|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ДОМАШНЯЯ РАБОТА**

**«Разработка простой экспертной системы»**

**ДИСЦИПЛИНА: «Экспертные системы»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-52Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Калашников А.С. )  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (      Амеличев Г.Э. )  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2022

**Цель**: формирование практических навыков по разработке экспертных систем.

**Задачи**:

1. получение навыка выделения знаний в заданной предметной области;
2. научиться реализовывать логику работы экспертной системы различными инструментами;
3. получить навыки по работе с правилами для обработки данных;
4. понимать формы представления знаний в ЭС.

**Вариант 5**

**Задание:**   
Разработать модель предметной области, указанной в варианте задания – выявить знания, которые необходимо представить в системе, разработать правила для обработки данных. Реализовать разработанную модель на любом высокоуровневом языке программирования. Разработать простую экспертную систему, позволяющую получить «экспертное мнение» в области, указанной вариантом задания.

**Требования к реализации:**

Экспертная система может быть реализована любыми доступными программными средствами. Помимо самого решения разработанная экспертная система должна сообщать пользователю как она пришла к полученному решению.

**Задание варианта:**

Экспертная система по выбору оборудования для компьютерной сети.

Входные данные:

1) количество компьютеров в сети;

2) топология сети;

3) пределы стоимости требуемого оборудования.

**Описание формы представления знаний:**

Экспертная система статическая и выполняет вопросно-ответное общение с пользователем

**Этапы обработки данных системой:**

Работа экспертной системы состоит в задаче вопросов и постепенном ответе на них, после чего идет отбор данных

**Исходный код разработанной программы (Python):**

from \_\_future\_\_ import print\_function

import copy

class Switchboard(object):

    name = None

    type = None

    count = None

    portSFP=None

    portQSFP=None

    cost = None

    purpose = None

    def \_\_init\_\_(self, name, type, count,portSFP,portQSFP, cost, purpose):

        self.name = name

        self.type = type

        self.count  = count

        self.portSFP=portSFP

        self.portQSFP=portQSFP

        self.cost = cost

        self.purpose = purpose

    def display(self):

        print("   "+self.name + ":\n" +

            "   Кол-во компьютеров: " + self.count + "\n" +

            "   Кол-во портов SFP: " + self.portSFP + "\n" +

            "   Кол-во портов QSFP+: " + self.portQSFP + "\n" +

            "   Топология сети: " + ','.join(self.purpose) + "\n" +

            "   Предел стоимости: " + self.cost + "\n" +

            "   Тип: " + self.type + "\n")

Counts = ["от 2 до 20 компьютеров", "от 20 до 40 компьютеров", "от 40 до 50 компьютеров"]

Topology = ["полносвязная", "ячеистая", "кольцевая", "звездообразная","смешанная","древовидная", "общая шина"]

Costs = ["от 20000р до 200000р", "от 200000р до 400000р", "от 400000р до 600000р"]

Characteristics = [Counts, Topology, Costs]

basic\_basic= Switchboard("Huawei S5735-L24T4S-A1","управляемый L2+",Counts[0],"1","0",Costs[0],[Topology[0],Topology[1],Topology[3],Topology[5], Topology[6]])

basic\_basic\_2 = Switchboard("D-Link DXS-1210-22C/A","управляемый",Counts[0],"1","0",Costs[0],[ Topology[2],Topology[4]])

basic\_optimal = Switchboard("D-Link DGS-3000-28XS","настраиваемый", Counts[0],"2","0",Costs[1],[ Topology[0],Topology[1],Topology[3],Topology[5],Topology[6]])

basic\_optimal\_2 = Switchboard("D-Link DXS-12-12SC/A","настраиваемый",Counts[0],"3","0",Costs[1],[ Topology[0],Topology[1],Topology[3],Topology[2],Topology[4]])

basic\_advanced = Switchboard("H3C LS-513S-52S-EI-GL","настраиваемый",Counts[0],"0","18",Costs[2],[ Topology[0],Topology[1],Topology[3],Topology[2],Topology[4],Topology[5],Topology[6]])

optimal\_basic = Switchboard("Zyxel NebulaFlex Pro GS2220-28-EU0101F","управляемый",Counts[1],"1","0",Costs[0],[Topology[0], Topology[3], Topology[5], Topology[6]])

optimal\_basic\_2 = Switchboard("Zyxel XGS2210-28-EU0101F","управляемый",Counts[1],"1","0",Costs[0],[Topology[1],Topology[2], Topology[4]])

optimal\_optimal = Switchboard("Zyxel NebulaFlex Pro GS2220-10HP-EU0101F","настраиваемый",Counts[1],"3","0",Costs[1],[Topology[0], Topology[1], Topology[2],Topology[4]])

optimal\_optimal\_2 = Switchboard("D-Link DGS-000-28XS","настраиваемый",Counts[1],"6","0",Costs[1],[Topology[3], Topology[5], Topology[6]])

optimal\_advanced = Switchboard("Zyxel NebulaFlex GS23-10HP-EU9990F","настраиваемый",Counts[1],"12","2",Costs[2],[Topology[0], Topology[1], Topology[2],Topology[4],Topology[3], Topology[5],Topology[6]])

advanced\_basic = Switchboard("H3C LS-5130S-52S-EI-GL","управляемый",Counts[2],"1","0",Costs[0],[Topology[0], Topology[1], Topology[4]])

advanced\_basic\_2 = Switchboard("H3C LS-5130S-52S-EI-GL","управляемый",Counts[2],"0","0",Costs[0],[Topology[5], Topology[6], Topology[2],Topology[3]])

advanced\_optimal = Switchboard("H3C LS-5130S-52S-EI-GL","настраиваемый",Counts[2],"4","0",Costs[1],[Topology[0], Topology[1], Topology[4]])

advanced\_optimal\_2 = Switchboard("H3C LS-5130S-52S-EI-GL","управляемый",Counts[2],"6","0",Costs[1],[Topology[5], Topology[6], Topology[2],Topology[3]])

advanced\_advanced = Switchboard("H3C S6520X-54QC-EI","настраиваемый",Counts[2],"48","2",Costs[2],[Topology[0], Topology[2], Topology[1], Topology[4], Topology[5],Topology[3],Topology[6]])

characteristicListSwitch = [

    basic\_basic,basic\_basic\_2,basic\_optimal,basic\_optimal\_2,basic\_advanced,

    optimal\_basic,optimal\_basic\_2,optimal\_optimal,optimal\_optimal\_2,optimal\_advanced,

    advanced\_basic,advanced\_basic\_2,advanced\_optimal,advanced\_optimal\_2,advanced\_advanced

]

Questions = \

    ["Какое будет количество компьютеров в сети?",

    "Какая будет топология сети?",

    "Каковы пределы стоимости требуемого оборудования?"]

Answers = \

    [["1 - от 10 до 20 компьютеров", "2 - от 20 до 40 компьютеров", "3 - от 40 до 50 компьютеров"],

    ["1 - Полносвязную", "2 - Ячеистую", "3 - Кольцевую","4 - Звездообразную","5 - Смешанную","6 - Древовидную","7 - Общую шину"],

    ["1 - от 200000р до 400000р", "2 - от 400000р до 800000р", "3 - от 800000р до 1000000р"]]

def filterDataCount(characteristicList):

    newSwitchboard = []

    for switch in characteristicList:

        if inputData[0] == switch.count:

            newSwitchboard.append(switch)

    return newSwitchboard

def filterDataPurpose(characteristicList):

    newSwitchboard = []

    for switch in characteristicList:

        if inputData[1] in switch.purpose:

            newSwitchboard.append(switch)

    return newSwitchboard

def filterDataCost(characteristicList):

    newSwitchboard = []

    for switch in characteristicList:

        if inputData[2] == switch.cost:

            newSwitchboard.append(switch)

    return newSwitchboard

listOfFilters = [filterDataCount, filterDataPurpose, filterDataCost]

inputData = []

resСharacteristicList = copy.deepcopy(characteristicListSwitch)

print("\nОтветьте пожалуйста на вопросы")

for i in range(len(Questions)):

    print("\n" + Questions[i])

    print(', '.join(Answers[i]))

    print("Ответ: ", end=' ')

    inputData.append(Characteristics[i][(int(input()) - 1)])

    resСharacteristicList = listOfFilters[i](resСharacteristicList)

print("\nВопросы:\n")

for i in range(len(Questions)):

    print(i + 1, "-", Questions[i])

print('\nБаза данных: \n')

print('№'.center(50), end='+')

print('{:^30}'.format(1), end='+')

print('{:^30}'.format(2), end='+')

print('{:^90}'.format(3))

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

for i in range(len(characteristicListSwitch)):

    print('\n', characteristicListSwitch[i].name.center(50), end='+')

    print('{:^30}'.format(characteristicListSwitch[i].count), end='+')

    print('{:^30}'.format(characteristicListSwitch[i].cost), end='+')

    print('{:^90}'.format(', '.join(characteristicListSwitch[i].purpose)), end='+')

    print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

print('\nВведеные данные: \n')

print("\n+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

print('№'.center(10), end='+')

print('{:^100}'.format("Характеристики конфигурации"), end='+')

print('{:^100}'.format("Введенные характеристики"))

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

print('{:^10}'.format(1), end='+')

print('{:^100}'.format(resСharacteristicList[0].count), end='+')

print('{:^100}'.format(inputData[0]))

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

print('{:^10}'.format(2), end='+')

print('{:^100}'.format(', '.join(resСharacteristicList[0].purpose)), end='+')

print('{:^100}'.format(inputData[1]))

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

print('{:^10}'.format(3), end='+')

print('{:^100}'.format(resСharacteristicList[0].cost), end='+')

print('{:^100}'.format(inputData[2]))

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

print('\nИТОГ: \n')

resСharacteristicList[0].display()

**Демонстрация работы программы:**

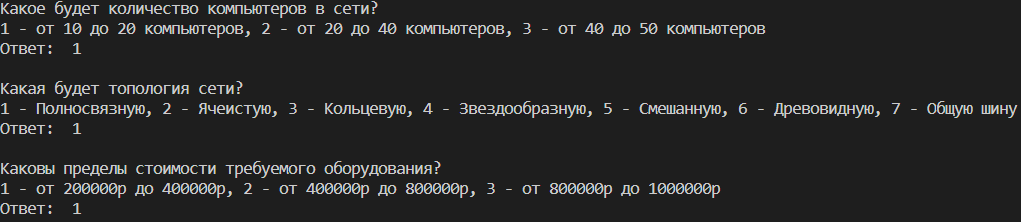


Рис.1. Процесс взаимодействия пользователя с экспертной системой

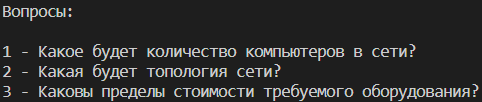


Рис.2. Вопросы

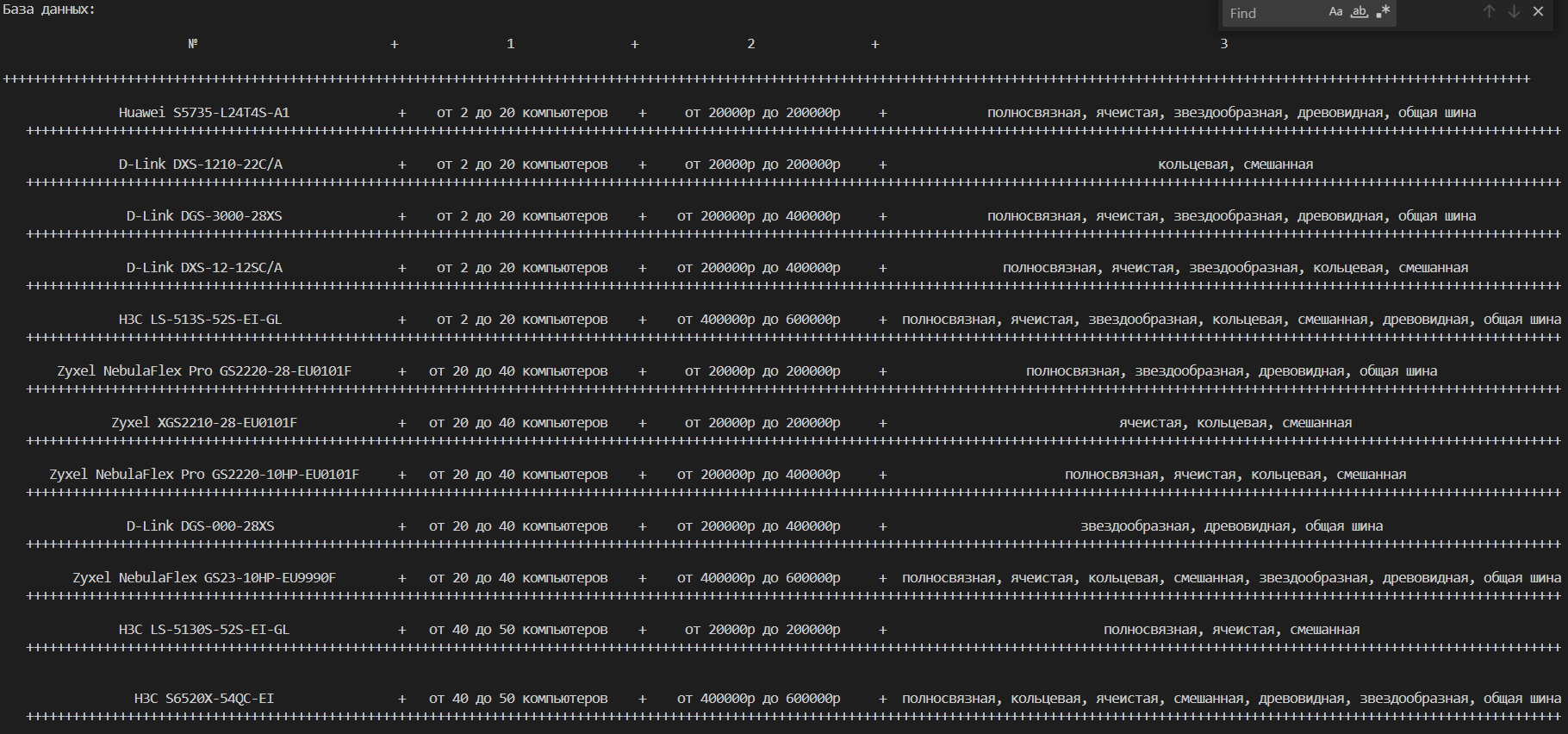


Рис.3. База данных

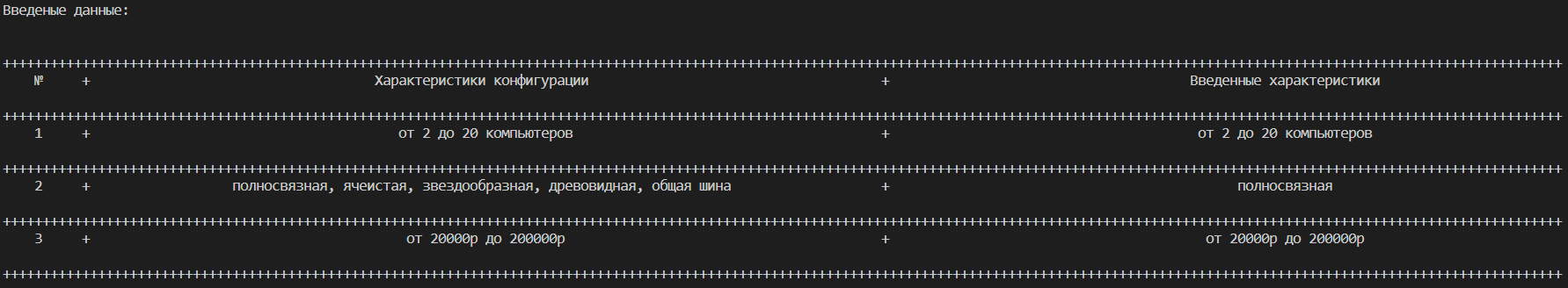


Рис.4. Введённые данные

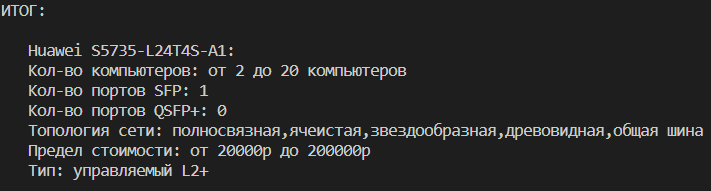


Рис.5. Итог

**Вывод:** в ходе выполнения данной домашней работы были приобретены практические навыки по разработке экспертных систем.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Малышева Е.Н. Экспертные системы [Электронный ресурс]: учебное пособие по специальности 080801 «Прикладная информатика (в информационной сфере)»/ Малышева Е.Н.— Электрон. текстовые данные.— Кемерово: Кемеровский государственный институт культуры, 2010.— 86 c.— Режим доступа: http://www.iprbookshop.ru/22126.— ЭБС «IPRbooks»
2. Павлов С.Н. Системы искусственного интеллекта. Часть 1 [Электронный ресурс]: учебное пособие/ Павлов С.Н.— Электрон. текстовые данные.— Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011.— 176 c.— Режим доступа: http://www.iprbookshop.ru/13974.— ЭБС «IPRbooks»
3. Павлов С.Н. Системы искусственного интеллекта. Часть 2 [Электронный ресурс]: учебное пособие/ Павлов С.Н.— Электрон. текстовые данные.— Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011.— 194 c.— Режим доступа: http://www.iprbookshop.ru/13975.— ЭБС «IPRbooks»
4. Чернышов, В.Н. Системный анализ и моделирование при разработке экспертных систем : учебное пособие / В.Н. Чернышов, А.В. Чернышов ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2012. - 128 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=277638 (22.02.2017).

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Воронов, А.Е. Технология использования экспертных систем / А.Е. Воронов. - М. : Лаборатория книги, 2011. - 109 с. : ил. - ISBN 978-5-504-00525-6 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=142527](https://biblioclub.ru/index.php?page=book&id=142527) (22.02.2017).
2. Трофимов, В.Б. Интеллектуальные автоматизированные системы управления технологическими объектами : учебно-практическое пособие / В.Б. Трофимов, С.М. Кулаков. - Москва-Вологда : Инфра-Инженерия, 2016. - 232 с. : ил., табл., схем. - Библиогр. в кн.. - ISBN 978-5-9729-0135-7 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=444175](https://biblioclub.ru/index.php?page=book&id=444175) (22.02.2017).
3. Интеллектуальные и информационные системы в медицине: мониторинг и поддержка принятия решений : сборник статей / . - М. ; Берлин : Директ-Медиа, 2016. - 529 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-7150-4 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=434736](https://biblioclub.ru/index.php?page=book&id=434736) (22.02.2017).
4. Джарратано Дж., Райли Г. Экспертные системы. Принципы разработки и программирование, 4-е издание.: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2007. – 1152 с.: ил. – Парал. тит. англ.

**Электронные ресурсы:**

1. https://ru.wikipedia.org/wiki/CLIPS - CLIPS — Википедия
2. http://clipsrules.sourceforge.net/ - A Tool for Building Expert Systems (англ.)
3. http://clipsrules.sourceforge.net/WhatIsCLIPS.html - What is CLIPS? (англ.)