


# Методические особенности использования интеллектуальных программных средств для систематизации и контроля знаний

*Касторнова Василина Анатольевна*, кандидат педагогических наук, доктор философии в области информатизации образования (Ph.D), доцент, ведущий научный сотрудник ЦИО ФГБНУ «Институт стратегии развития образования РАО»

**Интеллектуальные  
программные средства –  
систематизация и контроль  
знаний**






Интеллектуальные программные средства (ИПС), созданные на основе систем искусственного интеллекта, можно использовать в различных предметных областях для идентификации и распознавания объектов. С помощью систем ИИ можно решать и слабо формализуемые задачи из разных областей знаний, среди которых можно указать распознавание объектов по их характерным признакам.

При решении таких задач основным является выделение **характерных свойств (признаков, атрибутов)** этих объектов. Это распознавание можно осуществлять с помощью построения **экспертных систем и нейронных сетей**, являющихся представителями систем ИИ.

Примером использования ИИ при изучении предметной области «Информатика» может служить **изучение вопросов аппаратного и программного обеспечения ЭВМ**, таких как умение классифицировать: **типы ЭВМ, виды программного обеспечения (системного и прикладного), типы языков программирования**, а также основанные на **этих языках системы программирования.**



Рассмотрим методические подходы применения интеллектуальных программных средств для решения обозначенного круга задач по распознаванию объектов некоторой предметной области с целью систематизации и контроля знаний. Практика обучения показывает, что эти средства целесообразно использовать в виде следующих 5-ти форм работы с ними:

# Формы использования ИПС для целей обучения

## 1. Работа с созданной и обученной ИПС

Эта форма работы используется для контроля знаний, когда обучаемому ставится задание распознать все предусмотренные ИПС исходы. При этом преподаватель в случае работы с ИПС может просмотреть протокол работы обучаемого с системой.

## 2. Обучение ИПС

Здесь преподаватель вводит в БЗ только переменные и исходы, а учащийся должен обучить ее до получения правильных ответов. Можно также частично обучить БЗ и предложить учащимся добиться получения от нее безошибочных ответов путем ввода достаточного числа обучающих примеров.

## 3. Модификация БЗ ИПС

На этом этапе работы с готовой БЗ ставится задача изучения работоспособности системы, выявления лишних переменных, удаления их из базы и проведения последующего переобучения.

## 4. Добавление к БЗ ИПС новых объектов

Эта форма работы полезна тем, что учащиеся, добавляя новый объект, должны увидеть, достаточно ли существующих в БЗ признаков для идентификации нового объекта путем переобучения ИПС, или же необходимо дополнить этот список новой(ми) переменными.

## 5. Создание нового ИПС

В этом случае учащийся должен предварительно продумать список всех переменных и исходов, а затем ввести их в БЗ ИПС и произвести его обучение.

# 1. Работа с созданной и обученной ИПС

## 1.1. Экспертная система

При работе с ЭОС сначала запускается ее оболочка **obolochka\_ES**, расположенная в рабочей папке обучаемого (см. слайд 8), затем пунктом 7 меню загружается экспертная система и пунктом 5 (см. слайд 9) осуществляется переход на нормальное функционирование системы.

5. Нормальное функционирование системы

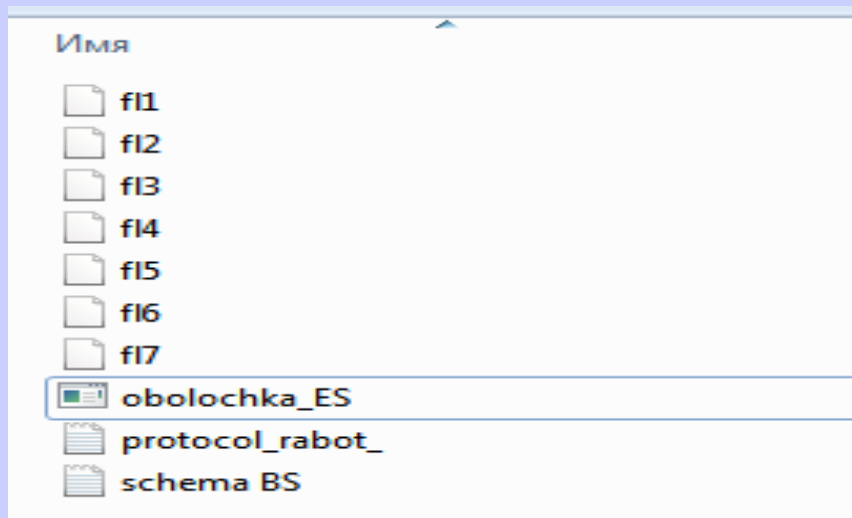
6. Запоминание текущего состояния системы

7. Загрузка экспертной системы

## Рабочая папка пользователя

Рабочая папка пользователя с расположенной в ней экспертной системой представляет собой следующий набор файлов:

1. Файлы fl1...fl7 есть файлы, содержащие БЗ.
2. Файл оболочка\_ES является запусковым для ЭОС.



3. Протокол работы с экспертной системой protocol\_rabot.
4. Схема базы знаний ЭОС schema BS.



# Структура разработанной ЭОС

Модули ЭС,  
начальный экран










## ЭКСПЕРТНАЯ СИСТЕМА

1. Инициализация системы
2. Ввод примера
3. Тренировка системы
4. Обучение системы
5. Нормальное функционирование системы
6. Запоминание текущего состояния системы
7. Загрузка экспертной системы
8. Контроль правил и примеров
9. Добавление и удаление новых переменных и исходов
10. Запись на диск схем ЭС
11. ВЫХОД

Выберите нужный пункт (введите соответствующую цифру)

База знаний,  
система файлов



Имя	Дата изменения
 fl1	17.04.2017 0:56
 fl2	17.04.2017 0:56
 fl3	17.04.2017 0:56
 fl4	17.04.2017 0:56
 fl5	17.04.2017 0:56
 fl6	17.04.2017 0:56
 fl7	17.04.2017 0:56

# Знакомство со структурой БЗ ЭОС

Перед началом работы по узнаванию объектов (исходов) пользователь имеет возможность ознакомиться со структурой БЗ, которая включает в себя информацию о числе *узлов*, а также *переменные* и *исходы* для каждого узла. Сделать это можно двумя способами:

1. Активизировать пункт 8 «Контроль правил и примеров» оболочки ЭОС (слайд 9).
2. Открыть и просмотреть файл *schema BS*.

## Нормальное функционирование системы

Этот модуль предназначен преимущественно для работы с ЭОС обучаемого (пользователя), а не разработчика. При запуске модуля на экране появляются вопросы, которые система последовательно ставит пользователю. Он сообщает ей, какие признаки у объекта, который идентифицируется/узнается с помощью ЭОС. Ответы пользователя должны быть точны, иначе ЭОС не даст правильного ответа (исхода). При многоузловой ЭОС сначала выдается исход 1-го узла, а потом – окончательный ответ. На слайде 12 представлен сеанс работы по узнаванию **ПО Delphi**.

Экспертная система КЛАССИФИКАЦИЯ ПО ЭВМ

Список исходов (определяемых объектов):

1. MS DOS
2. WINDOWS XP
3. ASSEMBLER
4. BASIC
5. PASCAL
6. DELPHI
7. PROLOG
8. PAINT
9. POWER POINT
10. MATHCAD
11. WINDOWS 3.1
12. WINDOWS NT
13. MS WORD
14. MS EXCEL
15. MS ACCESS

Введите название исхода, что Вы задумали определить - DELPHI\_

Узел 1

Свойство "СОЗДАЕТ ПРОГРАММЫ" есть у объекта " DELPHI"? <1/0> 1

Свойство "СОЗДАЕТ ИСПОЛНЯЕМЫЕ ФАЙЛЫ" есть у объекта " DELPHI"? <1/0> 1

Свойство "НУЖДАЕТСЯ В ТРАНСЛЯЦИИ" есть у объекта " DELPHI"? <1/0> 1

Предлагается "СИСТЕМА ПРОГРАММИРОВАНИЯ" в качестве возможного исхода

Узел 2

Свойство "НИЗКИЙ УРОВЕНЬ ЯЗЫКА" есть у объекта " DELPHI"? <1/0> 0

Свойство "НАДСТРОЙКА НАД PASCAL" есть у объекта " DELPHI"? <1/0> 1

Свойство "СОЗДАЕТ ТЕКСТЫ" есть у объекта " DELPHI"? <1/0> 0

Свойство "СОЗДАЕТ ЭТ" есть у объекта " DELPHI"? <1/0> 0

Свойство "РЕШАЕТ МАТЕМАТИЧЕСКИЕ ЗАДАЧИ" есть у объекта " DELPHI"? <1/0> 0

Свойство "СЕТЕВАЯ НАДСТРОЙКА НАД DOS" есть у объекта " DELPHI"? <1/0> 0

Свойство "ПРОЦЕДУРНЫЙ ЯЗЫК" есть у объекта " DELPHI"? <1/0> 1

Свойство "СОЗДАЕТ БЗ" есть у объекта " DELPHI"? <1/0> 0

Предлагается "DELPHI" в качестве возможного исхода

## Замечание

Следует заметить, что при работе с ЭОС в нормальном режиме система использует не все свойства объектов, которые указаны в исходной таблице для построения ЭОС. Сюда входят только те из них, которые позволяют сети отличить один объект от другого с учетом того, что они могут иметь одинаковые характерные признаки. А это позволяет, с методической точки зрения, обратить внимание обучаемых на важные (уникальные) свойства изучаемых объектов в той или иной предметной области.

## Методические рекомендации

При получении неправильного ответа пользователь может повторить сеанс узнавания выбранного им исхода или перейти к другому объекту. По завершению работы с ЭОС имеется возможность просмотра введенных обучаемым ответов путем загрузки текстового файла **protocol\_rabot**. Просмотр протокола позволяет обучаемому найти свои допущенные им ошибки, а преподавателю оценить его знания (интеллектуальное тестирование) по теме, заложенной в базу знаний ЭОС.

# Протокол работы с ЭОС

*Введите название исхода, что Вы задумали определить - "DELPHI"*

## **Узел 1**

Свойство "Создает программы" есть у объекта "DELPHI"? <1/0> 1

Свойство "Решает практические задачи" есть у объекта "DELPHI"? <1/0> 0

Свойство "Создает исполняемые файлы" есть у объекта "DELPHI"? <1/0> 1

*Предлагается "Система программирования" в качестве возможного исхода*

## **Узел 2**

Свойство "Использует режим форм" есть у объекта "DELPHI"? <1/0> 1

Свойство "Логический язык" есть у объекта "DELPHI"? <1/0> 0

Свойство "Процедурный язык" есть у объекта "DELPHI"? <1/0> 1

Свойство "Содержал первый компилятор" есть у объекта "DELPHI"? <1/0> 0

Свойство "Операторный язык" есть у объекта "DELPHI"? <1/0> 1

Свойство "Надстройка над PASCAL" есть у объекта "DELPHI"? <1/0> 1

*Предлагается "DELPHI" в качестве возможного исхода*

*Верно*

*Вы хотите продолжить нормальное функционирование? [y/n]*

# Методические возможности ЭОС

Экспертные обучающие системы призваны, в отличие от традиционных тестов, «интеллектуализировать» диагностику знаний, где обучающийся не выбирает правильный ответ из предлагающихся вариантов, а последовательно отвечает на ряд вопросов системы. Успех обучения зависит от квалификации эксперта (преподавателя), который в каждом случае контроля знаний должен грамотно построить узлы системы, сформулировать в них вопросы (переменные), ответы (исходы) и обучить полученную систему, то есть сформировать соответствующую базу знаний.

Мы считаем, что таким образом организованная и функционирующая ЭОС обладает большими методическими возможностями для усвоения, систематизации и контроля знаний по различным учебным дисциплинам. Обучение с помощью ЭОС ориентировано на извлечение знаний самим обучаемым.



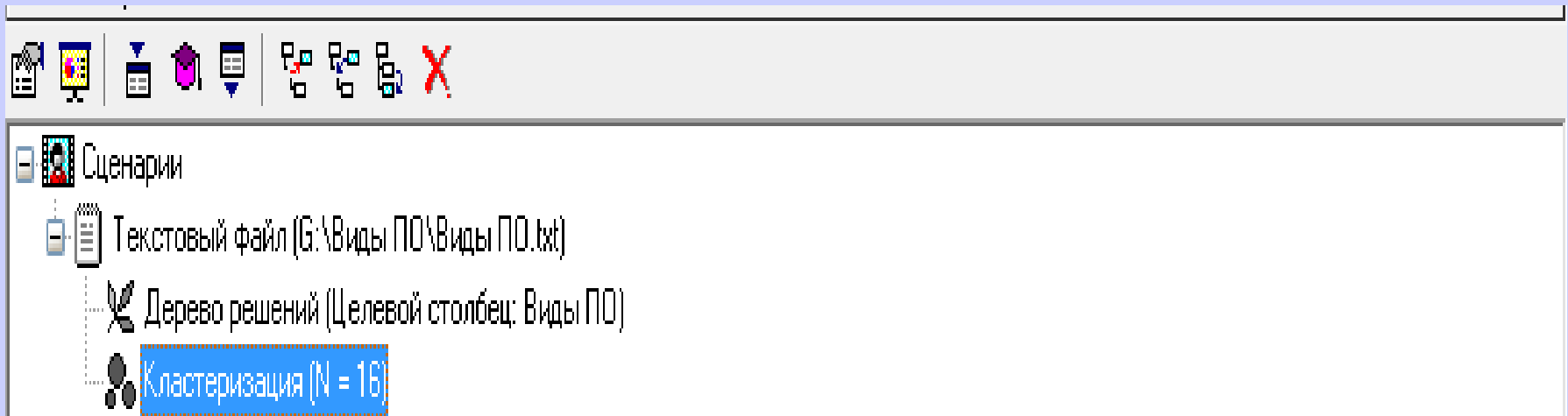
## 1.2. Нейронная сеть

Работа с ИПС в нейронной сети осуществляется путем запуска проекта с названием соответствующей БЗ, расположенной в рабочей папке пользователя:



ПО ЭВМ.ded

При активизации этого файла на экране появляется главное меню **Deductor**, в котором представлены текстовый файл в виде таблицы с указанием атрибутов и выходов (определяемых объектов), а также две формы Мастеров эмулятора сети: *Дерево решений* и *Кластеризация*.



# Дерево решений

В этом Мастере для осуществления процесса узнавания объектов по их атрибутам используется **Визуализатор «Что-если»**, позволяющий узнавать объект в интерактивном режиме по набору нужных свойств. Например, вида **ПО Delphi**:

Поле	Значение
<b>Входные</b>	
ab Создает папки	нет
ab Загружается автоматически	нет
ab Многозадачный режим работы	нет
ab Графический интерфейс	да
ab Создает исполняемые файлы	да
ab Создает программы	да
ab Производит трансляцию	да
ab Создает тексты	нет
ab Создает БД	нет
ab Создает графику	нет
ab Создает БЗ	нет
ab Создает презентации	нет
ab Процедурный язык	да
ab Операторный язык	да
ab Низкий уровень языка	да
ab Поддерживает работу сети	нет
ab Решает математические задачи	нет
ab Надстройка над Pascal	да
ab Сетевая надстройка над MS DOS	нет
<b>Выходные</b>	
ab Виды ПО	DELPHI

# Кластеризация

В этом Мастере входом являются сами объекты, а на выходе перечисляются их характерные признаки (атрибуты), заложенные в БЗ нейронной сети, что с методической точки зрения позволяет указать обучаемому на те атрибуты, которые являются значимыми для того или иного объекта:

Входные	
ab ВИДЫ ПО	8.DELPHI
Выходные	
ab 1. Создает папки	нет
ab 2. Загружается автоматически	нет
ab 3. Многозадачный режим работы	нет
ab 4. Графический интерфейс	да
ab 5. Создает исполняемые файлы	да
ab 6. Создает программы	да
ab 7. Нуждается в трансляции	да
ab 8. Создает тексты	нет
ab 9. Создает БД	нет
ab 10. Создает графику	нет
ab 11. Создает БЗ	нет
ab 12. Создает анимации	нет
ab 13. Процедурный язык	да
ab 14. Операторный язык	да
ab 15. Низкий уровень языка	нет
ab 16. Поддерживает работу сети	нет
ab 17. Решает математические задачи	нет
ab 18. Надстройка над Pascal	да
ab 19. Сетевая надстройка над DOS	нет

# Методические рекомендации

Эта форма работы с нейронной сетью, как и с ЭОС, направлена на отработку знаний учащихся характерных признаков узнаваемых с ее помощью объектов из какой-либо предметной области. Однако здесь есть существенные отличия в их работе. Если в ЭОС для узнавания объекта нужно указать если не все, но, крайней мере, несколько его свойств, то в НС объект может быть определен одним или двумя характерными признаками – уникальными атрибутами. Более того, возможен случай, когда объект определяется по отсутствию всех свойств, находящихся в БЗ, что невозможно для ЭОС. В ее БЗ должна быть хотя одна переменная, характеризующая каждый исход. Данные особенности работы с ИПС также важны для понимания характерных признаков изучаемых объектов.

## 2. Обучение ИПС

### 2.1. Экспертная система

Здесь преподаватель вводит в БЗ только переменные и исходы, а учащийся должен обучить ее до получения правильных ответов. Можно также частично обучить БЗ и предложить учащимся добиться получения от нее безошибочных ответов путем ввода достаточного числа обучающих примеров.

Эта форма работы осуществляется с помощью загрузки неполной базы знаний с помощью пункта 7 (слайд 9), а затем активизации модулей:

2. Ввод примеров
3. Тренировка системы
4. Обучение системы.

## Ввод примеров

Обучаемый, вводя примеры, проверяет свои знания в области характерных признаков узнаваемых объектов, при этом система выдает все переменные, входящие в загруженную структуру базы знаний.

ЭКСПЕРТНАЯ СИСТЕМА Классификация ПО ЭВМ

УЗЕЛ 1

ВВОД примеров :

ВНИМАНИЕ – число примеров равно количеству исходов данного узла!!!

Сколько примеров приготовлено для этого узла? – 3

ПРИМЕР 1 для узла 1

Свойство "Управляет работой ЭВМ" есть у объекта ""? <1/0> 1

Свойство "Решает практические задачи" есть у объекта ""? <1/0> 0

Свойство "Создает программы" есть у объекта ""? <1/0> 0

Свойство "Создает исполняемые файлы" есть у объекта ""? <1/0> 0

Свойство "Осуществляет трансляцию" есть у объекта ""? <1/0> 0\_

Укажите допустимый исход:

- 1 Системное ПО
- 2 Прикладное ПО
- 3 Система программирования

с указанием его номера 1

## Тренировка системы

С помощью этого модуля на основании введенных примеров создается массив правил для работы системы как эксперта по узнаванию объектов.

ТЕКУЩИЕ ПРАВИЛА ЭС:

1. Системное ПО    2. Прикладное ПО    3. Система программирования

1. Управляет работой ЭВМ	1	-1	-1	
2. Решает практические задачи		-1	1	-1
3. Создает программы	-2	0	0	
4. Создает исполняемые файлы		-1	-1	1
5. Осуществляет трансляцию		-1	-1	1

По этой таблице обучаемый должен определить, соответствует ли сформированный на основе введенных им ответов массив правил содержанию решаемой задачи. В этой таблице неотрицательность коэффициента (веса) в строке должна соответствовать наличию этого свойства нужному исходу.

## Обучение системы

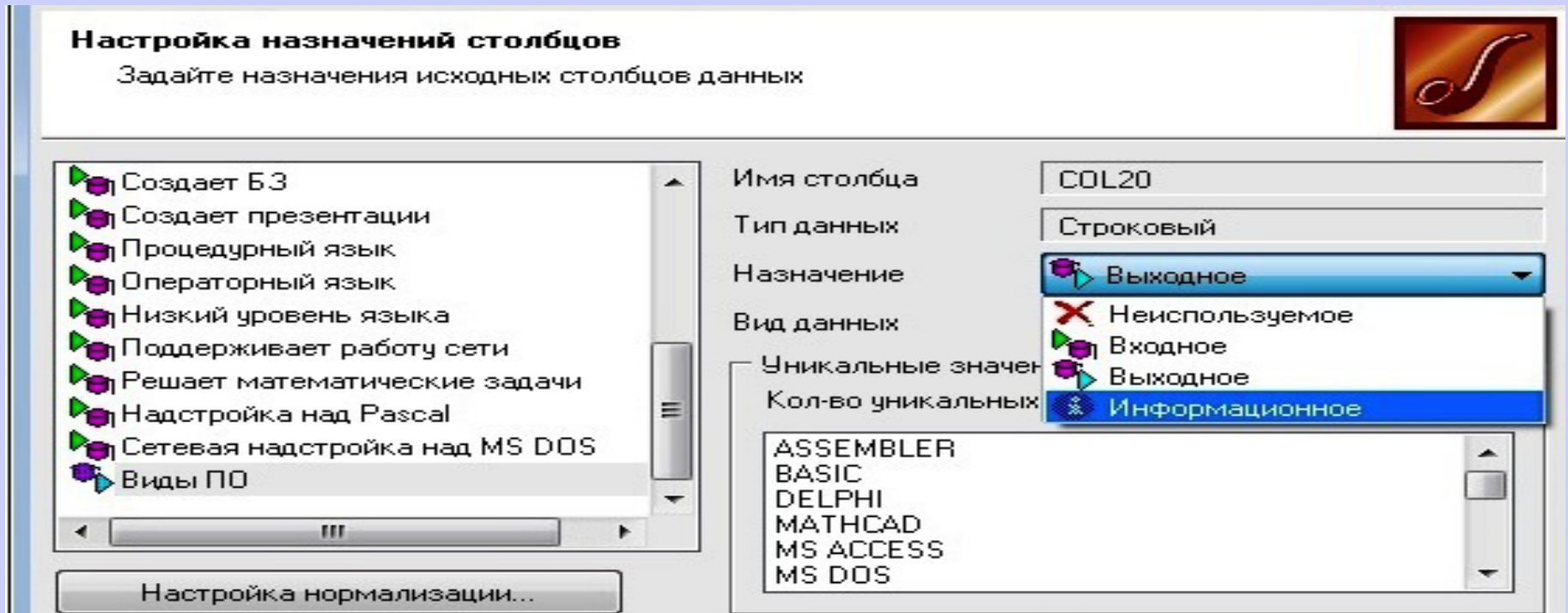
Если после ввода примеров и вычисления массива правил система работает не надежно и дает неправильные варианты исходов, то обучаемому рекомендуется переобучить систему, вводя дополнительные примеры. После ввода каждого примера система спрашивает, надо ли запоминать ответ, и если да, то она сама осуществляет коррекцию массива правил, который изменяет свои значения в тех строках, где речь идет о переменных, относящихся к свойствам неправильно узнанных исходов. Для надежности рекомендуется вводить один и тот же пример дважды. Можно дать столько же примеров, как было при вводе примеров на втором этапе. Можно же ограничиться примерами только для тех случаев, когда система выдает ошибочные исходы. По завершению сеанса обучения результаты записываются на диск в пункте 6 (слайд 9).



## 2.2. Нейронная сеть

### 2.2.1. Настройка назначения столбцов


На этом этапе работы с НС пользователь с помощью одного из Мастеров строит БЗ сети с помощью исходной таблицы, имеющейся в его распоряжении (подготовленную преподавателем). Первоначально производится **Настройка назначения столбцов** с точки зрения их принадлежности к входным и выходным данным.



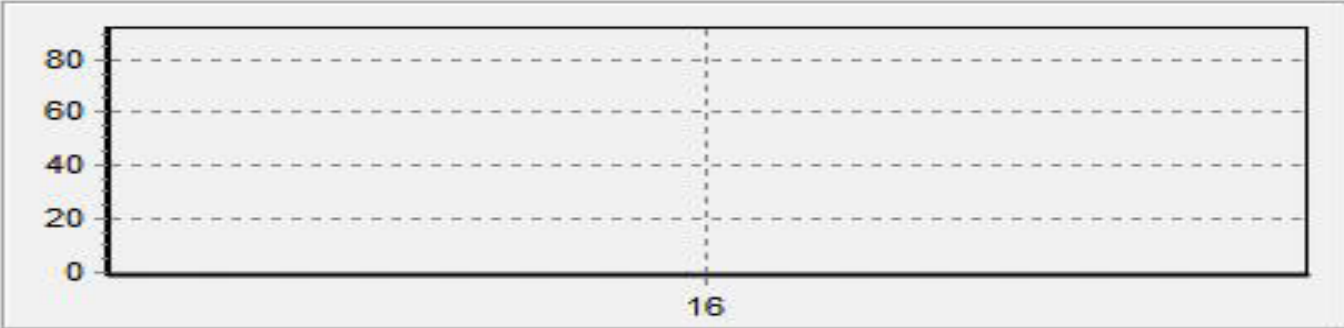
## 2.2.2. Построение БЗ нейронной сети

Для построения БЗ достаточно нажать на кнопку Пуск. На этом слайде показано, что НС успешно построена. Это видно из 100% распознавания. При неполном распознавании объектов необходимо вернуться назад и изменить исходную таблицу данных (добавить/удалить столбцы, изменить значения в ячейках и пр.).

**Построение дерева решений**  
Запустите процесс построения дерева решений



Распределено, шт.	Распознано, %	Кол-во узлов
<input type="checkbox"/> Распознано	Обучающее мн-во	<input type="text" value="31"/>
<input type="checkbox"/> Нераспознано	Тестовое мн-во	<input type="text" value="16"/>
<input type="text" value="91"/>	<input type="text" value="100,00"/>	Время обучения
<input type="text" value="0"/>	<input type="text" value="100,00"/>	<input type="text" value="00:00:00"/>
		Темп обновления
		<input type="text" value="100"/>



100%

### 2.2.3. Использование визуализаторов

Причины неправильного распознавания объектов можно определить с помощью визуализаторов эмулятора Deductor. Эффективным для этой цели является Визуализатор «Таблица сопряженности». На рисунке видно, что все объекты классифицированы. В случае ошибки два объекта попали бы в одну строку. Это означало бы, что в таблице данных не хватает атрибутов, которые бы отличали один объект от другого.

Таблица сопряженности					
Фактически	Классифицировано				
	Прикладное	Система программирования	Системное	Утилита	Итого
Прикладное	20				
Система программирования		20			
Системное			20		
Утилита				20	
Итого	20	20	20	20	

# Методические рекомендации

При обучении нейронной сети с готовой таблицей исходных данных можно преследовать две цели:

1. Научить учащихся освоить все этапы построения (а это и есть обучение) базы знаний, проверки эффективности ее работы при правильной таблице исходных данных.
2. Научить построению нейронной сети с поиском и исправлением неточностей в исходной таблице, используя для этого различные визуализаторы системы Deductor.

## 3. Модификация БЗ ИПС

На этом этапе работы с готовой БЗ ставится задача изучения работоспособности системы, выявления лишних переменных (атрибутов), удаления их из базы или добавления новых и проведения последующего переобучения.

### 3.1. Экспертная система

Переменные рекомендуются удалять только тогда, когда в процессе обучения система совсем не задает вопросов по наличию у исходов какого-либо признака – значит он не существенен, его можно удалить. Добавлять новые переменные нужно тогда, когда системе не удастся отличить один объект от другого – нужна новая уточняющая переменная. Это осуществляется с помощью модуля 8 «Удаление и добавление новых переменных и исходов».

# Модификация структуры БЗ ЭОС

При модификации ЭОС можно в каждом ее узле по их номерам, (в файле **schema BS**), можно удалять переменные и исходы. При необходимости, наоборот, добавлять в БЗ новые данные. На рисунке показана вставка одного нового исхода во второй узел ЭОС.

## ЭКСПЕРТНАЯ СИСТЕМА

Для модификации экспертной системы ответьте на следующие вопросы:

ВВОД и УДАЛЕНИЕ новых переменных и исходов

УЗЕЛ 1

Сколько новых переменных приготовлено для этого узла? - 0

Нужно ли удалять переменные? <y/n>n

Сколько новых исходов приготовлено для этого узла? - 0

Нужно ли удалять исходы? <y/n>n

ВВОД и УДАЛЕНИЕ новых переменных и исходов

УЗЕЛ 2

Сколько новых переменных приготовлено для этого узла? - 0


Нужно ли удалять переменные? <y/n>n

Сколько новых исходов приготовлено для этого узла? - 1

# Методические рекомендации

Эта форма работы может использоваться для решения двух основных задач создаваемой БЗ ЭОС:

I. **Тестирование ЭОС.** Это делается с помощью п. 5 «Нормальное функционирование системы». Если при этом обнаруживаются неправильно опознаваемые объекты, то рекомендуется дообучить систему с помощью п. 4 «Обучение системы». Иногда многократное дообучение не приводит к устранению ошибки. Например, может случиться так, что система не отличает друг от друга два близких объекта, поэтому для их разграничения нужно добавить дополнительное свойство (переменную). После этой операции следует систему переобучить (п. 2) или дообучить (п. 4).



II. Минимизация числа переменных. После обучения и тестирования ЭОС следует минимизировать структуру БЗ, то есть удалить из нее незначимые переменные. Это означает приведение информационной модели, оставив в ней только существенные переменные. Термин «существенные» понимается не с точки зрения сущности того или иного объекта, а с точки зрения его взаимоотношения с другими объектами изучаемой предметной области. Поиск лишних переменных можно вести как с помощью самой оболочки экспертной системы во время тестирования ЭОС, так и с помощью протокола работ, в котором можно увидеть лишние переменные.




## 3.2. Нейронная сеть

### 3.2.1. Тестирование НС

Тестирование нейронной сети осуществляется самой системой Deductor автоматически. Пользователь имеет возможность только изменять процентное соотношение между обучающей и тестирующей выборками. Увеличение тестирующей выборки повышает надежность работы НС.

**Разбиение исходного набора данных на подмножества**

Настройте разбиение исходного множества данных на обучающее и тестовое множества



Способ разделения исходного множества данных:

Столбец для разделения исходного множества:

Множество	Размер		Порядок сортировки
	В процентах	В строках	
<input checked="" type="checkbox"/> Обучающее	90,00	89	По возрастанию
<input checked="" type="checkbox"/> Тестовое	10,00	10	По возрастанию
ИТОГО:	100,00	99	

Navigation icons: A-Z, Z-A, Up, Down











### 3.2.2. Минимизация числа атрибутов

В Deductor **Визуализатор «Значимость атрибутов»** показывает, какие атрибуты значимы при проведении классификации объектов.

Целевой атрибут: Виды ЭВМ			
№	Номер	Атрибут	Значимость, %
1	6	6.Штучное производство	
2	9	9.Цветной дисплей	
3	8	8.Отечественная ЭВМ	
4	1	1.Предшественники ПК	
5	7	7.Карманная версия ПК	
6	13	13. Построена на транзисторах	
7	4	4.Первая советская ЭВМ 3-го пок...	
8	2	2.Открытость архитектуры	
9	5	5.Wi-Fi доступ к сети	
10	12	12. Наличие оптического диска	
11	11	11. Наличие сетевой ОС	
12	3	3.Пульт управления	
13	10	10.Наличие ОС	

На рисунке видно, что для решения задачи классификации видов ЭВМ достаточно 9 атрибутов. Удаление лишних атрибутов проводится либо средствами Deductor, либо путем изменения таблицы данных в текстовом редакторе. Затем строится новая НС (см. слайд 48).

Полученная НС сеть формально не содержит лишних атрибутов, однако ее использование в целях обучения методически не оправдано, так как в ней объект можно распознать по одному свойству, что не дает полной картины характеристики определяемых объектов (видов ЭВМ).

Целевой атрибут: Виды ЭВМ				
№	Номер	Атрибут	Значимость, %	/
1	7	8.Отечественная ЭВМ		19,037
2	5	6.Штучное производство		16,295
3	1	1.Предшественники ПК		13,791
4	8	10.Наличие ОС		9,732
5	3	4.Первая советская ЭВМ 3-го пок...		9,265
6	6	7.Карманная версия ПК		9,055
7	2	2.Открытость архитектуры		8,018
8	10	13. Построена на транзисторах		6,832
9	4	5.Wi-Fi доступ к сети		4,754
10	9	11. Наличие сетевой ОС		3,221

## 4. Добавление к БЗ ИПС новых объектов

Эта форма работы полезна тем, что учащиеся, добавляя новый объект, должны увидеть, достаточно ли существующих в БЗ признаков для идентификации нового объекта путем переобучения ИПС, или же необходимо дополнить этот список новой(ми) переменными.

### 4.1. Экспертная система

В ЭОС, как было показано на слайде 29, вставка нового исхода осуществляется с помощью модуля 8 «Удаление и добавление новых переменных и исходов». Новый(е) объект(ы) рекомендуется включать в БЗ знаний уже отлаженной экспертной системы. После вставки нового исхода следует с помощью п. 4 обучить систему его узнаванию. Здесь возможны два варианта развития событий:

- I. В БЗ достаточно переменных для узнавания нового объекта и тогда после обучения результаты записываются с помощью п. 10.
- II. Попытка обучения не приводит к успеху и тогда следует добавить в БЗ новую переменную(ые). Ввод новой переменной влечет за собой переобучение и тестирование всей системы целиком.

## 4.2. Нейронная сеть

Добавление нового объекта (выхода) в НС может быть осуществлено двумя способами:

### *I. Средствами эмулятора Deductor*

Загружается проект с нужной ЭС, с помощью имеющихся в нем средств модифицируется исходная таблица и строится целиком заново новая НС.

### *II. Средствами текстового редактора*

Таблица данных редактируется текстовым редактором, импортируется в Deductor и на ее основе создается новая НС с выполнением всех этапов этой работы.

Работоспособность расширенной НС сначала проверяется визуализатором «Таблица сопряженности», а потом тестируется визуализатором «Что-если».

# Методические рекомендации

Расширение ИПС за счет ввода в него новых объектов существенно повышает значимость (практическую полезность) этих средств для решения вопроса систематизации и контроля знаний в различных предметных областях. Эта формы работы вырабатывает у учащихся целостный, систематичный взгляд на определенную тему учебного курса. Подбор характерных признаков (переменных, атрибутов) для добавляемого объекта (исхода) позволяет лучше понять, какое место в ряду уже имеющихся объектов БЗ занимает добавляемый в нее объект. Что, несомненно, способствует выработке у учащихся навыков проведения систематизации объектов. Данная форма работы расширения готовой БЗ является подготовительным этапом к созданию нового ИПС.

## 5. Создание ИПС

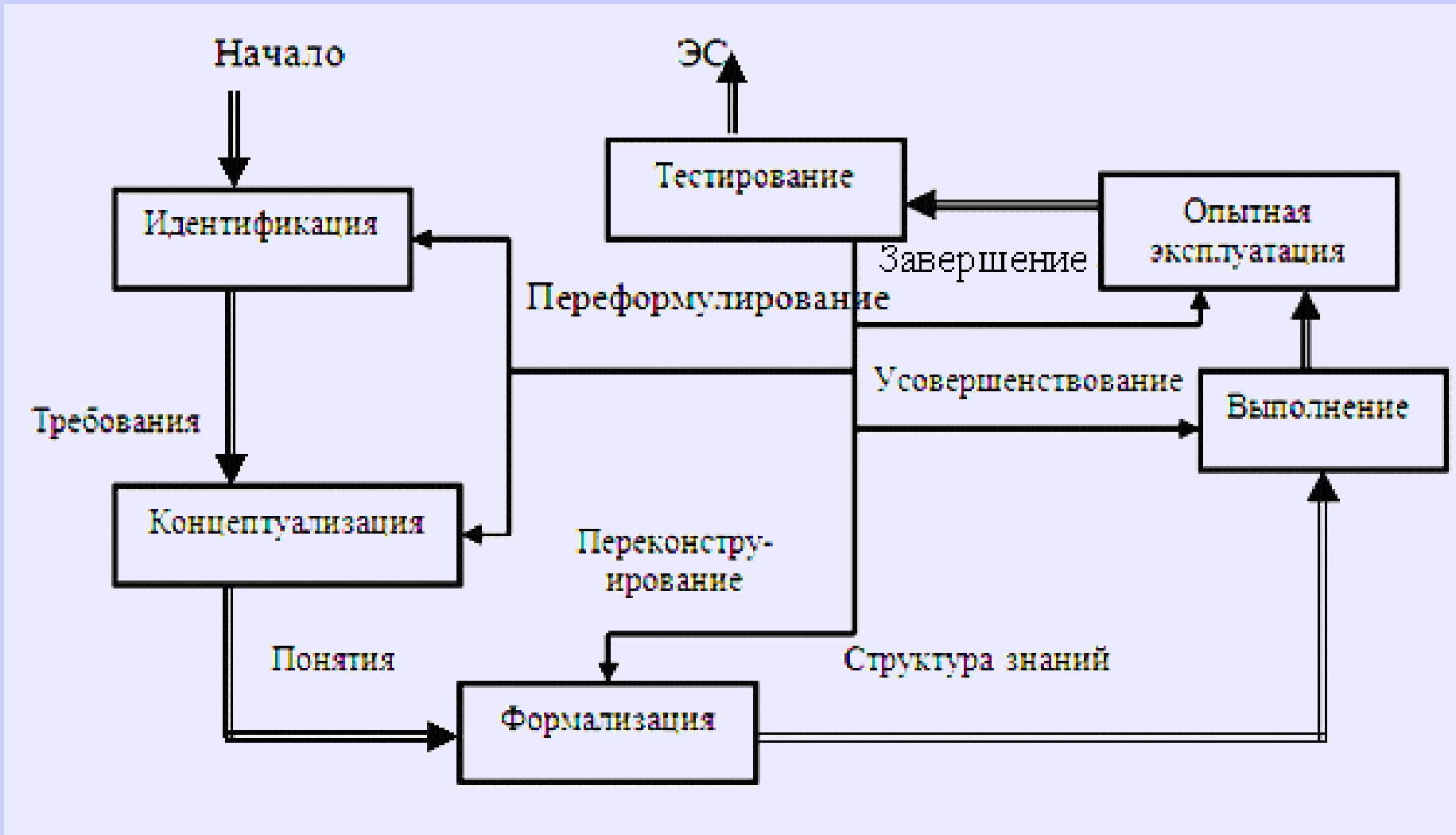
Эта форма работы с ИПС является завершающим этапом формирования навыков работы учащихся с ними. В этом случае учащийся должен предварительно продумать список всех переменных и исходов, а затем ввести их в БЗ ИПС и произвести его обучение.

Создание ИПС начинается с составления таблицы (например, в MS Excel), в столбцах которой перечисляется список всех атрибутов-переменных (входные данные), а в строках – перечень объектов для распознавания (выходы-исходы). Создание таблиц для ЭС и НС существенно отличается друг от друга. Для НС создается одна таблица, а для ЭОС число таблиц равно количеству ее узлов. В ЭОС эти таблицы создают единое ИПС, а в НС каждой таблице соответствует одна самостоятельная нейронная сеть. Правда, для НС можно создать одну большую таблицу, но тогда в ней окажется много лишних атрибутов.

Этапы создания ИПС выглядят следующим образом:

## 5.1. Экспертная система

### Этапы создания ЭОС





## 5.1.1. Создание таблиц переменных и исходов

### База знаний «Классификация ЭВМ»

#### УЗЕЛ 1

#### Переменные:

1. Многопроцессорная ЭВМ

2. Многопользовательский режим работы

3. Блочная конструкция

4. Общедоступность

5. Работа с большими БД

6. Поддерживает САПР

7. Мобильность

8. Сервер

9. Многотерминальность

10. Первая зарубежная ЭВМ

11. Первая отечественная ЭВМ

12. Двухпроцессорная архитектура

13. Синтез ПК и телефона

# База знаний «Классификация ЭВМ»

## УЗЕЛ 1

Исходы:



СуперЭВМ



Большие ЭВМ



МиниЭВМ



МикроЭВМ



Компьютерная  
телефония



# База знаний «Классификация ЭВМ»

## УЗЕЛ 2

### Переменные:

1. 1 поколение	11. АЦПУ	21. М-многоуровневый процессор
2. 2 поколение	12. Алфавитно-цифровой дисплей	22. Первичный ПК
3. 3 поколение	13. Магнитная дискета	23. Штучное производство
4. 4 поколение	14. Оптический диск	24. Первая составная ЭВМ 3-го поколения
5. Электронные лампы	15. Трансляторы	25. Wi-Fi доступ к сети
6. Транзисторы	16. Пакетный режим работы	26. Лазерный проектор
7. Интегральные схемы	17. ОС	27. Карманная версия ПК
8. БИС	18. Сетевая ОС	28. Матричный принтер
9. Пульт управления	19. Отечественная ЭВМ	29. Предмостаренный ПК
10. Клавиатура	20. Открытая архитектура	30. Графический дисплей

# База знаний «Классификация ЭВМ»

## УЗЕЛ 2

### Исходы:

1. ENIAC

2. МЭСМ

3. IBM-360

4. БЭСМ

5. IBM-370

6. ЕС ЭВМ

7. CRAY

8.  
Ломоносов

9. PDP-11

10. СМ  
ЭВМ

11. Наири

12. РС IBM

13. APPLE

14.  
Смартфон

## 5.1.2. Инициализация системы

Здесь разработчик задает число узлов, максимальное число переменных и исходов для всех узлов, а также вводит их в БЗ. Это делается модулем 1 оболочки ЭС так, как показано на рисунке.

```
Введите название ЭОС: КЛАССИФИКАЦИЯ ПО ЭВМ

      ЭКСПЕРТНАЯ СИСТЕМА КЛАССИФИКАЦИЯ ПО ЭВМ

Для установки экспертной системы ответьте на следующие вопросы:

Сколько узлов имеет система? - 2

Каково максимальное число переменных (свойств объектов ЭС)
на входе каждого из узлов? - 20

Каково максимальное число исходов (объектов ЭС) на выходе
любого из узлов? - 20

Узел 1
Сколько переменных на входе этого узла? - 5

Назовите эти переменные:
Переменная 1 - СОЗДАЕТ ИСПОЛНЯЕМЫЕ ФАЙЛЫ
Переменная 2 - СОЗДАЕТ ПРОГРАММЫ
Переменная 3 - ОСУЩЕСТВЛЯЕТ ТРАНСЛЯЦИЮ
Переменная 4 - УПРАВЛЯЕТ РАБОТОЙ ЭВМ
Переменная 5 - ГРАФИЧЕСКИЙ РЕЖИМ РАБОТЫ

Узел 1
Сколько возможных исходов на выходе этого узла? - 3

Назовите эти исходы:
Исход 1 - БАЗОВОЕ ПО
Исход 2 - СИСТЕМА ПРОГРАММИРОВАНИЯ
Исход 3 - ПРИКЛАДНОЕ ПО

Узел 2
Сколько переменных на входе этого узла? - 18
```

### *5.1.3. Ввод примеров*

Когда все переменные и исходы введены, то приступают к обучению ЭОС через ввод примеров (слайд 22).

### *5.1.4. Тренировка системы*

На основании введенных примеров вычисляется массив правил для работы системы как эксперта по распознаванию объектов (слайд 23).

### *5.1.5. Запоминание текущего состояния системы*

Для сохранения массива переменных, исходов, их числа, а также массива правил используется п. 6.

### *5.1.6. Обучение системы*

После ввода примеров и вычисления массива правил система чаще всего работает не надежно, поэтому обучаемому рекомендуется переобучить систему, вводя дополнительные примеры. По завершению сеанса обучения результаты сохраняются (п. 6).

# Методические рекомендации

Процесс создания ЭОС завершается ее тестированием и минимизацией числа переменных (слайды 29-30). Как показывает опыт, после обучения системы, которое, как правило, идет в порядке следования исходов, необходимо проверить правильность ее работы путем произвольного выбора исходов для их узнавания. В случае возникновения ошибки систему надо переобучить, то есть подкорректировать массив правил. Затем следует удалить из базы знаний лишние переменные (слайд 30). Удаление переменных может привести к сбою системы, поэтому необходимо повторное ее тестирование. По завершению этой работы ИПС готово для практического использования в качестве «интеллектуального» средства тестирования знаний.

## 5.2. Нейронная сеть

В режиме мастеров обработки Deductor позволяет сконструировать НС с заданной структурой, определить ее параметры и обучить с помощью одного из доступных в системе алгоритмов обучения.

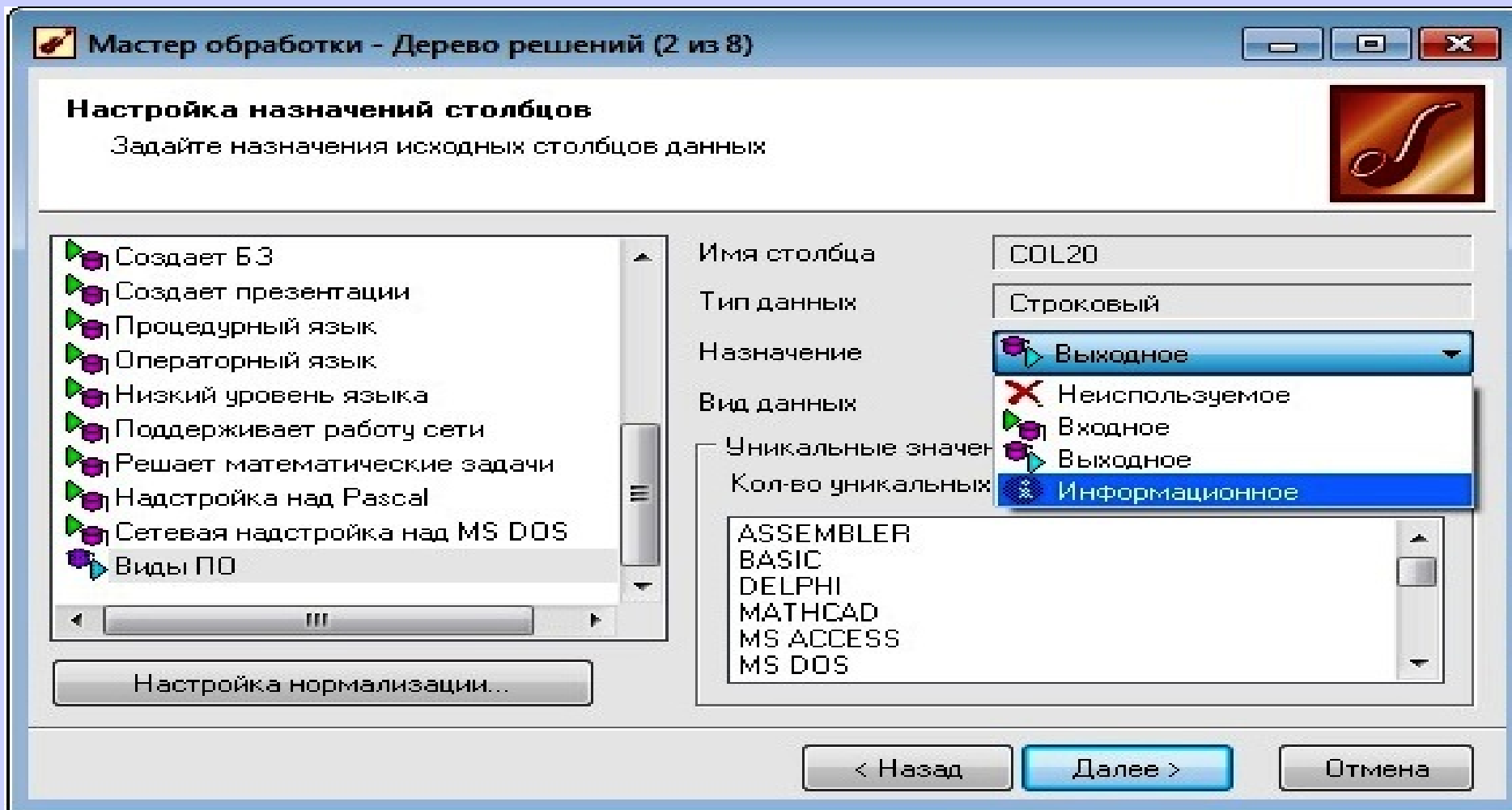
Проиллюстрируем работу по созданию ИПС на примере создания нейросети **«Виды ПО ЭВМ»**.

Первоначально создается таблица, в столбцах которой перечисляется список всех атрибутов (входные данные), а в строках – перечень объектов распознавания (выходные данные), которая с помощью редактора «Блокнот» импортируется в Deductor. В качестве входного параметра выбираются признаки объектов, а выходного – узнаваемый объект. Построенная при этом таблица может быть обработана с помощью различных мастеров обработки, из которых, на наш взгляд, для решения рассматриваемой задачи классификации объектов, эффективными являются **Дерево решений** и **Кластеризация**.



## 5.2.1. Дерево решений

Задаются назначения столбцов данных. Входными значениями делаются признаки объектов, выходными – столбец самих объектов.



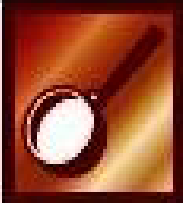
Запускается процесс построения нейронной сети, нажатием кнопки *Пуск*. 100% означает, что *дерево успешно построено* и все объекты распознаны.

The screenshot shows a software window titled "Мастер обработки - Дерево решений (6 из 8)". The main heading is "Построение дерева решений" with the instruction "Запустите процесс построения дерева решений". The interface is divided into several sections:

- Распределено, шт.:** A table with two rows: "Распознано" (91) and "Нераспознано" (0).
- Распознано, %:** A table with two rows: "Обучающее мн-во" (100,00) and "Тестовое мн-во" (100,00).
- Summary statistics:** "Кол-во узлов" (31), "Кол-во правил" (16), "Время обучения" (00:00:00), and "Темп обновления" (100).
- Controls:** Three buttons: "Пуск" (green play icon), "Пауза" (grey pause icon), and "Стоп" (grey stop icon).
- Progress bar:** A blue bar at the bottom showing 100% completion.
- Chart:** A coordinate system with a vertical axis from 0 to 80 and a horizontal axis with a value of 16. A vertical dashed line is drawn at x=16.

Способы отображения (визуализация) данных: *дерево решений, правила вывода, значимость атрибутов, таблица сопряженности и инструмент «Что-если»*

**Определение способов отображения**  
Выберите способ отображения данных



Название	Описание
<input checked="" type="checkbox"/> <b>Data Mining</b>	
<input checked="" type="checkbox"/> <b>Дерево решений</b>	Отображение дерева решений в виде дерева
<input checked="" type="checkbox"/> <b>Правила</b>	Отображение списка правил дерева решений
<input checked="" type="checkbox"/> <b>Значимость атрибутов</b>	Отображение значимости атрибутов
<input checked="" type="checkbox"/> <b>Таблица сопряженности</b>	Результаты построения модели в виде таблицы сопряженности
<input checked="" type="checkbox"/> <b>Что-если</b>	Анализ построенной модели по принципу что-если
<input type="checkbox"/> <b>Табличные данные</b>	

Выбор способа отображения данных нейронной сети

# Работа визуализатора «Значимость атрибутов» при проведении классификации объектов

Deductor Studio Academic

Файл Правка Вид Избранное Сервис Окно ?

Дерево решений X | Правила X | Значимость атрибутов X | Таблица сопряженности X | Что-если X

Целевой атрибут: Виды ПО

№	Номер	Атрибут	Значимость, %	/
1	7	Производит трансляцию		19,709
2	2	Загружается автоматически		16,434
3	4	Графический интерфейс		9,947
4	11	Создает БЗ		8,782
5	6	Создает программы		8,709
6	8	Создает тексты		7,006
7	12	Создает презентации		6,582
8	19	Сетевая надстройка над MS DOS		5,360
9	1	Создает папки		4,550
10	16	Поддерживает работу сети		3,303
11	9	Создает БД		3,303
12	14	Операторный язык		3,303
13	18	Надстройка над Pascal		3,010
14	17	Решает математические задачи		0,000
15	15	Низкий уровень языка		0,000
16	5	Создает исполняемые файлы		0,000
17	3	Многозадачный режим работы		0,000
18	13	Процедурный язык		0,000
19	10	Создает графику		0,000

# Работа визуализатора «Что-если» по распознаванию объектов с помощью их свойств (атрибутов)

Поле	Значение
<b>Входные</b>	
ab Создает папки	нет
ab Загружается автоматически	нет
ab Многозадачный режим работы	нет
ab Графический интерфейс	да
ab Создает исполняемые файлы	да
ab Создает программы	да
ab Производит трансляцию	да
ab Создает тексты	нет
ab Создает БД	нет
ab Создает графику	нет
ab Создает БЗ	нет
ab Создает презентации	нет
ab Процедурный язык	да
ab Операторный язык	да
ab Низкий уровень языка	да
ab Поддерживает работу сети	нет
ab Решает математические задачи	нет
ab Надстройка над Pascal	да
ab Сетевая надстройка над MS DOS	нет
<b>Выходные</b>	
ab Виды ПО	DELPHI

## 5.2.2. Кластеризация

При кластеризации изменяется подход для построения НС. Сеть строится так, что атрибуты становятся выходными значениями, а выходной параметр – входным.

Column	Property	Value
13.Процедурный язык	Имя столбца	COL3
14.Операторный язык	Тип данных	Строковый
15.Низкий уровень абстракции	Назначение	Выходное
16.Поддерживает	Вид данных	Дискретный
17.Решает математические задачи	Уникальные значения	Кол-во уникальных значений: 2
18.Надстройка на базе		
19.Сетевая надстройка		
ВИДЫ ПО		

Назначение столбцов при кластеризации


# Работа визуализатора «Что-если» мастера *Кластеризация* с обученной нейронной сетью

Входные	Выходные	DELPHI
ab Виды ПО		
	ab Создает папки	нет
	ab Загружается автоматически	нет
	ab Организует вычисления по формулам	нет
	ab Графический интерфейс	да
	ab Создает исполняемые файлы	да
	ab Создает программы	да
	ab Производит трансляцию	да
	ab Создает тексты	нет
	ab Создает БД	нет
	ab Создает графику	нет
	ab Создает БЗ	нет
	ab Создает презентации	нет
	ab Процедурный язык	да
	ab Операторный язык	да
	ab Низкий уровень языка	нет
	ab Поддерживает работу сети	нет
	ab Решает математические задачи	нет
	ab Надстройка над Pascal	да
	ab Сетевая надстройка над MS DOS	нет


## Сравнительная характеристика ЭОС и НС

Рассмотренные два способа (с помощью экспертных систем и нейронных сетей) решения задач идентификации объектов, несмотря на их внешнее различие, имеют много общего. Оба они предполагают составление списков входных данных (переменных) и выходных (исходов) данных. У обоих способов создается база знаний (система правил), базирующаяся на весовых коэффициентах. В обоих случаях осуществляется этап обучения, на котором и проявляются существенные отличия. Так, обучение нейронной сети производится нейроэмулятором Deductor Academic после ввода обучающей выборки. В экспертных же системах, после ввода примеров и получения системы правил, чаще всего, приходится проводить ее дополнительное обучение.





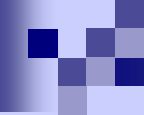
Представленные программные среды (оболочка ЭОС и Deductor Academic) эффективно использовать совместно с целью оптимизации числа атрибутов (характерных признаков) распознаваемых объектов. Действительно, работая в среде ЭОС довольно трудно определить значимые и незначимые атрибуты (переменные). А инструмент дедуктора «Значимость атрибутов» позволяет легко определить лишние атрибуты (свойства объектов). При этом надо учитывать то обстоятельство, что ЭОС и нейросети ведут себя по-разному. Так, нейросеть может идентифицировать объект по отсутствию у него указанных свойств. И это логично, так как отсутствие у объекта какого-либо свойства есть его характеристика. А для ЭОС идентификация объекта обязательно требует наличия у него хотя бы одного характерного признака, т.к. такова особенность работы алгоритма, заложенного в ее оболочку.



Существует еще одно характерное отличие ЭОС от нейронной сети. Как уже отмечалось, экспертная система может быть многоузловой, что обуславливается многоуровневой классификацией объектов. В рассматриваемом нами примере «Виды ПО ЭВМ» идентифицируемая компьютерная программа сначала в первом узле соотносится по своим признакам с системным, прикладным ПО или системой программирования, а во втором - по переменным этого узла определяется название программы. Нейросеть, по определению, не имеет узлов, в ней задается только один список атрибутов, позволяющих сразу определить название узнаваемой программы. Этот список атрибутов может совпадать со списком переменных второго узла ЭОС, таким образом, нейросеть можно формально считать одноузловой экспертной системой.

## **Методические рекомендации по использованию ИПС в системе образования**

ИПС, основанные на знаниях, вполне вписываются в компьютерные системы обучения в качестве инструмента систематизации и контроля знаний в различных предметных областях, в том числе и «Информатика». Они являются средством представления знаний и организации диалога пользователя с системой, обеспечивая при этом выработку стратегии и тактики решения задач изучаемой предметной области, а также осуществление контроля уровня знаний.



Представленные программные среды можно использовать и для разработки компьютерных тестов, которые, в отличие от традиционных, призваны «интеллектуализировать» процесс тестирования. Узнавание объекта зависит от уровня знаний учащихся, так как экспертная система задает следующий вопрос, исходя из ответа на предыдущий, а в нейронной сети узнавание объекта можно осуществить и по одному (уникальному) атрибуту.

# Практика использования ИПС

Представленные материалы исследования вопросов использования интеллектуальных средств для систематизации и контроля знаний в области информатики нашли практическое применение в работе кафедры математики и информатики Череповецкого государственного университета.

*Работа велась в двух направлениях:*

# 1. Освоение технологии создания ИПС

Этот процесс проходил в рамках преподавания двух учебных дисциплин на 3 курсе:


- «Нейроинформатика» (18/8/14);
- «Системы искусственного интеллекта» (18/10/14).

Студенты специальности «Прикладная математика и информатика» (2015-2018 годы) (47 чел.) изучали эти дисциплины, на практических занятиях проводили знакомство с описанными в отчете средами, а на лабораторных работах создавали экспертные обучающие системы и нейронные сети для решения задач в различных предметных областях.

## 2. Тестирование знаний

Это направление использования ИПС осуществлялось в процессе тестирования студентов 1 курса (2016-2018 годы) специальностей:

- «Прикладная математика и информатика» (дневное отделение – 68 чел. (3 группы), дисциплина «Основы информатики» (20/10/10));
- инженерных специальностей (заочное отделение – 220 чел. (10 групп), дисциплина «Информатика» (6/12)).



При тестировании использовались как традиционные тесты, так и тесты на основе ЭОС и НС. Практика показала, что тесты на основе ЭОС и НС воспринимается учащимися с большим интересом и качество проверки знаний в этом случае является более объективным, так как нет возможности случайного узнавания правильного ответа (для его получения не достаточно выбрать один ответ из предлагаемых вариантов). При работе с интеллектуальной системой для узнавания объекта нужно ответить не на один, а на несколько вопросов. Отвечая на них, учащийся получает более полную характеристику того или иного объекта.



## Публикации:

1. Касторнова В.А., Касторнов А.Ф. Технология компонентного (модульного) проектирования экспертной системы для систематизации и контроля знаний // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): материалы III Междунар. научн.-техн. конф. (Минск, 21-23 февраля 2013 г.) / редкол.: В.В. Голенков (отв. ред.) [и др.]. - Минск: БГУИР, 2013. - 592 с. С.461-466.
2. Касторнова В.А., Касторнов А.Ф. Технология проектирования обучающей экспертной системы для систематизации и контроля знаний // Экономика и управление: проблемы, решения. - Издательский дом "Научная библиотека". - № 3, 2013 С.58-65.
3. Касторнова В.А., Касторнов А.Ф. Модульное проектирование экспертной системы, применяемой для контроля знаний // Ученые записки. Вып. 47. – М.: ИИО РАО, 2013. С.80-94.
4. Касторнова В.А., Касторнов А.Ф. Проектирование экспертной системы для систематизации контроля знаний // Современные информационные технологии. Теория и практика Материалы I Всероссийской научно-практической конференции. Под ред. Е.А. Смирновой. Череповец: ЧГУ, 2015 С.127-133.
5. Касторнова В.А., Касторнов А.Ф., Андреев А.Е., Яламов Г.Ю. Применение и перспективы развития технологии модульного проектирования экспертных систем контроля знаний в сфере образования // Управление образованием: теория и практика. – 2016. № 1(21) С.52-63.
6. Касторнова В.А., Касторнов А.Ф. Интеллектуальное тестирование знаний // Информационные и педагогические технологии в современном образовательном учреждении: Материалы VIII Всерос. науч.-практ. конф. (г. Череповец, 21 апреля 2017 г.) / Отв. ред. Е.А. Смирнова. Череповец, 2017. 148 с. С.14-17.
7. Касторнова В.А., Касторнов А.Ф. Применение технологии модульного проектирования экспертных систем контроля знаний в сфере образования // Информатизация образования - 2017 : сборник материалов Международной научно-практической конференции (Чебоксары, 15 июня - 17 июня 2017 года) / отв. ред. Н. В. Софронова. - Чебоксары: Чуваш, гос. пед. ун-т, 2017. — 516 с. С.156-163.
8. Касторнова В.А., Касторнов А.Ф. Использование экспертных систем контроля знаний в сфере образования // Сборник трудов участников VII Всероссийской научно-практической конференции «Современное непрерывное образование и инновационное развитие» // Под редакцией профессора А.Н. Царькова и профессора И.А. Бугакова. – Серпухов: МОУ «ИИФ», 2017. - 1056 с. С.707-708.
9. Касторнова В.А., Касторнов А.Ф. Интеллектуальное тестирование знаний // Труды VI Всероссийской научно-методической интернет-конференция «Повышение квалификации педагогических кадров в изменяющемся образовании» (Москва, 7-8 декабря 2017 г.). С.1-5.

## Публикации:

10. Касторнова В.А. Использование интеллектуальных программных средств в осуществлении систематизации контроля знаний // Проблемы эффективности и безопасности функционирования сложных технических систем : сборник трудов XXXVII Всероссийской научно-технической конференции (г. Серпухов, 28-29 июня 2018 г.). Часть 6. / под общ. ред. Ю.В. Астапенко, Романенко Ю.А. - Серпухов: Изд-во Военной академии РВСН им. Петра Великого (филиал г. Серпухов Московской обл.), 2018. С.234-241.
11. Касторнова В.А. Интеллектуальные программные средства как инструмент осуществления систематизации и контроля знаний // Труды Международной научно-практической конференции «Информатизация образования – 2018». 11–12 сентября 2018 г., г. Москва. В 2 ч. Ч. 2. М.: Изд-во СГУ, 2018. С.141-149.
12. Касторнова В.А. Информационное моделирование на основе интеллектуальных систем как инструмент формирования компетенций в области использования информационных ресурсов // Электронные ресурсы в непрерывном образовании : труды VII Международного научно-методического симпозиума «ЭРНО-2018» (Геленджик, 24–27 сентября, 2018 г.) ; Южный федеральный университет / [редкол.: В. И. Мареев и др.]. – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2018. С. 27-31.
13. Касторнова В.А. Системы искусственного интеллекта как технологическая основа решения неформализованных задач в информационно-образовательном пространстве предметной области «Информатика» // Мир науки, культуры, образования. – 2018. - № (73) С.17-20.
14. Касторнова В.А. Системы искусственного интеллекта как технологическая основа решения неформализованных задач в информационно-образовательном пространстве предметной области «Информатика» // Педагогическая информатика. – 2018. - № 4 С.47-58.
15. Касторнова В.А. Технологии искусственного интеллекта как средство систематизации и контроля знаний в предметной области «Информатика» // Постсоветское пространство – территория инноваций. 5-я Международная научно-практическая конференция: доклады и сообщения. 24-25 декабря 2018 г. Москва. Под общей редакцией проф. Б.К. Тебиева. – М.: Московский региональный социально-экономический институт, 2018. – 129 с. С.79-83.
16. Касторнова В.А., Касторнов А.Ф. Информационное моделирование на основе интеллектуальных систем как инструмент систематизации знаний в предметной области «Информатика» // Вестник ЧГУ. – 2019. – № 1(88). С.175-187.
17. Касторнова В.А. Методические особенности использования интеллектуальных программных средств для систематизации и контроля знаний // Педагогическая информатика. – 2019. – № 1. С.92-102.
18. Касторнова В.А., Касторнов А.Ф. Технология информационного моделирования с помощью систем искусственного интеллекта как инструмент систематизации знаний // Теория и практика информатизации образования: внедрение результатов и перспективы развития: Сборник научных трудов юбилейной Международной научно-практической конференции, посвященной 35-летию становления информатизации отечественного образования (г. Москва, 19 декабря 2019 г.). / Под общ. ред. И.В. Роберт. М.: Изд-во АЭО, 2020 С.314-325.



***Спасибо за внимание!***

**kastornova\_vasya@mail.ru**