеллектуальные информационные системы и технологии

Выполнил: студент группы ИСТ-17-1бз

Лодейщикова Яна Ильдусовна

Бакланова Алёна Михайловна

Проверил: старший преподаватель

Истомин Денис Андреевич

Пермь, 2023

**Оглавление**

[Лабораторная работа №1 3](#_Toc123561110)

[Постановка задачи 3](#_Toc123561111)

[Предметная область и база знаний 3](#_Toc123561112)

[Скрипт для системы CLIPS 7](#_Toc123561113)

[Лабораторная работа №2 14](#_Toc123561114)

[Постановка задачи 14](#_Toc123561115)

[Структура программы 14](#_Toc123561116)

[Код программы 15](#_Toc123561117)

[Файл Types.cs 16](#_Toc123561118)

[Файл MemoryComponent.cs 18](#_Toc123561119)

[Файл InferenceComponent.cs 19](#_Toc123561120)

[Файл ExplanationComponent.cs 21](#_Toc123561121)

[Файл основной программы Program.cs 22](#_Toc123561122)

[Файл ConditionConverter.cs 23](#_Toc123561123)

[База знаний 25](#_Toc123561124)

[Лабораторная работа №3 38](#_Toc123561125)

[Постановка задачи 38](#_Toc123561126)

[Вопросы к системе 38](#_Toc123561127)

[База знаний 39](#_Toc123561128)

# Лабораторная работа №1

## Постановка задачи

1. Выбрать предметную область, в рамках которой планируется построить БЗ. Должна решаться задача, например, диагностики или прогнозирования.
2. Визуально оформить граф И/ИЛИ.
3. Описать правила БЗ в текстовом виде в нотации IF-THEN.
4. Ознакомиться с тестовыми примерами, которые идут совместно с поставкой среды CLIPS.
5. В инструменте разработки ЭС CLIPS реализовать сформированные правила.

## Предметная область и база знаний

Тема, выбранная для экспертной системы – выбор наушников. Визуально оформленный граф И/ИЛИ правил для системы изображен на Рис. 1.

Правила в нотации IF-THEN:

1. Правило: Вид - Затычки

IF «Устройство» = «Телефон» THEN «Вид» = «Затычки»

1. Правило: Вид - Охватывающие

IF «Устройство» = «Компьютер» OR «Устройство» = «ТВ» THEN «Вид» = «Охватывающие»

1. Правило: Микрофон - Да

IF «Назначение» = «Звонки» OR «Назначение» = «Игры» THEN «Микрофон» = «Да»

1. Правило: Микрофон - Нет

IF «Назначение» = «Видео» OR «Назначение» = «Музыка» THEN «Микрофон» = «Нет»

1. Правило: Беспроводные - Да

IF «Устройство» = «ТВ» OR «Устройство» = «Телефон» AND («Назначение» = «Звонки» OR «Назначение» = «Музыка») OR «Устройство» = «Компьютер» AND («Назначение» = «Игры» OR «Назначение» = «Видео» OR «Назначение» = «Музыка») THEN «Беспроводные» = «Да»

1. Правило: Беспроводные - Нет

IF «Устройство» = «Телефон» AND («Назначение» = «Игры» OR «Назначение» = «Видео») OR «Устройство» = «Компьютер» AND «Назначение» = «Звонки» THEN «Беспроводные» = «Нет»

1. Правило: CADENA YH-12

IF «Вид» = «Затычки» AND «Беспроводные» = «Нет» AND «Микрофон» = «Да» AND «Бюджет» = «1000» THEN «Результат» = «CADENA YH-12»

1. Правило: JVC HA-FX38

IF «Вид» = «Затычки» AND «Беспроводные» = «Нет» AND «Микрофон» = «Нет» AND «Бюджет» = «1000» THEN «Результат» = «JVC HA-FX38»

1. Правило: QCY T1c

IF «Вид» = «Затычки» AND «Беспроводные» = «Да» AND «Бюджет» = «1000» THEN «Результат» = «QCY T1c»

1. Правило: Sennheiser CX 300S

IF «Вид» = «Затычки» AND «Беспроводные» = «Нет» AND «Бюджет» = «5000» THEN «Результат» = «Sennheiser CX 300S»

1. Правило: JBL Wave 200TWS

IF «Вид» = «Затычки» AND «Беспроводные» = «Да» AND «Бюджет» = «5000» THEN «Результат» = «JBL Wave 200TWS»

1. Правило: Samsung Buds 2

IF «Вид» = «Затычки» AND «Бюджет» = «10000» THEN «Результат» = « Samsung Buds 2»

1. Правило: A4Tech HS-200

IF «Вид» = «Охватывающие» AND «Беспроводные» = «Нет» AND «Бюджет» = «1000» THEN «Результат» = «A4Tech HS-200»

1. Правило: Harper HB-210

IF «Вид» = «Охватывающие» AND «Беспроводные» = «Да» AND «Бюджет» = «1000» THEN «Результат» = «Harper HB-210»

1. Правило: Soundcore Life 2 Neo Q10i

IF «Вид» = «Охватывающие» AND «Беспроводные» = «Да» AND «Микрофон» = «Нет» AND «Бюджет» = «5000» THEN «Результат» = «Soundcore Life 2 Neo Q10i»

1. Правило: SVEN AP-B900MV

IF «Вид» = «Охватывающие» AND «Беспроводные» = «Да» AND «Микрофон» = «Да» AND «Бюджет» = «5000» THEN «Результат» = «SVEN AP-B900MV»

1. Правило: SteelSeries Arctis 1

IF «Вид» = «Охватывающие» AND «Беспроводные» = «Нет» AND «Бюджет» = «5000» THEN «Результат» = «SteelSeries Arctis 1»

1. Правило: HyperX Cloud Alpha S HX-HSCAS

IF «Вид» = «Охватывающие» AND «Беспроводные» = «Нет» AND «Микрофон» = «Да» AND «Бюджет» = «10000» THEN «Результат» = «HyperX Cloud Alpha S HX-HSCAS»

1. Правило: JBL Tune 710BT

IF «Вид» = «Охватывающие» AND «Беспроводные» = «Да» AND «Микрофон» = «Да» AND «Бюджет» = «10000» THEN «Результат» = «JBL Tune 710BT»

1. Правило: Fostex T20RP MK3

IF «Вид» = «Охватывающие» AND «Беспроводные» = «Нет» AND «Микрофон» = «Нет» AND «Бюджет» = «10000» THEN «Результат» = «Fostex T20RP MK3»

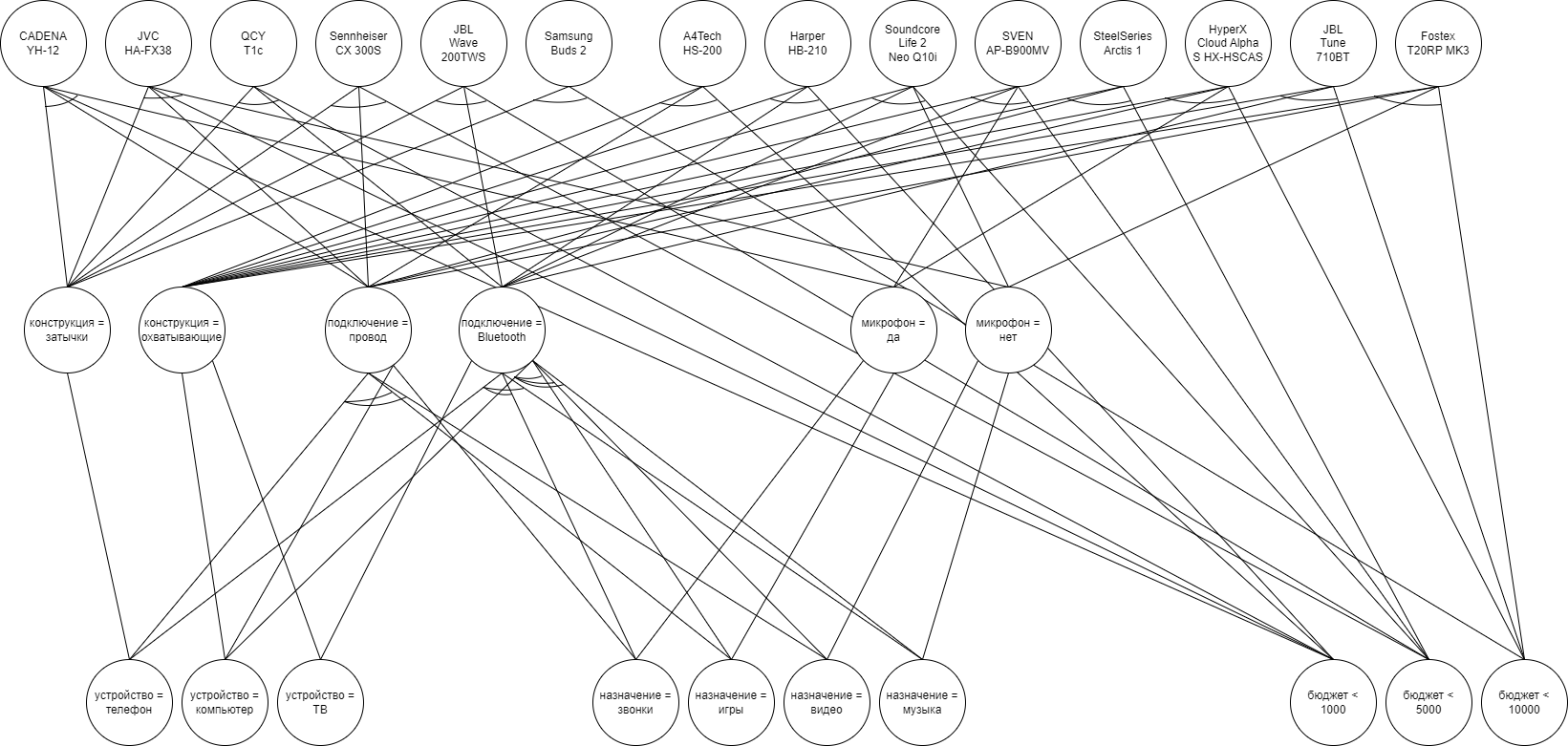


Рис. И/ИЛИ-граф продукций

## Скрипт для системы CLIPS

;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;;\* DEFFUNCTIONS \*

;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(deffunction ask-question (?question $?allowed-values)

  (printout t ?question)

  (bind ?answer (read))

  (if (lexemep ?answer)

    then (bind ?answer (lowcase ?answer))

  )

  (while (not (member$ ?answer ?allowed-values)) do

    (printout t ?question)

    (bind ?answer (read))

    (if (lexemep ?answer)

      then (bind ?answer (lowcase ?answer)))

  )

  ?answer

)

(deffunction ask-question-yes-no (?question)

  (bind ?response (ask-question ?question yes no y n))

  (if (or (eq ?response yes) (eq ?response y))

    then yes

    else no

  )

)

;;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;;;\* QUERY RULES \*

;;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(defrule determine-device ""

  (not (device ?))

  (not (result ?))

  =>

  (assert (device (ask-question

    "К какому устройству вы будете подключать наушники (phone/pc/tv)? "

    phone pc tv

  )))

)

(defrule determine-purpose ""

  (not (purpose ?))

  (not (result ?))

  =>

  (assert (purpose (ask-question

    "Для чего чаще всего будут использоваться наушники (calls/games/video/music)? "

    calls games video music

  )))

)

(defrule determine-budget ""

  (not (budget ?))

  (not (result ?))

  =>

  (assert (budget (ask-question

    "Какой у вас бюджет (1000/5000/10000)?"

    1000 5000 10000

  )))

)

;;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;;;\* INTERMEDIATE RULES \*

;;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;Определение подходящей конструкции, затычки или охватывающие

(defrule determine-form-in-ear ""

  (device phone)

  (not (form ?))

  (not (result ?))

  =>

  (assert (form in-ear))

)

(defrule determine-form-over-ear ""

  (or (device pc) (device tv))

  (not (form ?))

  (not (result ?))

  =>

  (assert (form over-ear))

)

;Определение способа подключения, проводные или беспроводные

(defrule determine-wireless-yes ""

  (or

    (device tv)

    (and (device phone) (or (purpose calls) (purpose music)))

    (and (device pc) (or (purpose games) (purpose video) (purpose music)))

  )

  (not (wireless ?))

  (not (result ?))

  =>

  (assert (wireless yes))

)

(defrule determine-wireless-no ""

  (or

    (and (device phone) (or (purpose games) (purpose video)))

    (and (device pc) (purpose calls))

  )

  (not (wireless ?))

  (not (result ?))

  =>

  (assert (wireless no))

)

;Определение, нужен микрофон или нет

(defrule determine-microphone-yes ""

  (or (purpose calls) (purpose games))

  (not (microphone ?))

  (not (result ?))

  =>

  (assert (microphone yes))

)

(defrule determine-microphone-no ""

  (or (purpose video) (purpose music))

  (not (microphone ?))

  (not (result ?))

  =>

  (assert (microphone no))

)

;;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;;;\* RESULT RULES \*

;;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(defrule no-result ""

  (declare (salience -10))

  (not (result ?))

  =>

  (assert (result "Невозможно найти подходящий вариант"))

)

(defrule result-cadena ""

  (form in-ear)

  (wireless no)

  (microphone yes)

  (budget 1000)

  (not (result ?))

  =>

  (assert (result "CADENA YH-12"))

)

(defrule result-jvc ""

  (form in-ear)

  (wireless no)

  (microphone no)

  (budget 1000)

  (not (result ?))

  =>

  (assert (result "JVC HA-FX38"))

)

(defrule result-qcy ""

  (form in-ear)

  (wireless yes)

  (budget 1000)

  (not (result ?))

  =>

  (assert (result "QCY T1c"))

)

(defrule result-sennheiser ""

  (form in-ear)

  (wireless no)

  (budget 5000)

  (not (result ?))

  =>

  (assert (result "Sennheiser CX 300S"))

)

(defrule result-jbl-1 ""

  (form in-ear)

  (wireless yes)

  (budget 5000)

  (not (result ?))

  =>

  (assert (result "JBL Wave 200TWS"))

)

(defrule result-samsung ""

  (form in-ear)

  (budget 10000)

  (not (result ?))

  =>

  (assert (result "Samsung Buds 2"))

)

(defrule result-a4tech ""

  (form over-ear)

  (wireless no)

  (budget 1000)

  (not (result ?))

  =>

  (assert (result "A4Tech HS-200"))

)

(defrule result-harper ""

  (form over-ear)

  (wireless yes)

  (budget 1000)

  (not (result ?))

  =>

  (assert (result "Harper HB-210"))

)

(defrule result-soundcore ""

  (form over-ear)

  (wireless yes)

  (microphone no)

  (budget 5000)

  (not (result ?))

  =>

  (assert (result "Soundcore Life 2 Neo Q10i"))

)

(defrule result-sven ""

  (form over-ear)

  (wireless yes)

  (microphone yes)

  (budget 5000)

  (not (result ?))

  =>

  (assert (result "SVEN AP-B900MV"))

)

(defrule result-steelseries ""

  (form over-ear)

  (wireless no)

  (budget 5000)

  (not (result ?))

  =>

  (assert (result "SteelSeries Arctis 1"))

)

(defrule result-hyperx ""

  (form over-ear)

  (wireless no)

  (microphone yes)

  (budget 10000)

  (not (result ?))

  =>

  (assert (result "HyperX Cloud Alpha S HX-HSCAS"))

)

(defrule result-jbl-2 ""

  (form over-ear)

  (wireless yes)

  (microphone yes)

  (budget 10000)

  (not (result ?))

  =>

  (assert (result "JBL Tune 710BT"))

)

(defrule result-fostex ""

  (form over-ear)

  (wireless no)

  (microphone no)

  (budget 10000)

  (not (result ?))

  =>

  (assert (result "Fostex T20RP MK3"))

)

;;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;;;\* STARTUP AND CONCLUSION RULES \*

;;;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(defrule system-banner ""

  (declare (salience 10))

  =>

  (printout t crlf crlf)

  (printout t "Система рекомендации покупки наушников")

  (printout t crlf crlf)

)

(defrule print-result ""

  (declare (salience 10))

  (result ?item)

  =>

  (printout t crlf crlf)

  (printout t "Предлагаемая модель:")

  (printout t crlf crlf)

  (format t " %s%n%n%n" ?item)

)

Пример работы системы:

Система рекомендации покупки наушников

Какой у вас бюджет (1000/5000/10000)? 5000

Для чего чаще всего будут использоваться наушники (calls/games/video/music)? calls

К какому устройству вы будете подключать наушники (phone/pc/tv)? phone

Предлагаемая модель:

JBL Wave 200TWS

# Лабораторная работа №2

## Постановка задачи

1. Взять предметную область и БЗ из лабораторной работы №1.
2. В текстовом виде описать БЗ в выбранной нотации (JSON).
3. Спроектировать интерфейсы модулей реализуемой оболочки ЭС
4. Начать разработку в системе хранения контроля версий.

## Структура программы

Экспертная система состоит из трех основных компонентов – Рабочая память (MemoryComponent), Механизм логического вывода (InferenceComponent), Компонента объяснения (ExplanationComponent).

Память содержит список действующих фактов, список правил базы знаний, а также отдельно списки активированных правил и правил, готовых к активации (условия которых выполнены).

Компонента вывода имеет один основной метод – вывод результата. Алгоритм вывода выбирает одно из правил, готовых к активации (условия которых выполнены), добавляет правую часть этого правила к списку действующих фактов (при необходимости – задает вопрос пользователю), если один из добавленных фактов является результатом – завершает работу.

Компонента объяснения определяет, какие факты и правила привели к полученному результату.

Также используется вспомогательный JSON-конвертер ConditionConverter для правильного чтения подтипов Condition.

## Код программы

Пример работы программы:

Какой у вас бюджет?

1 - 1000

2 - 5000

3 - 10000

1

Для чего чаще всего будут использоваться наушники?

1 - Звонки

2 - Игры

3 - Видео

4 - Музыка

1

К какому устройству вы будете подключать наушники?

1 - Телефон

2 - Компьютер

3 - ТВ

2

Результат:

A4Tech HS-200

Факт: 'A4Tech HS-200'

Добавлен в результате срабатывания правила: (('Вид = Охватывающие' И 'Беспроводные = Нет' И 'Бюджет = 1000') ==> 'A4Tech HS-200')

В связи с наличием фактов:

'Вид' = Охватывающие

'Беспроводные' = Нет

'Бюджет' = 1000 (Вопрос: 'Какой у вас бюджет?' (1000/5000/10000))

Факт: 'Бюджет' = 1000 (Вопрос: 'Какой у вас бюджет?' (1000/5000/10000))

Добавлен в результате срабатывания правила: (Не определено 'Бюджет' ==> 'Бюджет' = 1000 (Вопрос: 'Какой у вас бюджет?' (1000/5000/10000)))

Факт: 'Беспроводные' = Нет

Добавлен в результате срабатывания правила: ((Не определено 'Беспроводные' И (('Устройство = Телефон' И ('Назначение = Игры' ИЛИ 'Назначение = Видео')) ИЛИ ('Устройство = Компьютер' И 'Назначение = Звонки'))) ==> 'Беспроводные' = Нет)

В связи с наличием фактов:

'Устройство' = Компьютер (Вопрос: 'К какому устройству вы будете подключать наушники?' (Телефон/Компьютер/ТВ))

'Назначение' = Звонки (Вопрос: 'Для чего чаще всего будут использоваться наушники?' (Звонки/Игры/Видео/Музыка))

Факт: 'Назначение' = Звонки (Вопрос: 'Для чего чаще всего будут использоваться наушники?' (Звонки/Игры/Видео/Музыка))

Добавлен в результате срабатывания правила: (Не определено 'Назначение' ==> 'Назначение' = Звонки (Вопрос: 'Для чего чаще всего будут использоваться наушники?' (Звонки/Игры/Видео/Музыка)))

Факт: 'Устройство' = Компьютер (Вопрос: 'К какому устройству вы будете подключать наушники?' (Телефон/Компьютер/ТВ))

Добавлен в результате срабатывания правила: (Не определено 'Устройство' ==> 'Устройство' = Компьютер (Вопрос: 'К какому устройству вы будете подключать наушники?' (Телефон/Компьютер/ТВ)))

Факт: 'Вид' = Охватывающие

Добавлен в результате срабатывания правила: ((Не определено 'Вид' И ('Устройство = Компьютер' ИЛИ 'Устройство = ТВ')) ==> 'Вид' = Охватывающие)

В связи с наличием фактов:

'Устройство' = Компьютер (Вопрос: 'К какому устройству вы будете подключать наушники?' (Телефон/Компьютер/ТВ))

Факт: 'Устройство' = Компьютер (Вопрос: 'К какому устройству вы будете подключать наушники?' (Телефон/Компьютер/ТВ))

Добавлен в результате срабатывания правила: (Не определено 'Устройство' ==> 'Устройство' = Компьютер (Вопрос: 'К какому устройству вы будете подключать наушники?' (Телефон/Компьютер/ТВ)))

### Файл Types.cs

using System;

using System.Collections.Generic;

using System.Linq;

namespace ExpertSystem;

// Конъюнкция и дизъюнкция

enum ExpressionType

{

And, Or,

}

// Правило состоит из условия, при котором правило активируется (левая часть правила),

// и факта, который добавляется в базу после активации правила (правая часть)

struct Rule

{

public Condition Condition;

public Fact Fact;

public override string ToString() => $"({Condition} ==> {Fact})";

}

// Условие может быть либо простым условием сравнения факта со значением,

// либо конъюнкцией/дизъюнкцией набора других условий

abstract class Condition

{

// Выполняется ли условие при наличии данного набора фактов

// В том числе возвращает список конкретных фактов, которых достаточно для выполнения условия

public abstract (bool CanActivate, List<Fact> Requirements) Evaluate(IEnumerable<Fact> facts);

}

// Конъюнкция или дизъюнкция набора условий

class ConditionExpression : Condition

{

public string Operator;

public Condition[] Conditions;

public ExpressionType Type =>

Operator == "AND" ? ExpressionType.And : ExpressionType.Or;

public override (bool CanActivate, List<Fact> Requirements) Evaluate(IEnumerable<Fact> facts)

{

var results = Conditions.Select(condition => condition.Evaluate(facts));

if (Type == ExpressionType.And)

{

var requirements = results

.SelectMany(result => result.Requirements)

.ToList();

return (results.All(result => result.CanActivate), requirements);

}

foreach (var result in results)

{

if (result.CanActivate) return result;

}

return (false, new());

}

public override string ToString() => $"({String.Join<Condition>(Type == ExpressionType.And ? " И " : " ИЛИ ", Conditions)})";

}

// Проверка, определен или не определен факт с заданным именем, или сравнение факта с заданным значением

class ValueCondition : Condition

{

public string FactName;

public string? Value;

public override (bool CanActivate, List<Fact> Requirements) Evaluate(IEnumerable<Fact> facts)

{

var foundFact = facts.FirstOrDefault(fact => fact.Name == FactName);

// Условие выполняется, если факт с заданным именем не найден и условие - "Факт не определен",

// либо значение факта равно значению условия

return foundFact is null

? (Value is null, new List<Fact>() )

: (Value == foundFact.Value, new List<Fact> { foundFact } );

}

public override string ToString() =>

Value is null

? $"Не определено '{FactName}'"

: $"'{FactName} = {Value}'";

}

// Факт может иметь определенное значение, его значение может определяться вопросом к пользователю,

// либо (при отсутствии значения и вопроса) факт является искомым результатом

class Fact

{

public string Name;

public string? Value;

public Input? Input;

public bool IsResult => Value == null && Input == null;

public override string ToString()

{

string value = $"'{Name}'";

if (Value != null) value += $" = {Value}";

if (Input != null) value += $" (Вопрос: {Input})";

return value;

}

}

// Вопрос к пользователю с выбором одного из предложенных значений

struct Input

{

public string Question;

public string[] Values;

public override string ToString()

{

string value = $"'{Question}'";

if (Values != null) value += $" ({string.Join("/", Values)})";

return value;

}

}

### Файл MemoryComponent.cs

using System.Collections.Generic;

using System.Linq;

namespace ExpertSystem;

class MemoryComponent

{

// Список действующих фактов. Каждому факту в соответствие поставлено правило,

// в результате активации которого получен данный факт.

public readonly Dictionary<Fact, Rule> Facts;

// Все правила базы знаний

public readonly Rule[] Rules;

// Список правил, условия которых уже выполнены, и которые могут быть активированы в данный момент

// (вместе со списком фактов, наличие которых обращает условие данного правила в истину)

public readonly Dictionary<Rule, List<Fact>> Agenda = new ();

// Активированные правила, правая часть которых уже внесена в список действующих фактов

public readonly Dictionary<Rule, List<Fact>> ActivatedRules = new ();

public Rule[] NotActivatedRules => Rules.Where(rule => !ActivatedRules.ContainsKey(rule)).ToArray();

public MemoryComponent(IEnumerable<Rule> rules)

{

Rules = rules.ToArray();

Facts = new Dictionary<Fact, Rule>();

}

public void AddFact(Fact fact, Rule reason) => Facts.Add(fact, reason);

public void ActivateRule(Rule rule, List<Fact> requirements)

{

ActivatedRules.Add(rule, requirements);

RemoveFromAgenda(rule);

}

public void AddToAgenda(Rule rule, List<Fact> requirements)

{

if (Agenda.ContainsKey(rule)) RemoveFromAgenda(rule);

Agenda.Add(rule, requirements);

}

public void RemoveFromAgenda(Rule rule) => Agenda.Remove(rule);

}

### Файл InferenceComponent.cs

using System;

using System.Collections.Generic;

using System.Linq;

namespace ExpertSystem;

class InferenceComponent

{

private MemoryComponent Memory;

public InferenceComponent(MemoryComponent memory)

{

Memory = memory;

}

// Алгоритм вывода выбирает одно из правил, готовых к активации (условия которых выполнены),

// добавляет правую часть этого правила к списку действующих фактов

// если добавленный факт является результатом - завершает работу.

public Fact? GetResult()

{

UpdateAgenda();

while (Memory.Agenda.Any())

{

var (ruleToActivate, requirements) = GetOrderedAgenda().First();

Memory.ActivateRule(ruleToActivate, requirements);

if (ruleToActivate.Fact.Input is Input input) ruleToActivate.Fact.Value = AskQuestion(input.Question, input.Values);

Memory.AddFact(ruleToActivate.Fact, ruleToActivate);

if (ruleToActivate.Fact.IsResult) return ruleToActivate.Fact;

UpdateAgenda();

}

return null;

}

// Алгоритм разрешения конфликтов

// Правила отсортированы по порядку выполнения их условия (сначала самые новые)

private Dictionary<Rule, List<Fact>> GetOrderedAgenda() =>

Memory.Agenda

.Select((rule, i) => new

{

Rule = rule.Key,

OrderInAgenda = i,

Requirements = rule.Value,

})

.OrderByDescending(ruleWithOrdering => ruleWithOrdering.OrderInAgenda)

.ToDictionary(

ruleWithOrdering => ruleWithOrdering.Rule,

ruleWithOrdering => ruleWithOrdering.Requirements

);

private void UpdateAgenda()

{

foreach (var (rule, \_) in Memory.Agenda)

{

var result = rule.Condition.Evaluate(Memory.Facts.Keys);

if (!result.CanActivate)

{

Memory.RemoveFromAgenda(rule);

}

}

foreach (var rule in Memory.NotActivatedRules)

{

var result = rule.Condition.Evaluate(Memory.Facts.Keys);

if (result.CanActivate)

{

Memory.AddToAgenda(rule, result.Requirements);

}

}

}

// Принимает введенное пользователем значение, валидирует

// и при необходимости задает вопрос повторно.

public static string AskQuestion(string question, string[] inputValues)

{

Console.WriteLine(question);

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

{

Console.WriteLine($"{i + 1} - {inputValues[i]}");

}

string? value = null;

while (value == null)

{

var input = Console.ReadLine();

if (input == null) continue;

if (ParseInt(input) is int index && index >= 1 && index <= inputValues.Length)

{

value = inputValues[index - 1];

}

else

{

input = input.ToLower();

var foundValue = inputValues.FirstOrDefault(value => value.ToLower() == input);

if (foundValue != null) value = foundValue;

}

}

return value;

}

private static int? ParseInt(string input)

{

try

{

return int.Parse(input);

}

catch (FormatException)

{

return null;

}

}

}

### Файл ExplanationComponent.cs

using System;

using System.Collections.Generic;

using System.Linq;

namespace ExpertSystem;

class ExplanationComponent

{

private MemoryComponent Memory;

public ExplanationComponent(MemoryComponent memory)

{

Memory = memory;

}

// Алгоритм объяснения последовательно перебирает факты, определяет правило, из которого получен данный факт,

// и факты, наличие которых привело к активации правила.

public void GetReasoning(Fact resultFact)

{

Stack<Fact> stack = new();

stack.Push(resultFact);

while (stack.Any())

{

var fact = stack.Pop();

Console.WriteLine($"\nФакт: {fact}");

var reason = Memory.Facts[fact];

Console.WriteLine($"Добавлен в результате срабатывания правила: {reason}");

var requirements = reason is Rule rule ? Memory.ActivatedRules[rule] : null;

if (requirements is List<Fact> facts && facts.Any())

{

Console.WriteLine($"В связи с наличием фактов:\n{string.Join("\n", requirements)}");

facts.ForEach(requirement => stack.Push(requirement));

}

}

}

}

### Файл основной программы Program.cs

using System;

using System.IO;

using System.Text.Json;

using ExpertSystem;

// Загрузка набора правил из файла

var options = new JsonSerializerOptions { IncludeFields = true, IgnoreReadOnlyFields = true, IgnoreReadOnlyProperties = true };

options.Converters.Add(new ConditionConverter());

var jsonString = File.ReadAllText("../../../Rules.json");

var rules = JsonSerializer.Deserialize<Rule[]>(jsonString, options);

// Создание компонентов экспертной системы: память, механизм вывода, компонент объяснения

var memory = new MemoryComponent(rules);

var inferenceComponent = new InferenceComponent(memory);

var explanationComponent = new ExplanationComponent(memory);

// Запуск работы системы: общение с пользователем, вывод результата и объяснения

var result = inferenceComponent.GetResult();

if (result is Fact resultFact)

{

Console.WriteLine($"Результат:\n{resultFact.Name}");

Console.ReadKey();

explanationComponent.GetReasoning(resultFact);

}

else

{

Console.WriteLine("Результат не определен");

}

### Файл ConditionConverter.cs

using System;

using System.Text.Json;

using System.Text.Json.Serialization;

namespace ExpertSystem;

// JSON-конвертер условий. Нужен для правильного чтения подтипов Condition

internal class ConditionConverter : JsonConverter<Condition>

{

public override bool CanConvert(Type typeToConvert) => typeof(Condition).IsAssignableFrom(typeToConvert);

public override Condition Read(ref Utf8JsonReader reader, Type typeToConvert, JsonSerializerOptions options)

{

if (reader.TokenType != JsonTokenType.StartObject)

{

throw new JsonException($"Cannot convert JSON to type {typeToConvert}: invalid JSON structure");

}

reader.Read();

if (reader.TokenType != JsonTokenType.PropertyName)

{

throw new JsonException($"Cannot convert JSON to type {typeToConvert}: invalid JSON structure");

}

var propertyName = reader.GetString();

Condition condition;

switch (propertyName)

{

case nameof(ConditionExpression.Operator):

case nameof(ConditionExpression.Conditions):

condition = new ConditionExpression();

break;

case nameof(ValueCondition.FactName):

case nameof(ValueCondition.Value):

condition = new ValueCondition();

break;

default:

throw new JsonException($"Cannot convert JSON to type {typeToConvert}: property name {propertyName} not recognized");

}

do

{

if (reader.TokenType == JsonTokenType.EndObject)

{

return condition;

}

if (reader.TokenType == JsonTokenType.PropertyName)

{

propertyName = reader.GetString();

reader.Read();

switch (propertyName)

{

case nameof(ConditionExpression.Operator)

when condition is ConditionExpression conditionExpression:

var operatorString = reader.GetString();

if (operatorString is null) throw new JsonException($"Cannot convert JSON to type {typeToConvert}: property {propertyName} is not nullable");

conditionExpression.Operator = operatorString;

break;

case nameof(ConditionExpression.Conditions)

when condition is ConditionExpression conditionExpression:

var conditions = JsonSerializer.Deserialize<Condition[]>(ref reader, options);

if (conditions is null) throw new JsonException($"Cannot convert JSON to type {typeToConvert}: property {propertyName} is not nullable");

conditionExpression.Conditions = conditions;

break;

case nameof(ValueCondition.FactName) when condition is ValueCondition valueCondition:

var factName = reader.GetString();

if (factName is null) throw new JsonException($"Cannot convert JSON to type {typeToConvert}: property {propertyName} is not nullable");

valueCondition.FactName = factName;

break;

case nameof(ValueCondition.Value) when condition is ValueCondition valueCondition:

var value = JsonSerializer.Deserialize<string>(ref reader, options);

valueCondition.Value = value;

break;

default:

throw new JsonException($"Cannot convert JSON to type {typeToConvert}: property name {propertyName} not recognized");

}

}

} while (reader.Read());

throw new JsonException($"Cannot convert JSON to type {typeToConvert}: invalid JSON structure");

}

public override void Write(Utf8JsonWriter writer, Condition value, JsonSerializerOptions options)

{

throw new NotImplementedException();

}

}

## База знаний

[

{

"Condition": { "FactName": "Устройство" },

"Fact": {

"Name": "Устройство",

"Input": {

"Question": "К какому устройству вы будете подключать наушники?",

"Values": [

"Телефон",

"Компьютер",

"ТВ"

]

}

}

},

{

"Condition": { "FactName": "Назначение" },

"Fact": {

"Name": "Назначение",

"Input": {

"Question": "Для чего чаще всего будут использоваться наушники?",

"Values": [

"Звонки",

"Игры",

"Видео",

"Музыка"

]

}

}

},

{

"Condition": { "FactName": "Бюджет" },

"Fact": {

"Name": "Бюджет",

"Input": {

"Question": "Какой у вас бюджет?",

"Values": [

"1000",

"5000",

"10000"

]

}

}

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{ "FactName": "Вид" },

{

"FactName": "Устройство",

"Value": "Телефон"

}

]

},

"Fact": {

"Name": "Вид",

"Value": "Затычки"

}

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{ "FactName": "Вид" },

{

"Operator": "OR",

"Conditions": [

{

"FactName": "Устройство",

"Value": "Компьютер"

},

{

"FactName": "Устройство",

"Value": "ТВ"

}

]

}

]

},

"Fact": {

"Name": "Вид",

"Value": "Охватывающие"

}

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{ "FactName": "Беспроводные" },

{

"Operator": "OR",

"Conditions": [

{

"FactName": "Устройство",

"Value": "ТВ"

},

{

"Operator": "AND",

"Conditions": [

{

"FactName": "Устройство",

"Value": "Телефон"

},

{

"Operator": "OR",

"Conditions": [

{

"FactName": "Назначение",

"Value": "Звонки"

},

{

"FactName": "Назначение",

"Value": "Музыка"

}

]

}

]

},

{

"Operator": "AND",

"Conditions": [

{

"FactName": "Устройство",

"Value": "Компьютер"

},

{

"Operator": "OR",

"Conditions": [

{

"FactName": "Назначение",

"Value": "Игры"

},

{

"FactName": "Назначение",

"Value": "Видео"

},

{

"FactName": "Назначение",

"Value": "Музыка"

}

]

}

]

}

]

}

]

},

"Fact": {

"Name": "Беспроводные",

"Value": "Да"

}

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{ "FactName": "Беспроводные" },

{

"Operator": "OR",

"Conditions": [

{

"Operator": "AND",

"Conditions": [

{

"FactName": "Устройство",

"Value": "Телефон"

},

{

"Operator": "OR",

"Conditions": [

{

"FactName": "Назначение",

"Value": "Игры"

},

{

"FactName": "Назначение",

"Value": "Видео"

}

]

}

]

},

{

"Operator": "AND",

"Conditions": [

{

"FactName": "Устройство",

"Value": "Компьютер"

},

{

"FactName": "Назначение",

"Value": "Звонки"

}

]

}

]

}

]

},

"Fact": {

"Name": "Беспроводные",

"Value": "Нет"

}

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{ "FactName": "Микрофон" },

{

"Operator": "OR",

"Conditions": [

{

"FactName": "Назначение",

"Value": "Звонки"

},

{

"FactName": "Назначение",

"Value": "Игры"

}

]

}

]

},

"Fact": {

"Name": "Микрофон",

"Value": "Да"

}

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{ "FactName": "Микрофон" },

{

"Operator": "OR",

"Conditions": [

{

"FactName": "Назначение",

"Value": "Видео"

},

{

"FactName": "Назначение",

"Value": "Музыка"

}

]

}

]

},

"Fact": {

"Name": "Микрофон",

"Value": "Нет"

}

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Затычки"

},

{

"FactName": "Беспроводные",

"Value": "Нет"

},

{

"FactName": "Микрофон",

"Value": "Да"

},

{

"FactName": "Бюджет",

"Value": "1000"

}

]

},

"Fact": { "Name": "CADENA YH-12" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Затычки"

},

{

"FactName": "Беспроводные",

"Value": "Нет"

},

{

"FactName": "Микрофон",

"Value": "Нет"

},

{

"FactName": "Бюджет",

"Value": "1000"

}

]

},

"Fact": { "Name": "JVC HA-FX38" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Затычки"

},

{

"FactName": "Беспроводные",

"Value": "Да"

},

{

"FactName": "Бюджет",

"Value": "1000"

}

]

},

"Fact": { "Name": "QCY T1c" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Затычки"

},

{

"FactName": "Беспроводные",

"Value": "Нет"

},

{

"FactName": "Бюджет",

"Value": "5000"

}

]

},

"Fact": { "Name": "Sennheiser CX 300S" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Затычки"

},

{

"FactName": "Беспроводные",

"Value": "Да"

},

{

"FactName": "Бюджет",

"Value": "5000"

}

]

},

"Fact": { "Name": "JBL Wave 200TWS" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Затычки"

},

{

"FactName": "Бюджет",

"Value": "10000"

}

]

},

"Fact": { "Name": "Samsung Buds 2" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Охватывающие"

},

{

"FactName": "Беспроводные",

"Value": "Нет"

},

{

"FactName": "Бюджет",

"Value": "1000"

}

]

},

"Fact": { "Name": "A4Tech HS-200" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Охватывающие"

},

{

"FactName": "Беспроводные",

"Value": "Да"

},

{

"FactName": "Бюджет",

"Value": "1000"

}

]

},

"Fact": { "Name": "Harper HB-210" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Охватывающие"

},

{

"FactName": "Беспроводные",

"Value": "Да"

},

{

"FactName": "Микрофон",

"Value": "Нет"

},

{

"FactName": "Бюджет",

"Value": "5000"

}

]

},

"Fact": { "Name": "Soundcore Life 2 Neo Q10i" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Охватывающие"

},

{

"FactName": "Беспроводные",

"Value": "Да"

},

{

"FactName": "Микрофон",

"Value": "Да"

},

{

"FactName": "Бюджет",

"Value": "5000"

}

]

},

"Fact": { "Name": "SVEN AP-B900MV" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Охватывающие"

},

{

"FactName": "Беспроводные",

"Value": "Нет"

},

{

"FactName": "Бюджет",

"Value": "5000"

}

]

},

"Fact": { "Name": "SteelSeries Arctis 1" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Охватывающие"

},

{

"FactName": "Беспроводные",

"Value": "Нет"

},

{

"FactName": "Микрофон",

"Value": "Да"

},

{

"FactName": "Бюджет",

"Value": "10000"

}

]

},

"Fact": { "Name": "HyperX Cloud Alpha S HX-HSCAS" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Охватывающие"

},

{

"FactName": "Беспроводные",

"Value": "Да"

},

{

"FactName": "Микрофон",

"Value": "Да"

},

{

"FactName": "Бюджет",

"Value": "10000"

}

]

},

"Fact": { "Name": "JBL Tune 710BT" }

},

{

"Condition": {

"Operator": "AND",

"Conditions": [

{

"FactName": "Вид",

"Value": "Охватывающие"

},

{

"FactName": "Беспроводные",

"Value": "Нет"

},

{

"FactName": "Микрофон",

"Value": "Нет"

},

{

"FactName": "Бюджет",

"Value": "10000"

}

]

},

"Fact": { "Name": "Fostex T20RP MK3" }

}

]

# Лабораторная работа №3

## Постановка задачи

1. Взять предметную область из лабораторной работы №1.
2. Спроектировать БЗ в выбранном способе представления знаний (Семантическая сеть).
3. Сформулировать несколько запросов к системе, работающей с созданной БЗ.

## Вопросы к системе

1. Является ли «USB» «Звуковым устройством»?
2. Является ли «Lightning» «Устройством»?
3. Является ли «Sennheiser CX 300S» «Устройством записи»?
4. Какие есть типы «Способов подключения»?
5. Сколько существует «характеристик» у «Наушников»?
6. «Имеет» ли «Samsung Buds 2» «Шумоподавление»?
7. Какое «значение» характеристики «Цена» у «SVEN AP-B900MV»?
8. «Поддерживает» ли «Sennheiser CX 300S» такой же «Способы подключения», который «поддерживает» «iPhone 11»?
9. Сколько имеется «Наушников» типа «Охватывающие»?
10. Сколько «значений» характеристики «Цвет» имеют «Звуковые устройства» типа «Микрофон»?

## База знаний

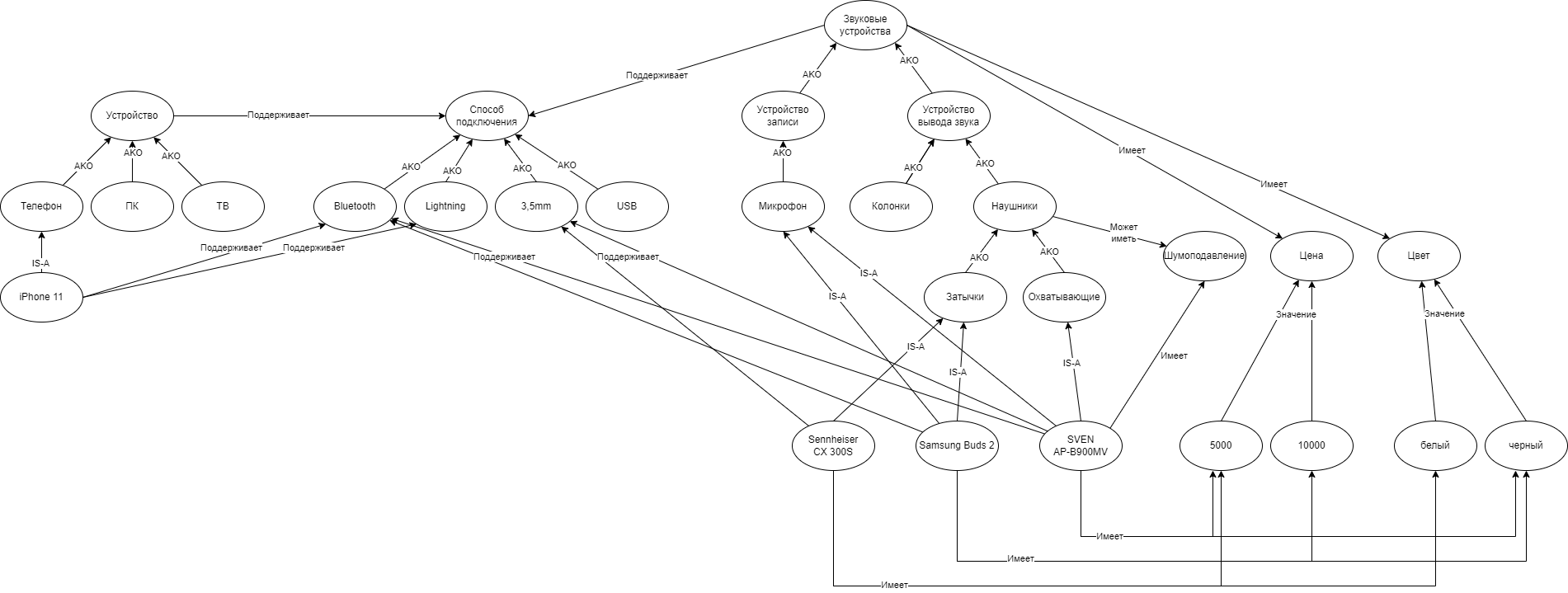


Рис. Семантическая сеть