Министерство науки и образование Российской Федерации
ФГБОУ ВО «Уральский государственный педагогический университет»
Институт физики, технологии и экономики
Кафедра физики и математического моделирования

РАЗРАБОТКА И ВНЕДРЕНИЕ ЦИФРОВОГО КОНТЕНТА ДЛЯ ДИСЦИПЛИНЫ “ОСНОВЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА”

Выпускная квалификационная работа
по направлению 09.03.03 «Прикладная информатика»
профиль «Прикладная информатика в образовании»

Квалификационная работа
допущена к защите

Зав. кафедрой физики и
математического моделирования

_____ Сидоров В.Е

д. ф-м. наук, профессор

«____» _____2017 г.

Исполнитель:
Долженко Андрей Алексеевич,
обучающийся БИ-41 группы

_____
подпись

Научный руководитель:
Минина Елена Евгеньевна,
Доцент кафедры Физики и
Математического
моделирования, к.п.н.

_____
подпись

Екатеринбург 2017

Содержание

Введение 3

Глава 1. Описание предметной области исследования 5

1.1 Понятие ”нечетких множеств”, операции над нечеткими множествами 5

1.2 Средства Delphi для описания нечетких множеств 19

Глава 2. Структура и содержание цифрового контента по теме “нечеткие множества” 26

2.1 Разработка структуры цифрового контента 26

2.2 Реализация приложения “нечеткие множества” в среде Delphi 30

2.3 Разработка блога по теме: «Нечеткие множества» 37

2.4 Разработка вебинара по теме: «Нечеткие множества» 42

Заключение 47

Список литературы 49

Приложение 53

Введение

Умозаключения, прогнозирования либо человеческое мышление являются по своей природе далеко не четкими понятиями и постоянно нуждаются в описании с использованием теории нечетких множеств, которую, в свое время, предложил американский математик Лотфи Заде.

Актуальность выбранной темы прежде всего объясняется задачами, которые возникают перед человеком в самых разных отраслях знаний и представляют собой слишком сложными и не однозначными для того, чтобы использовать для их решения стандартную бинарную логику и чётко определенные модели и алгоритмы.

Зачастую, содержащие в своей постановке нечеткость, задачи характеризуются существенным наличием неопределенностей.

Мощь и интуитивная простота нечеткой логики, как методологии разрешения проблем, обеспечивает гарантию успешного использования встроенных систем контроля и анализа информации. В этот же момент происходит подключение человеческой интуиции и опыта оператора.

Нечеткая логика, в отличии от традиционной математики, которая нуждается в точной и однозначной формулировки моделировании на каждом этапе закономерности, дает полностью другой уровень мышления, который, в свою очередь, используя минимальной набор закономерностей сопровождает творческий процесс моделирования на самом высок уровне абстракции.

Объект исследования: Раздел “Нечеткое множество” нечеткой логики.

Предмет исследования: Содержание цифрового контента.

Цель исследования: Разработать цифровой контент для раздела “нечеткие множества” нечеткой логики.

В соответствии с целью исследования были поставлены следующие задачи:

1. проанализировать учебную литературу по теме исследования;

2. раскрыть сущность понятия «Нечеткое множество»;

3. проанализировать различные логические операции над нечеткими множествами;

4. определить структуру и разработать цифровой контент для раздела “нечеткие множества” нечеткой логики;

5. выбрать среду разработки приложения “Основы искусственного интеллекта”;

6. разработать приложение для дисциплины “Основы искусственного интеллекта”;

7. разработать, записать и провести вебинар по теме “нечеткие множества” нечеткой логики;

8. разработать и опубликовать блог для раздела “нечеткие множества” нечеткой логики;

9. апробировать цифровой контент при проведении занятий по дисциплине “Основы искусственного интеллекта”;

Нечеткие числа, которые получаются в результате "не полностью точных измерений", во многом схожи с распределениями теории вероятностей, но свободны от присущих последним недостатков: небольшое количество пригодных к анализу функций распределения, необходимость их принудительной нормализации, соблюдение требований аддитивности, трудность обоснования адекватности математической абстракции для описания поведения фактических величин. В пределе, при увеличении точности, нечеткая логика приходит к стандартной, Булевой. По сравнению с вероятностным методом, нечеткий метод позволяет быстро сократить объем производимых вычислений, что, в свою очередь, приводит увеличивает быстродействия нечетких систем.

Глава 1. Описание предметной области исследования

1.1 Понятие ”нечетких множеств”, операции над нечеткими множествами

Классическая логика по определению не может оперировать с нечетко определенными понятиями, так как все высказывания в формальных логических системах имеют только два взаимоисключающих состояния: «истина», со значением истинности «1», и «ложь», со значением истинности «0». Трехзначная логика, с дополнительным состоянием «возможно» со значением истинности «0,5», является одной из попыток уйти от традиционной двузначной бинарной логики Яном Лукасевичем [18]. Внедряя в рассмотрение нечеткое множество, Заде предложил обобщить традиционную двузначную логику на основе рассмотрения бесконечного множества значений истинности. В предложенном Лютфи Заде варианте нечеткой логики огромное количество значений истинности выражений обобщается до промежутка [0;1] , то есть включает как частные случаи традиционную бинарную логику и трехзначную логику Яна Лукасевича[18]. Таковой подход разрешает рассматривать выражения с разными значениями истинности и выполнять рассуждения с неопределенностью.

Нечеткое высказывание – это завершенная мысль, о истинности либо ложности которой возможно судить исключительно с некой степенью уверенности [0;1]: «может быть истинно», «может быть неверно» и т.д. Чем выше уверенность в истинности выражения, тем ближе значение степени истинности к 1. В предельных вариантах 0, в случае если абсолютная уверенность в ложности выражения, и 1, если полная уверенность в истинности выражения, соответствует традиционной бинарной логике В нечеткой логике нечеткие высказывания обозначаются так же, как и нечеткие множества: A, B, C… Введем нечеткое отображение $T:\Omega \rightarrow [0;1]$, которое

действует на множестве нечетких высказываний Ω=A, B, C… В этом случае значение истинности высказывания A∈Ω определяется как T:A ∈ [0;1] и является количественной оценкой нечеткости, неопределенности, содержащейся в высказывании A .

Нечёткое множество (время от времени размытое, расплывчатое, неясное)— понятие, которое использовал Лотфи Заде в 1965 году в заметке «FuzzySets» в журнале Informationand Control, в котором расширил одно из традиционных понятий множества, допустив, собственно характеристическая функция множества (названная Заде функцией принадлежности для нечеткого множества) имеет возможность принимать всевозможные значения в промежутке [0;1], но не лишь значения 0 либо 1. Считается базисным понятием нечёткой логики.

Нечеткое множество (fuzzyset) представляет из себя совокупность элементов случайной природы, сравнительно которых невозможно точно утверждать – владеют ли эти составляющие неким характеристическим свойством, которое используется для задания нечеткого множества.

Определение нечеткого множества. Пусть U – полное множество объектов некоторого класса (Универсум). Нечетким множеством A называется множество упорядоченных пар (x, μ_A), где x∈U, а $\mu_A(x)$ – функция принадлежности (степень принадлежности), которая ставит в соответствие каждому из элементов x∈U некоторое действительное число из отрезка [0;1], то есть данная функция определяется в форме отображения μA(x): E → [0;1].

Функция принадлежности может быть определена явным образом в виде функциональной зависимости (например, $\mu_A(x) = \exp\left(-\left(\frac{x-1}{2}\right)^2\right)$), либо дискретно (если множество U конечно) – путем задания конечной последовательности значений x∈{x_i} в виде: $\mu_A(x) = \{\mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \ldots + \mu_A(x_n)/x_n\}$.

Пустое нечеткое множество ∅ - это нечеткое множество с функцией принадлежности $\mu_{\varnothing}(x) = 0$.

Универсум - нечеткое множество с функцией принадлежности $\mu_U(x)= 1$.

Нечеткие числа – это нечеткие переменные, определенные на числовой оси (U=R).

Пример 1.1. Пусть универсальное множество X соответствует множеству возможных значений толщин изделия от 10 мм до 40 мм с дискретным шагом 1 мм. Нечеткое множество A, соответствующее нечеткому понятию «малая толщина изделия», может быть представлено в следующем виде:

A = 1 / 10 ; 0,9 / 11 ; 0,8 / 12 ; 0,7 / 13 ; 0,5 / 14 ; 0,3 / 15 ; 0,1 / 16 ; 0 / 17 ; … ; 0 / 40, или

A = 1 / 10 + 0,9 / 11 + 0,8 / 12 + 0,7 / 13 + 0,5 / 14 + 0,3 / 15 + 0,1 / 16 + 0 / 17 + … + 0 / 4 ,

где знак суммирования обозначает не операцию арифметического сложения, а объединения элементов в одно множество. Носителем нечеткого множества A будет конечное подмножество (дискретный носитель):

[SA] = 10; 11; 12; 13; 14; 15; 16.

Если же универсальное множество X является множеством действительных чисел от 10 до 40, т.е. толщина изделия может принимать все возможные значения в этих пределах, то носителем нечеткого множества A является отрезок [SA] = 10;16 .

Нечеткое множество с дискретным носителем может быть представлено в виде отдельных точек на плоскости, нечеткое множество с непрерывным носителем может быть представлено в виде кривой, что соответствует дискретной и непрерывной функциям принадлежности μ A x , заданным на универсальном множестве X.

На рисунке 1.1 представлен график функции принадлежности для отрезка [SA] в примере 1.1.

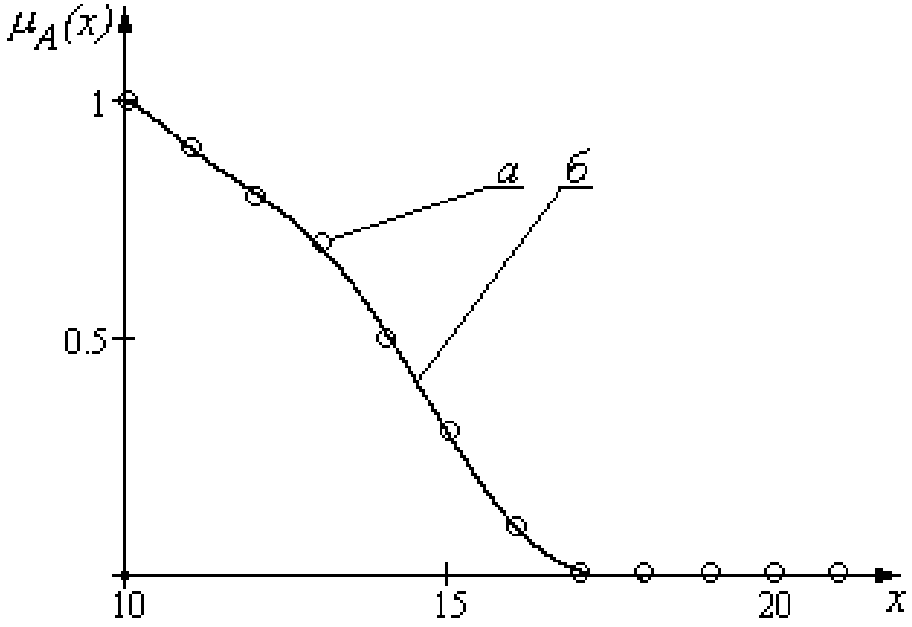

Рисунок № 1.1 - График функции принадлежности.

Нечеткое множество А называется конечным, если его носитель [SA] является конечным четким множеством. При этом, по аналогии с обычными множествами, можно говорить, что такое нечеткое множество имеет конечную мощность card A = card [SA]. Нечеткое множество А называется бесконечным, если его носитель [SA] не является конечным четким множеством. При этом счетным нечетким множеством будет называться нечеткое множество с счетным носителем, имеющим счетную мощность в обычном смысле в терминах теории четких множеств, т.е. если [SA] содержит бесконечное число элементов, которые однако можно пронумеровать натуральными числами 1, 2 ,3 . . . , причем достичь последнего элемента при нумерации принципиально невозможно. Несчетным нечетким множеством будет называться нечеткое множество с несчетным носителем, имеющим несчетную мощность универсума, т.е. если SA содержит бесконечное число элементов, которые невозможно пронумеровать натуральными числами 1, 2 ,3 . . .

На рисунке 1.2 Представлена возможная классификации нечеткого множества.

Рисунок № 1.2 - Классификация нечеткого множества

Пример 1.2. Нечеткое понятие «очень маленькое количество деталей» может быть представлено в виде конечного нечеткого множества A = 1 / 0 + 0,9 / 1 + 0,8 / 2 + 0,7 / 3 + 0,5 / 4 + 0,1 / 5 + 0 / 6 + … с мощностью card(A) = 6 и носителем [SA]= 0; 1; 2; 3; 4; 5, который является конечным четким множеством. Нечеткое понятие «очень большое количество деталей» может быть представлено в виде A = 0/0 + … + 0,1/ 10 + 0,4 11 + 0,7/12 + 0,9/13 + 1/14 + 1/15 + … + 1/n + … , n ∈ N – нечеткого множества с бесконечным счетным носителем [SA] ≡ N (множество натуральных чисел), который имеет счетную мощность в обычном смысле.

Пример 1.3. Несчетное нечеткое множество A, соответствующее нечеткому понятию «очень горячо», задано на универсальном множестве значений температур (в Кельвинах) температурой x ∈[0 ; ∞) и функцией принадлежности μ(A) = 1 − e − x , с носителем S(A) ≡ R + (множество неотрицательных действительных чисел), который имеет несчетную мощность континуума.

Величина sup x ∈ X (μ) A (x) называется высотой нечеткого множества.

Нечеткое множество A нормально, если его высота равна 1 , т.е. верхняя граница его функции принадлежности sup x ∈ X μ_A x = 1 . При sup x ∈ X μ_A<1 нечеткое множество называется субнормальным.

Нечеткое множество называется пустым, если ∀ x ∈ X μ_A x = 0 .

Непустое субнормальное множество всегда можно нормализовать, разделив все значения функции принадлежности на ее максимальное значение $\mu_A x \sup x \in X \mu_A x$.

Нечеткое множество называется унимодальным, если $\mu_A x = 1$ только для одной точки x (моды) универсального множества X .

Нечеткое множество называется точечным, если $\mu_A x > 0$ только для одной точки x универсального множества X .

Множеством α-уровня нечеткого множества A, определенного на универсальном множества X, называется четкое подмножество A(α) универсального множества X , определяемое в виде:

$A(\alpha) = x \in X | \mu_A x \geq \alpha$,где $\alpha \in [0;1]$.

Пример 1.4. A = 0,8/1 + 0,6/2 + 0,2/3 + 1/4 , A 0,5 = 1 ; 2 ; 4 , где A 0,5 – четкое множество, включающее те элементы x упорядоченных пар $\mu_A x / x$, составляющих нечеткое множество A, для которых значение функции принадлежности которых удовлетворяет условию $\mu A x \geq \alpha$.

Для множеств α -уровня выполняется следующее свойство: если $\alpha 1 \geq \alpha 2$, то мощность подмножества A(α1) не больше мощности подмножества A(α2) .

Элементы $x \in X$, для которых $\mu_A x = 0,5$ называются точками перехода нечеткого множества A .

Ядром нечеткого множества A, определенного на универсальном множестве X , называется четкое множество core A, элементы которого удовлетворяют условию $\text{core } A = x \in X | \mu_A x = 1$.

Границей нечеткого множества A, определенного на универсальном множестве X, называется четкое множество front A, элементы которого удовлетворяют условию $\text{front } A = x \in X | 0 < \mu_A x < 1$.

Пример 1.5. Пусть X = 0; 1; 2; … ; 10, M = [0;1] . Нечеткое множество можно определить на универсальном множестве натуральных чисел

следующим образом: несколько = 0,5 / 3 + 0,8/4 + 1/5 + 1/6 + 0,8/7 + 0,5/8; его характеристики: высота = 1 , носитель = 3; 4; 5; 6; 7; 8, точки перехода = 3; 8, ядро = 5; 6, граница = 3; 4; 7; 8.

Нечеткое множество A, определенное на универсальном множестве X , называется выпуклым, если $\mu A x \geq \min \mu A a ; \mu A b ; a < x < b ; x , a , b \in X$. На рисунке 1.3 представлен пример выпуклого и невыпуклого нечеткого множества.

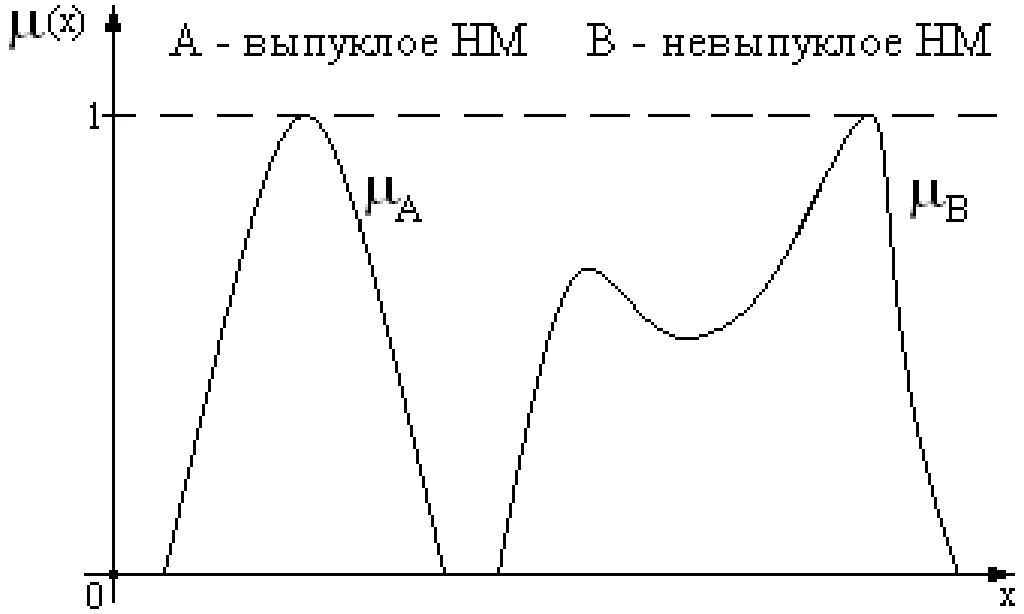

Рисунок № 1.3 - Выпуклое, невыпуклое нечеткое множество

2. *Операции над нечеткими множествами*.

Включение. Предоставим, что A и B – нечеткие множества на универсальном множестве U. Говорят, что B содержится в A (B подмножество A), если $\forall x \in U \mu_B(x) \leq \mu_A(x)$. Обозначение, операция включения, как: $A \subset B$.

На рисунке 1.4 страницы 12, представлен пример операции включения нечетких множеств.

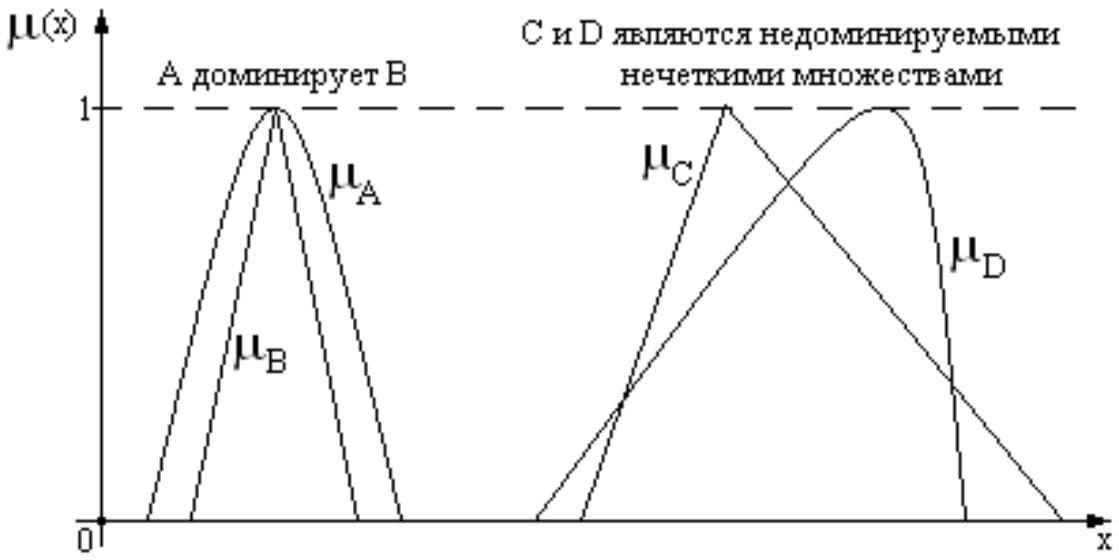

Рисунок № 1.4 - Включение нечетких множеств

Равенство. Представим, что A и B – нечеткие множества на универсальном множестве X. Нечеткие множества A и B равны, если $\forall x \in U \mu_A(x) = \mu_B(x)$.

На рисунке 1.5 представлен результат выполнения операции равенства нечетких множеств.

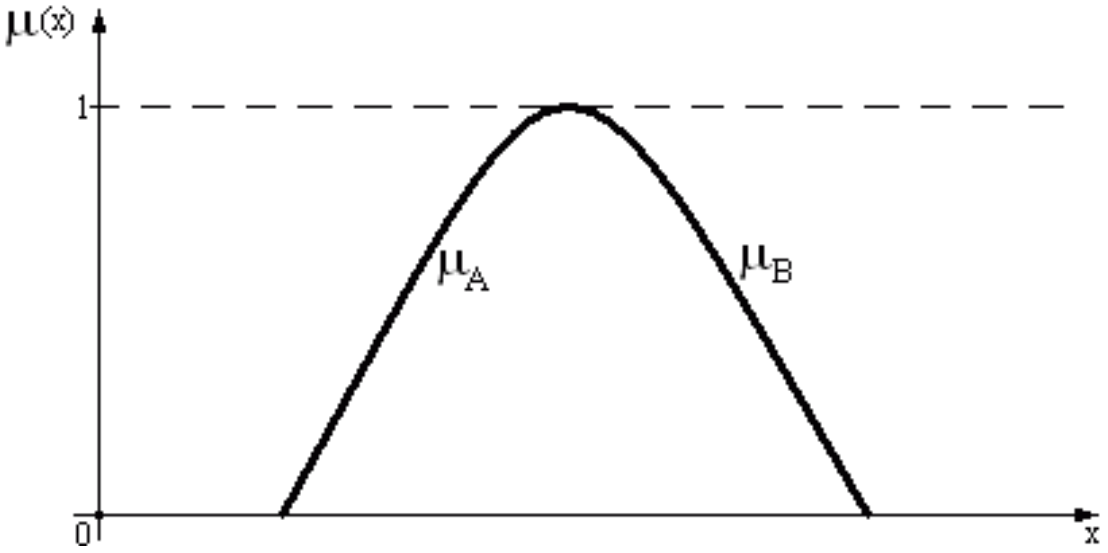

Рисунок № 1.5 - Равенство нечетких множеств

Операция дополнения: Представим, что A и B – нечеткие множества с множеством принадлежностей характеристических функций M = 0;1 , заданные на универсальном множестве X . Говорят, что A и B дополняют друг друга, т.е. A = notB или B =notA , если $\forall x \in X\ \mu_A x = 1 - \mu_B x$. Аналогичное определение функции дополнения - инверсия. Обозначается как: A= notA.

На рисунках 1.6 и 1.7 представлен результат выполнения операции дополнения нечетких множеств.

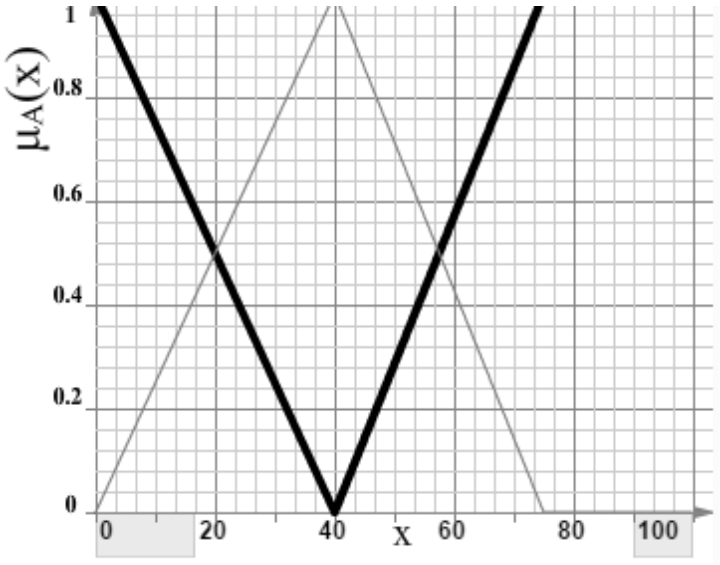

Рисунок № 1.6 - Дополнение нечетких множеств

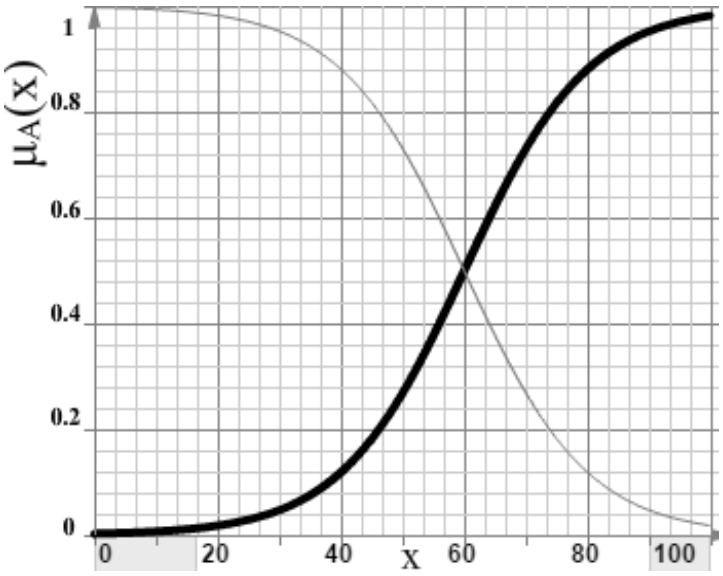

Рисунок № 1.7 - Дополнение нечетких множеств

Операция объединения. Представим, что A и B – нечеткие множества, заданные на U. Объединение нечетких множеств A и B , заданных на универсальном множестве X , – это наименьшее нечеткое множество A ∪ B , включающее как A , так и B с функцией принадлежности, заданной следующим образом: $\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_A(x)\} = \mu_A(x) \vee \mu_A(x)$. Обозначение операция объединения как: A∪B.

На рисунках 1.8 и 1.9 представлены результаты выполнения операции объединения нечетких множеств.

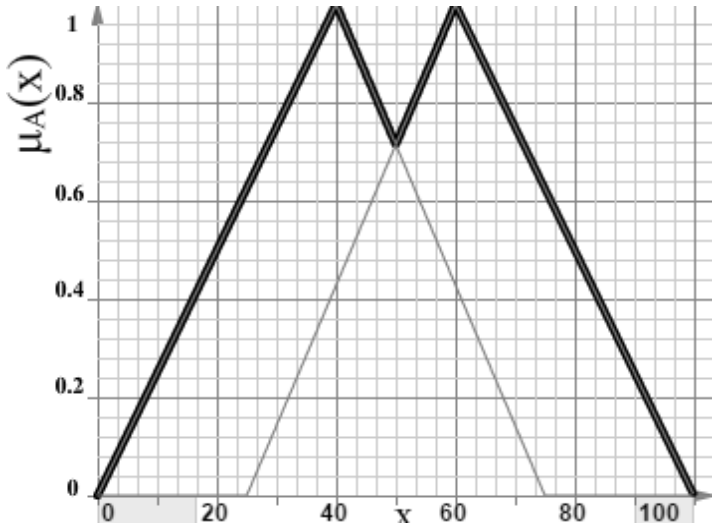

Рисунок № 1.8 - Объединение нечетких множеств

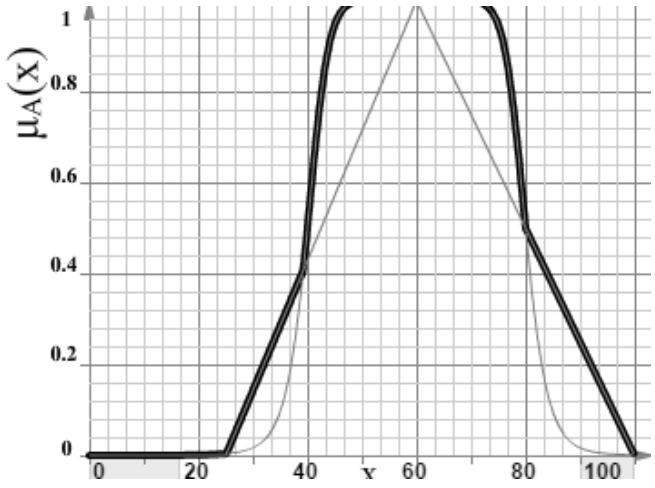

Рисунок № 1.9 - Объединение нечетких множеств

Операция пересечения. Представим, что A и B – нечеткие множества, заданные на U. Пересечение нечетких множеств A и B , заданных на универсальном множестве X , – это наибольшее нечеткое множество $A \cap B$, содержащееся одновременно и в A , и в B с функцией принадлежности, заданной следующим образом:$\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_A(x)\} = \mu_A(x) \wedge \mu_A(x)$. Операция пересечения обозначается как: $A \cap B$.

На рисунках 1.10 и 1.11 представлены результаты выполнения операции пересечения нечетких множеств.

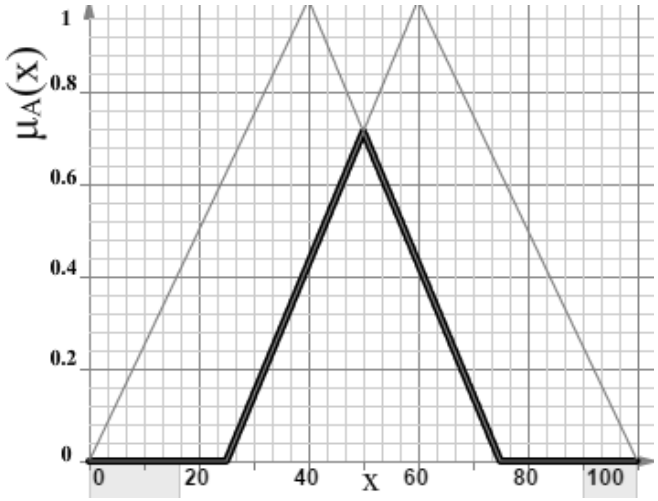

Рисунок № 1.10 - Пересечение нечетких множеств

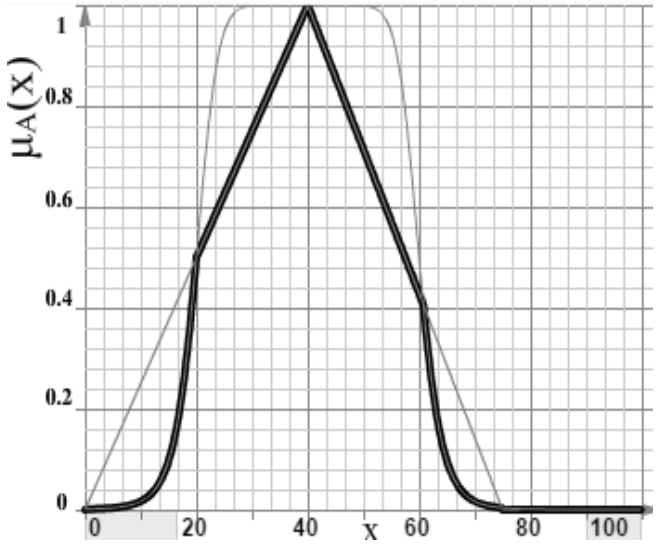

Рисунок № 1.11 - Пересечение нечетких множеств

Разность. Предоставим, что A и B – нечеткие множества, заданные на X. Разность нечетких множеств A и B, которые заданы на универсальном множестве X , – это нечеткое множество A \ B = A ∩notB с функцией принадлежности, заданной следующим образом:$\mu_{A-B}(x) = \min\{\mu_A(x), 1 - \mu_B(x)\}$. Обозначается разность нечетких множеств как A-B.

На рисунках 1.12 и 1.13 представлены результаты выполнения операции разности нечетких множеств.

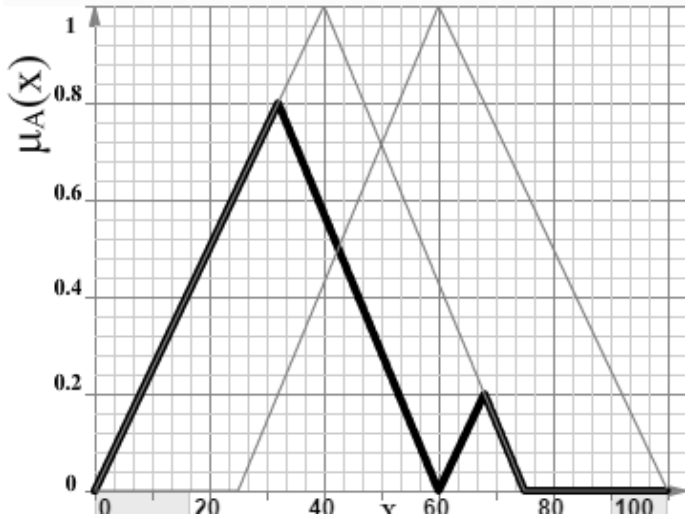

Рисунок № 1.12 - Разность нечетких множеств

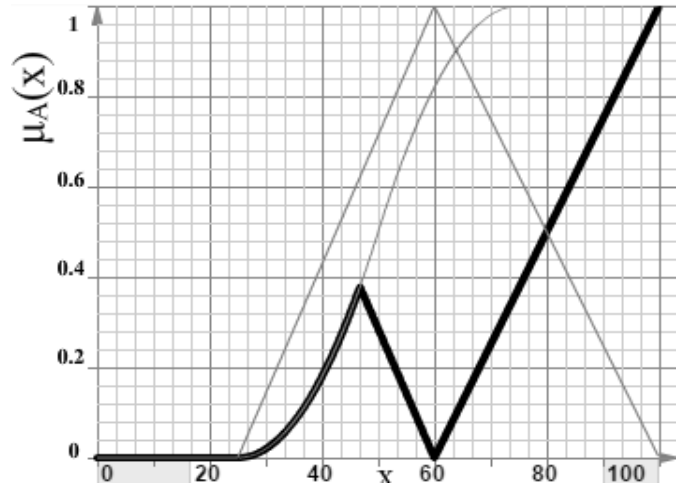

Рисунок № 1.13 - Разность нечетких множеств

Симметрическая разность. Представим, что A и B – нечеткие множества, заданные на U. Симметрической разностью A и B называется нечеткое множество $(A \cup B) \cap (\overline{A} \cup \overline{B})$ с функцией принадлежности $\mu_{A\Delta B}(x) = \min\{\max\{\mu_A(x), \mu_B(x)\}, \max\{1-\mu_A(x), 1-\mu_B(x)\}\}$. Обозначение симметрической разности: AΔB.

На рисунках 1.14 и 1.15 представлены результаты выполнения операции симметрической разности нечетких множеств.

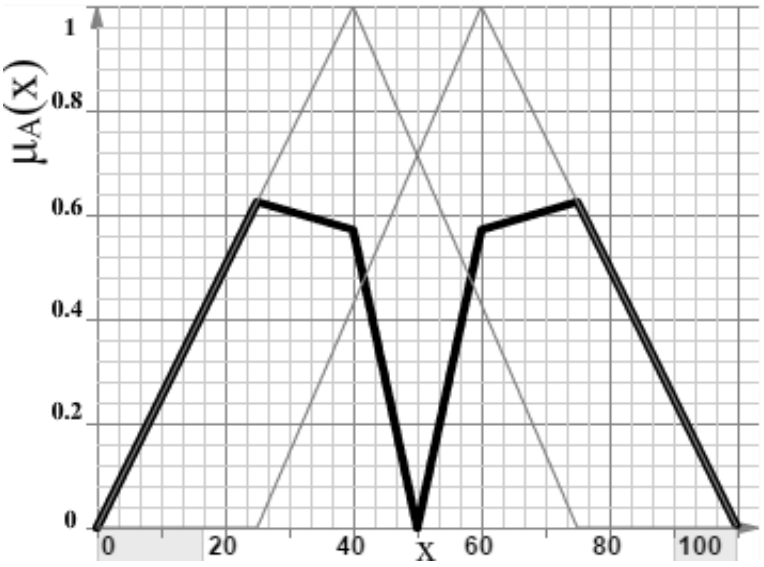

Рисунок № 1.14 - симметрическая разность нечетких множеств

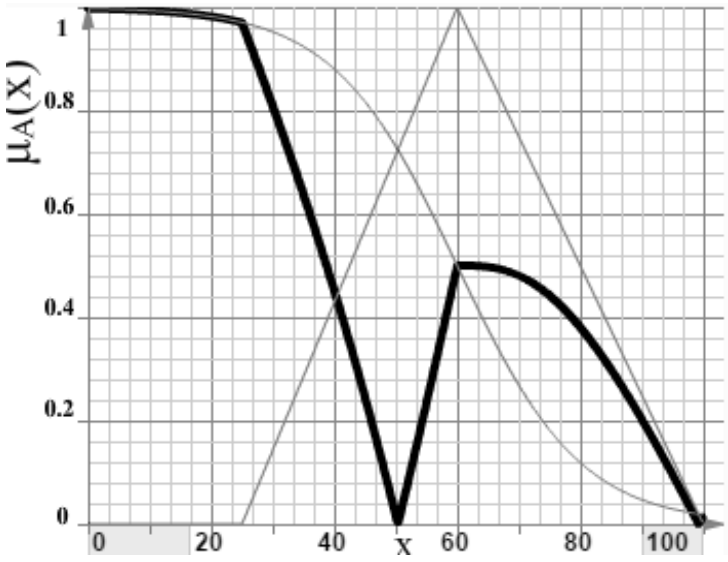

Рисунок № 1.15 - симметрическая разность нечетких множеств

Алгебраическое произведение. Представим, что A и B – нечеткие множества, заданные на U. Алгебраическим произведением A и B называется нечеткое множество с функцией принадлежности: $\forall x \in U \mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x)$. Обозначение алгебраического произведения: $A \cdot B$.

Алгебраическая сумма. Предоставим, что A и B – нечеткие множества, заданные на U. Алгебраической суммой A и B будет называться нечеткое множество с функцией принадлежности: $\forall x \in U \mu_{A \hat{+} B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$. Обозначение алгебраической суммы: $A \hat{+} B$.

Степень множества. Пусть A – нечеткое множество, заданное на U. Нечеткое множество A^{α} , где α - положительное число определяется функцией принадлежности: $\forall x \in U \mu_A{}^{\alpha}(x) = \mu^{\alpha}{}_A(x)$.

Декартово (прямое произведение). Пусть A_1, A_2, ….,A_n – нечеткие подмножества универсальных множеств U_1, U_2, ….,U_n соответственно. Декартово или прямое произведение $A = A_1 \times A_2 \times \ldots \times A_n$ является нечетким подмножеством множества $U = U_1 \times U_2 \times \ldots \times U_n$ с функцией принадлежности $\mu(x_1, x_2, ..., x_n) = \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), ..., \mu_{A_n}(x_n)\}$.

Ближайшее четкое к *A* множество $\underline{A}$: $\mu_{\underline{A}} = \begin{cases} 1, \mu_A > 0.5 \\ 0, \mu_A < 0.5 \\ 0, \mu_A = 0.5 \end{cases}$.

Изучение окружающего нас мира, проектирование новой техники и создание новых технологий невозможны без проведения разнообразных экспериментов. При этом далеко не всегда могут быть поставлены натурные эксперименты, зачастую они слишком дороги и требуют значительного времени, во многих случаях их проведение связано с риском и большими материальными или моральными издержками. В таких ситуациях предпочтительнее компьютерное моделирование, которое, однако, невозможно без использования математических моделей изучаемых объектов и процессов или проектируемых изделий.

Одним из главных требований к таким моделям является требование адекватности, то есть соответствия модели рассматриваемому явлению. Для многих технических систем и их элементов давно существуют весьма точные модели, которые зарекомендовали себя настолько хорошо, что зачастую удается провести процесс проектирования без обращения к натурному эксперименту. Этап испытаний изготовленных изделий (его можно рассматривать как натурный эксперимент) необходим в основном для выявления производственных дефектов.

Иначе обстоит дело со сложными системами, где человек играет активную роль. Здесь действует так называемый принцип несовместимости: для получения существенных выводов о поведении сложной системы необходимо отказаться от высоких стандартов точности и строгости, которые характерны для сравнительно простых систем, и привлекать к ее анализу подходы, которые являются приближенными по своей природе. Это все наглядно демонстрирует важность использования “нечеткого множества” в современном мире для принятия какого-либо решения.

1.2 Средства Delphi для описания нечетких множеств

Разработка прикладного программного обеспечения, или же приложений, Delphi производится во встроенной среде разработки IDE (Integrated Development Environment). IDE работает для организации взаимодействия с разработчиком программного обеспечения и включает в себя ряд окон, содержащих всевозможные управляющие составляющие. С поддержкой средств встроенной среды разработчик имеет возможность комфортно проектировать интерфейсную составляющую приложения, а также писать программный код и связывать его с управляющими элементами.

При этом вся работа по созданию приложения, охватывая отладку, происходит в встроенной среде разработки Delphi.

Delphi - структурированный, объектно-ориентированный язык программирования со четкой статической типизацией переменных. Одна из основных задач Delphi – проектирование и разработка приложений с объектно-ориентированными расширениями.

В настоящее время термин ObjectPascal чаще всего употребляется в значении языка среды программирования Delphi. Начиная с Delphi 7, в официальных документах Borland стала использовать название Delphi для обозначения языка ObjectPascal.

При создании языка (и здесь качественное отличие от языка C) не ставилась задача обеспечить максимальную производительность исполняемого кода или лаконичность исходного кода для экономии оперативной памяти. Изначально, язык ставил во главу угла стройность и высокую читаемость, поскольку был предназначен для обучения дисциплине программирования. Эта изначальная стройность, в дальнейшем, как по мере роста аппаратных мощностей, так и в результате появления новых парадигм, упростила расширение языка новыми конструкциями.

Особенности Delphi, учитываемые при разработке приложения:

– формальное начало любой программы четко отличается от других участков кода и должно располагаться в определенном, единственном в рамках проекта, исходном файле с расширением .dpr (тогда как другие файлы исходных текстов программы имеют расширение pas);

– идентификаторы типов, переменных, а равно и ключевые слова читаются независимо от регистра: например идентификатор SomeVar полностью эквивалентен somevar. Регистро-зависимые идентификаторы в начале компьютерной эпохи ускоряли процесс компиляции, и кроме того, позволяли использовать очень короткие имена, порой отличающиеся лишь регистром;

– в исходных файла с расширением .pas (которые, как правило, и содержат основное тело программы) на уровне языковых средств введено строгое разделение на интерфейсный раздел и раздел реализации. В интерфейсной части содержатся лишь объявления типов и методов, тогда как код реализации в интерфейсной части не допускается на уровне компиляции;

– метод или функция четко определяются зарезервированными для этого ключевыми словами «procedure” или “function”;

– начало и конец программного блока выделяются ключевыми словами “begin” и “end”.

Расширения файлов используемых в разработке программы:

- .pas — исходное расширение модуля (pascal);
- .dpr — исходное расширение проекта (pascal);
- .dproj — исходное расширение проекта (xml);
- .dproj.local — исходное расширение проекта (xml);
- .dfm — исходное расширение формы;
- .dpk — исходное расширение проекта пакета;
- .bpl — скомпилированный пакет;
- .dcu — скомпилированный модуль;
- .exe — скомпилированное приложение;
- .res — ресурсы;
- .dsk — привязки к файлам;
- .identcache — кэшированные привязки к файлам;

Средства Delphi для описания нечетких множеств.

В процессе разработки приложения “Нечеткие множества” использовались средства работы с графическим интерфейсом ввода-вывода данных, средства табличного инструментария обработки информации.

Используемый графический интерфейс основывается на применении класса “TCanvas”, являющееся свойством различных визуальных компонентов Delphi.

На поверхности компонентов, имеющих канву, можно печатать тексты, рисовать пером, красить кистью и отображать картинки.

Канва предоставляет «холст» (контекст графического устройства GDI), инструменты для рисования (перо, кисть, шрифт), а так же набор функций по изображению рисунков и типовых геометрических фигур типа эллипса, линии и т.п. Канвой обладают такие компоненты, как форма (TForm), наследники класса TgraphicControl – метка (TLabel), панель для рисования (TPaintBox) и др.

Канва (класс TCanvas) для рисования включает в себя шрифт (свойство Font), перо (свойство TPen), кисть (свойство TBrush) и дискриптор (свойство Handle). На поверхности канвы можно рисовать любым из следующих способов:

1. Рисовать сложные графические изображения по отдельным точкам с помощью свойства Pixels [X, Y:Integer]:TColor.
2. Помещать готовые графические изображения, задаваемые полем Graphic.
3. Проводить линии и изображать простые геометрические фигуры текущим пером. Если геометрическая фигура замкнутая, то после её рисования её внутренность закрашивается текущей кистью.

Для «сбрасывания» текущих установок шрифта, пера и кисти (установки их значений по умолчанию) используется метод Refresh.

При изменении канвы (то есть при любом рисовании) возбуждаются события OnChange и OnChanging (соответственно до начала изменений и по их завершению).

В качестве средств ввода-вывода информации используются компоненты Label, ComboBox.

Компонент Label. Для того чтобы вывести на форму текст чаще всего используется базовый компонент Label. Компонент Label предназначен для отображения текста, который в свою очередь не меняется на форме с момента начала и до конца выполнения программы.

Сведения, отображаемые в Label, определяются значением его свойства Caption. Свойство можно устанавливать в процессе проектирования или задавать и изменять программно во время выполнения приложения. Пример приведенного окна, использования компонента label, соответствует выполнению команды, приведен на рисунке 1.16.

Label1.Caption := 'Пример текста';

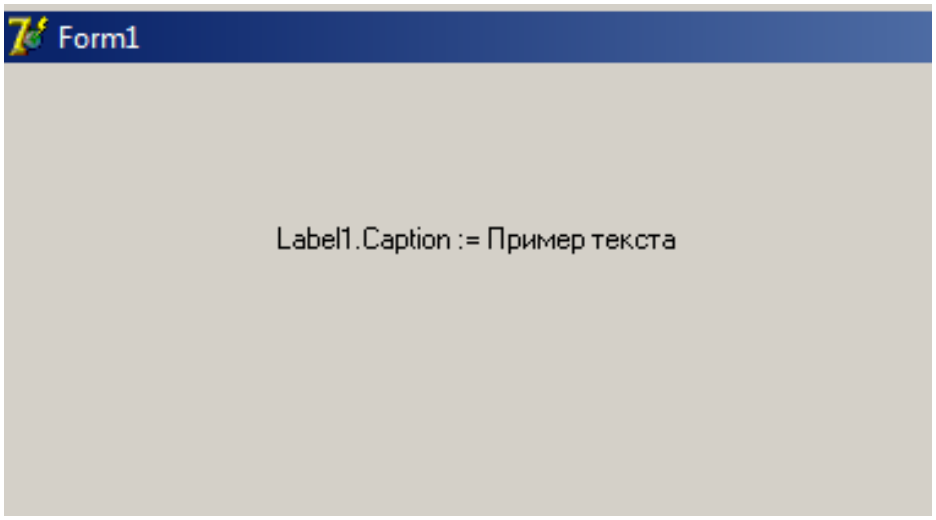

Рисунок № 1.16 - Использование компонента label

Компонент ComboBox отображает списки строк. ComboBox позволяет также редактировать данные, отображать список как в развернутом виде, так и в виде выпадающего списка, что обычно удобнее, так как экономит площадь окна приложения. Пример использование компонента ComboBox приведен на рисунке 1.17.

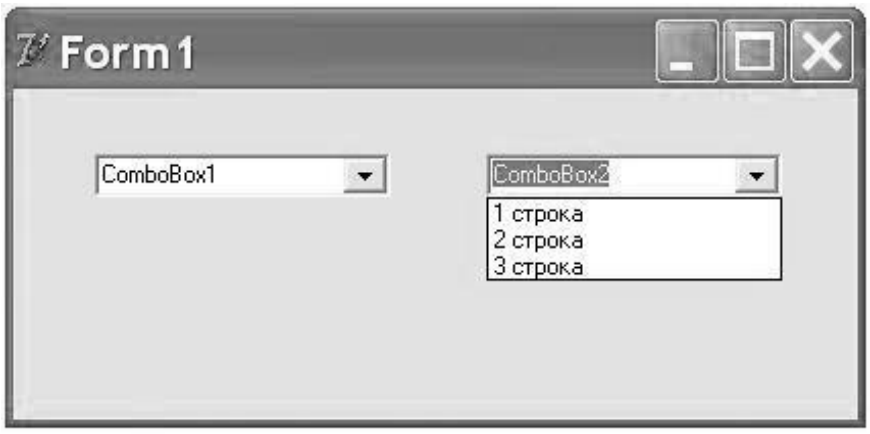

Рисунок № 1.17 - Использование компонента ComboBox

КомпонентStringGridDelphi находится на странице Additional палитры компонентов. StringGrid - компонент для отображения различных данных в табличной форме. Как следует из названия, ячейки компонента StringGridDelphi могут содержать данные, имеющие тип String, а также отображать графику.

Таблица StringGrid состоит из выделенных серым FixedCols и FixedRows - зафиксированных ячеек-заголовков, и обычных, белых ячеек. Содержимое Fixed ячеек недоступно редактированию, и меняется только программно. За возможность редактирования обычных ячеек отвечает одно из значений свойства Options. Пример использования на форме компонента StringGrid приведен на рисунке 1.18.

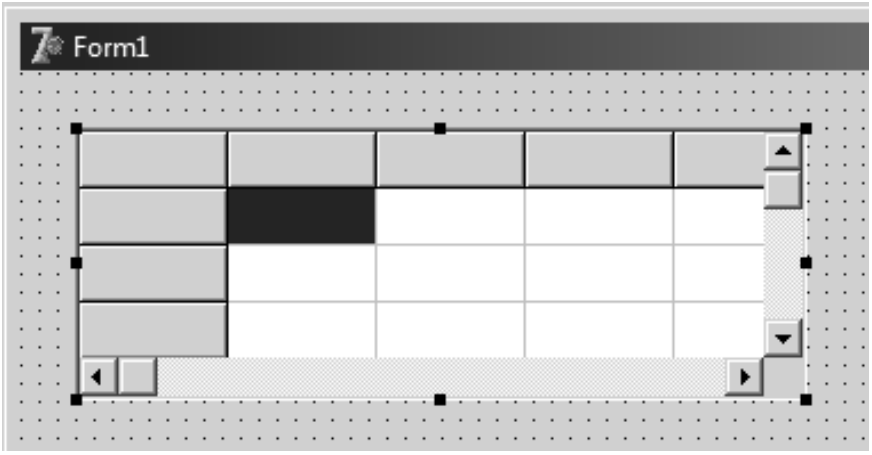

Рисунок № 1.18 - Использования на форме компонента StringGrid

Преимущества Delphi по сравнению с аналогичными программными продуктами:

– быстрота разработки приложения (RAD);

– высокая производительность разработанного приложения;

– низкие требования разработанного приложения к ресурсам компьютера;

– наращиваемость за счет встраивания новых компонент и инструментов в среду Delphi;

– возможность разработки новых компонентов и инструментов собственными средствами Delphi (существующие компоненты и инструменты доступны в исходных кодах);

– удачная проработка иерархии объектов.

Система программирования Delphi рассчитана на программирование различных приложений и предоставляет большое количество компонентов для этого. К тому же, работодателей интересует, прежде всего, скорость и качество создания программ, а эти характеристики может обеспечить только среда визуального проектирования, способная взять на себя значительные объемы рутинной работы по подготовке приложений, а также согласовать деятельность группы постановщиков, кодировщиков, тестеров и технических писателей. Возможности Delphi полностью отвечают подобным требованиям и подходят для создания систем любой сложности.

Глава 2. Структура и содержание цифрового контента по теме “нечеткие множества”

2.1 Разработка структуры цифрового контента

Одним из важных компонентов цифрового контента является разработка его структуры. Под структурой цифрового контента понимается определение порядка и способов организации алгоритма – посредством, которого будет реализовываться цифровой контент.

Существуют десятки сервисов, такие как Mindomo, Freemind, BubblUs, MindMeister, Mapul, WiseMapping, Mind42 и др., реализовать разработанную структуру цифрового контента, включающие в себя не только текст, но и другие элементы, такие как изображения, ссылки и видео. Возможность работать над картами с другими пользователями, делиться картами в социальных сетях, и интегрировать их в сайты, а так же в RTF и PDF.

В качестве средства разработки структуры цифрового контента был выбран сервис «Mindomo», ссылка на сервис: https://www.mindomo.com/ /

Mindomo — это незаменимый учебный инструмент для преподавателей и студентов. Здесь есть все для построения интеллект-карт, концепт-карт и схем в процессе обучения. Создавая карты и схемы и совместно над ними работая, студенты учатся обсуждать, критически мыслить, делать выводы, анализировать, предлагать новые идеи, решать поставленные задачи и исследовать информацию. Кроме того именно в Mindomo впервые появились такие функции, как презентация интеллект-карт в формате слайдов, постановка задач для совместной работы преподавателей и студентов, «умные карты» (частично заполненные шаблоны, которые помогают новичкам освоиться в системе интеллект-карт), а также преобразование карт в удобные схемы для редактирования в реальном времени. Эти функции очень полезны для развития исследовательского подхода в обучении — как

для студентов, так и для преподавателей. Mindomo предлагает как частично бесплатное онлайн-решение, так и бесплатные нативные приложения для мобильных платформ iPad и Android. Выбирайте удобный формат и внедряйте интеллект-карты в свой учебный процесс.

Основные возможности сервиса:

1. Совместные задачи для работы над интеллект-картами в реальном времени.
2. Интеграция с Google Apps for Education, Office365, Google Диском, Dropbox.
3. Интеграция с популярными системами управления обучением: Moodle, Canvas, Blackboard, Desire2Learn, itslearning, Schoology.
4. Экспорт интеллект-карт в другие приложения для их редактирования: .pdf, .rtf, .ppt, .txt, .opml, .mpx, .html, .zip, .png.
5. Импорт интеллект-карт из других приложений для их редактирования.
6. Создание аккаунтов для студентов без регистрации по электронной почте.
7. Загрузка изображений, видео и аудио.
8. Запись звука.
9. Пользовательские темы интеллект-карты.
10. Добавление примечаний, гиперссылок и вложений.

Структура приложения составлена в сервисе ментальных карт mindomo и реализована впоследствии в объектно-ориентированном языке программирования delphi 7, включающая в себя:

- Сравнение нечетких множеств;
- Формулировка множеств;
- Основные логические операции;

Приложение реализует основные функции, такие как тренаж и демонстрация, для реализации основных логических операций ”Нечеткого множества”:

– Инверсия (Логическое отрицание);
– Конъюнкция (Логическое умножение);
– Дизъюнкция (Логическое сложение);

На рисунке представлена структура приложения ”Нечеткое множество”.

Тестирование и отладка

Разработанное приложение ”Нечеткое множество”, полностью реализует основные операции ”Нечеткого множества”:

– Сравнение нечетких множеств;
– Формулировка множеств;
– Основные логические операции;

Приложение ”Нечеткое множество» реализовано в объектно-ориентированном языке программирования delphi 7 и представлено в исполняемом EXE-файле, что позволяет приложению:

– запуск приложения на любом персональном компьютере;
– запускается приложение без установки;
– небольшой размер файла;
– минимальные системные требования;

На рисунке 2.1, представлен пример разработанной структуры цифрового контента в сервисе Mindomo.

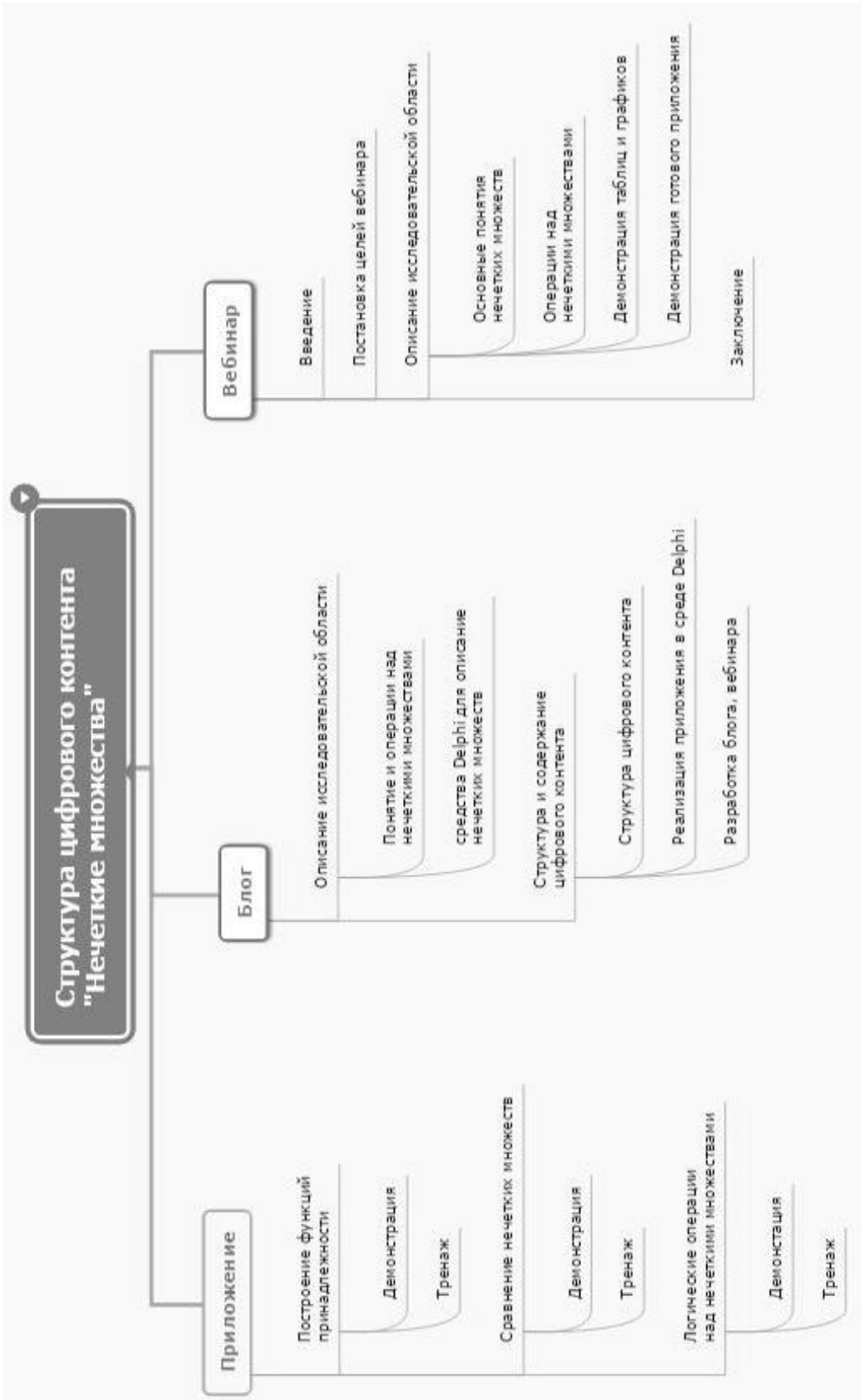

Рисунок № 2.1 - Пример разработанной структуры цифрового контента в сервисе Mindomo.

2.2 Реализация приложения “нечеткие множества” в среде Delphi

С введением нечетких чисел оказалось возможным прогнозировать будущие значения параметров, которые меняются в установленном расчетном диапазоне. Вводится набор операций над нечеткими числами, которые сводятся к алгебраическим операциям с обычными числами при задании определенного интервала достоверности (уровня принадлежности). Применение нечетких чисел позволяет задавать расчетный коридор значений прогнозируемых параметров. Тогда ожидаемый эффект оценивается экспертом также как нечеткое число со своим расчетным разбросом (степенью нечеткости).

Нечеткая логика, как модель человеческих мыслительных процессов, встроена в системы искусственного интеллекта и в автоматизированные средства поддержки принятия решений (в частности, в системы управления технологическими процессами).

В рамках изучения дисциплины “Основы искусственного интеллекта” раздела нечеткие множества “Нечеткой логики”, знакомство и изучение дисциплины на практике проходит с использованием такого программного обеспечения, как: Microsoft Excel, Mathcad, Mathlab. Несмотря на то, что функционала данных программ хватает на реализацию основных заданий, в частности, построение универсумов, их сравнение и реализация основных логических операций нечетких множеств (конъюнкция, дизъюнкция, инверсия и др.), возникает ряд проблем:

Во-первых, программное обеспечение специально не адаптировано для изучения дисциплины “Основы искусственного интеллекта”;

Во-вторых, предоставляемые пакеты программного обеспечения должны быть строго лицензионными, что в свою очередь несет финансовые затраты.

В-третьих, для использования такого программного обеспечения необходимы более мощные системные ресурсы, по сравнению с используемым приложением.

Например, приобрести лицензионный Office 365 с Excel для дома и учебы 2016 на официальном портале www.microsoft.com можно за 5 199,00 ₽, при этом необходимо обеспечить следующие ресурсы:

– процессор, 32- или 64-разрядный процессор с тактовой частотой 1 ГГц или выше с набором инструкций SSE2;

– операционная система, Windows 7 или выше, Windows 10 Server, Windows Server 2012 R2, Windows Server 2008 R2 или Windows Server 2012;

– память, 1 ГБ ОЗУ (32-разрядная версия); 2 ГБ ОЗУ (64-разрядная версия);

– свободное место на жестком диске, 3 ГБ свободного места на жестком диске;

– монитор, Разрешение 1024 x 768;

– графика, Для использования аппаратного ускорения требуется видеоадаптер, поддерживающий DirectX 10.

Для разработки приложения традиционно выделяются следующие основные 5 этапов жизненного цикла программного обеспечения:

1) анализ требований;

2) проектирование;

3) программирование;

4) тестирование и отладка;

5) эксплуатация и сопровождение;

Анализ требований

Программное обеспечение для раздела “нечеткие множества” нечеткой логики быть удобным в использовании, содержать все необходимые материалы и модули для обучения студентов дисциплины “Основы

искусственного интеллекта” института физики, технологии и экономики Уральского Государственного Педагогического университета.

Разрабатываемое программное обеспечение должно соответствовать требованиям

Разрабатываемое приложение ”Нечеткое множество” должно реализовать основные операции ”Нечеткого множества”:

- Сравнение нечетких множеств;
- Формулировка множеств;
- Основные логические операции;

Титульное окно приложения “Нечеткие множества” состоит из 4 кнопочных компонентов: Функции принадлежности, Сравнение множеств по включению, Логические операции и кнопки полного выхода из приложения. Скриншот титульного окна представлен на рисунке 2.2.

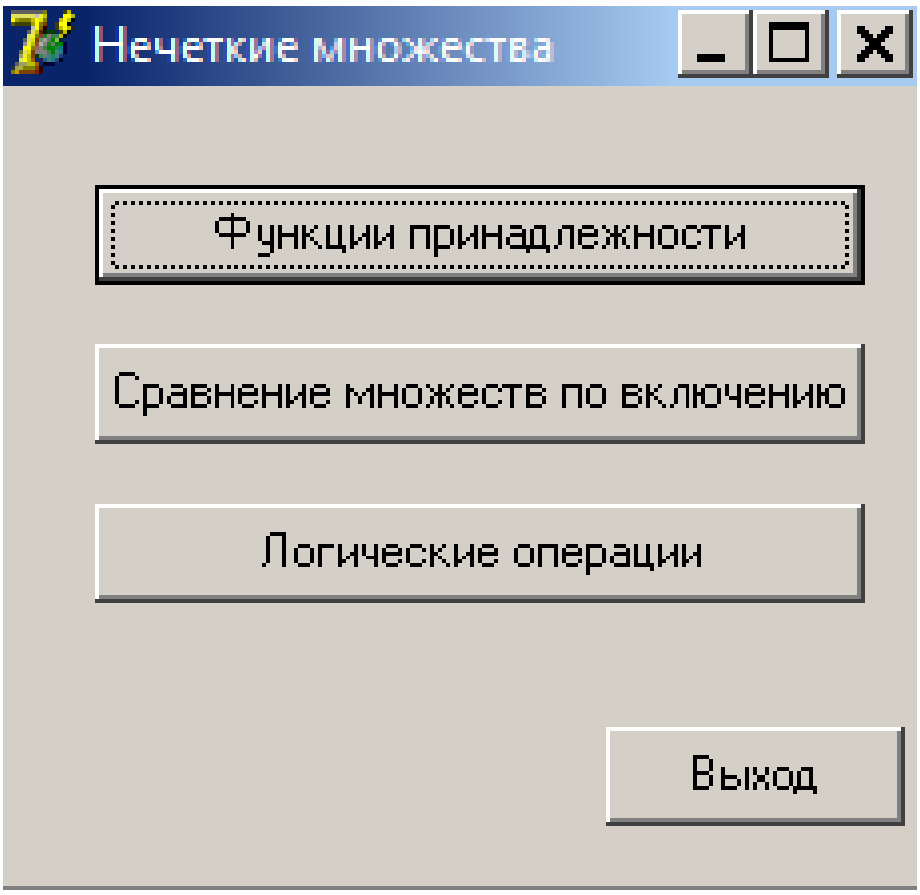

Рисунок № 2.2 - Титульное окно приложение “Нечеткие множества”.

В разделе “Функции принадлежности” разработан тренаж на построение универсума по заданным значениям. Студенту предлагается ввести полученные табличные значения и построить график. Скриншот построенного графика по табличным значениям в разделе “Функции принадлежности” представлен на рисунке 2.3.

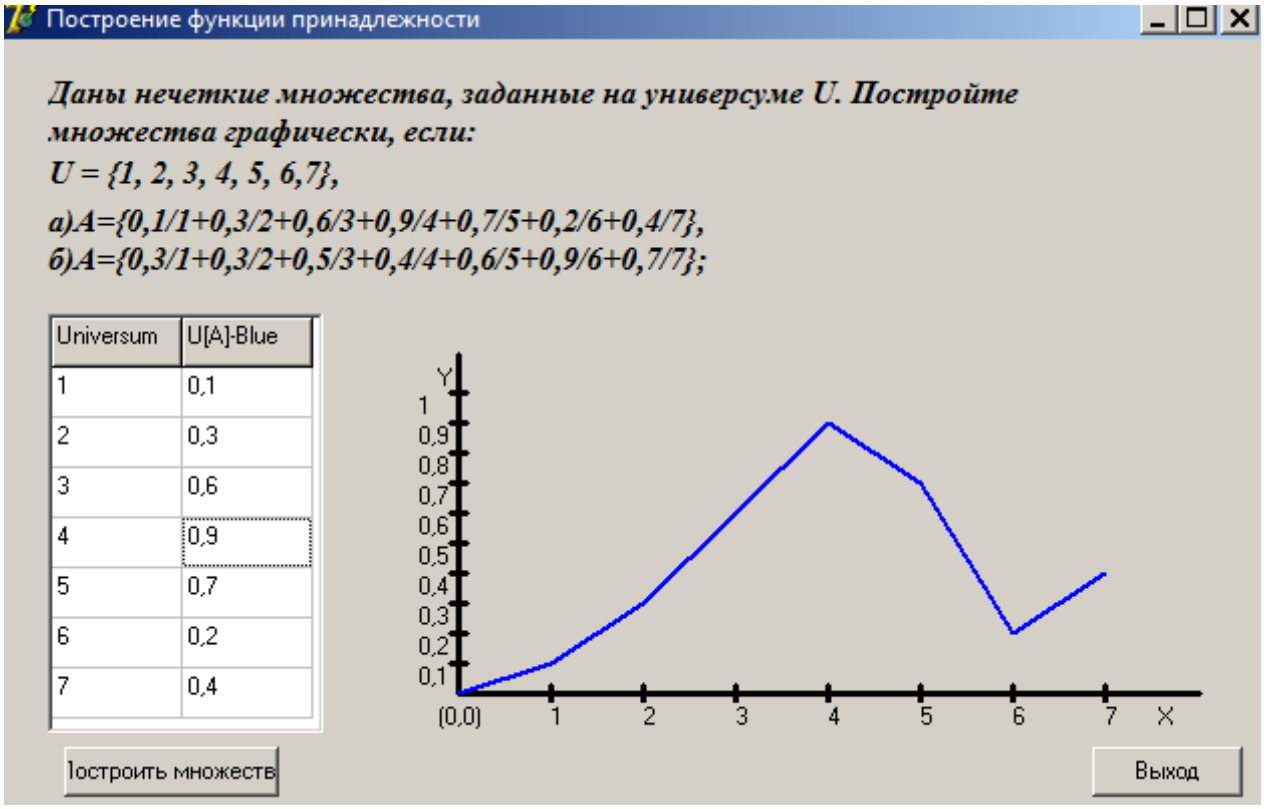

Рисунок № 2.4 - Раздел “Функции принадлежности”.

В разделе “Сравнение множеств по включению” разработан тренаж на построение универсума по случайно генерирующимся значениям. Студенту предлагается ввести построить график и сравнить Red и Blue универсумы на их включение. Скриншот, тренажа в разделе “Сравнение множеств по включению”, представлен на рисунке 2.4 страницы 36.

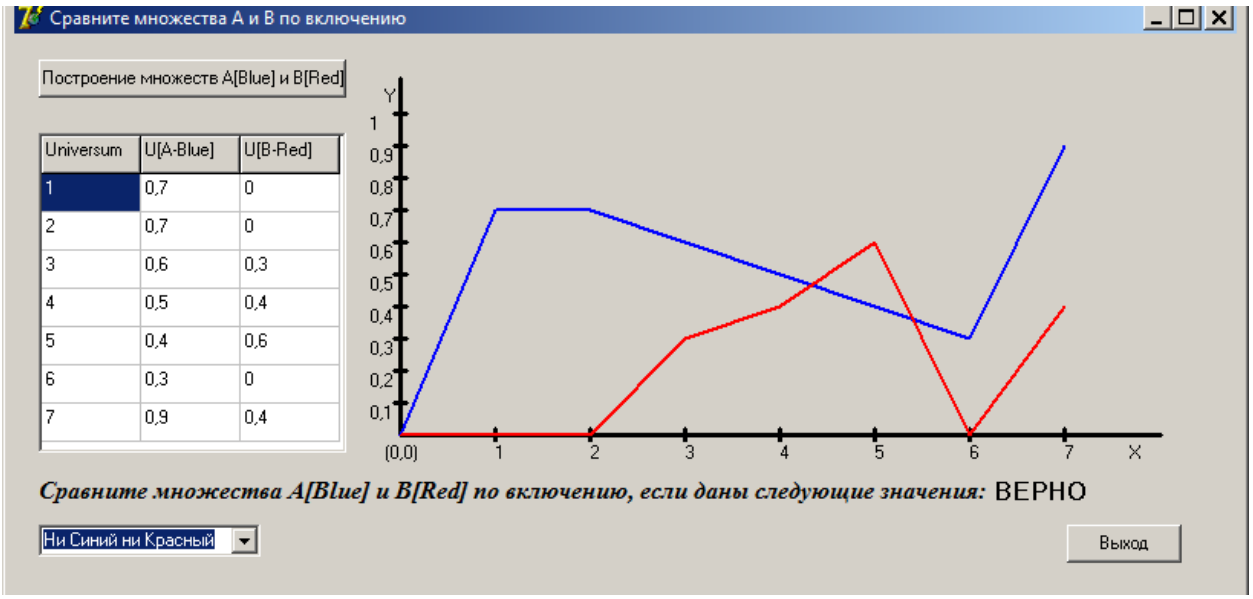

Рисунок № 2.4 - Раздел “Сравнение множеств по включению”.

Раздел “Логические операции” приложения “Нечеткие множества” состоит из 4 кнопочных компонентов: Инверсия, конъюнкция, дизъюнкция кнопки выхода в титульный раздел. Скриншот раздела “Логические операции” приложения “Нечеткие множества” представлен на рисунке 2.5.

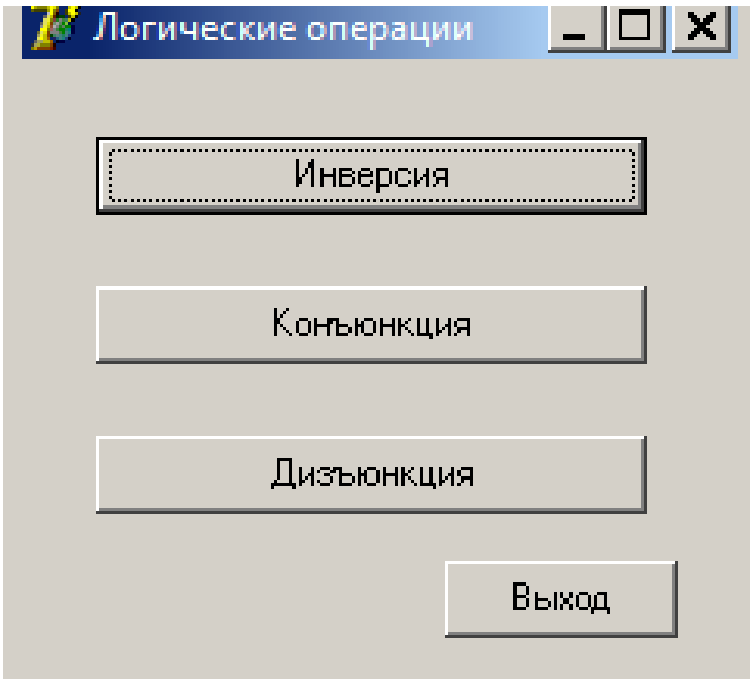

Рисунок № 2.5 - Раздел “Логические операции” приложения “Нечеткие множества”.

В разделе “Логические операции: Инверсия” разработана демонстрация построения универсума по случайно генерирующимся значениям U[A] и его инверсии U[-A] на графике. Студенту предлагается построить график и ознакомится с инверсией нечетких множеств. Скриншот раздела “Логические операции: Инверсия” представлен на рисунке 2.6.

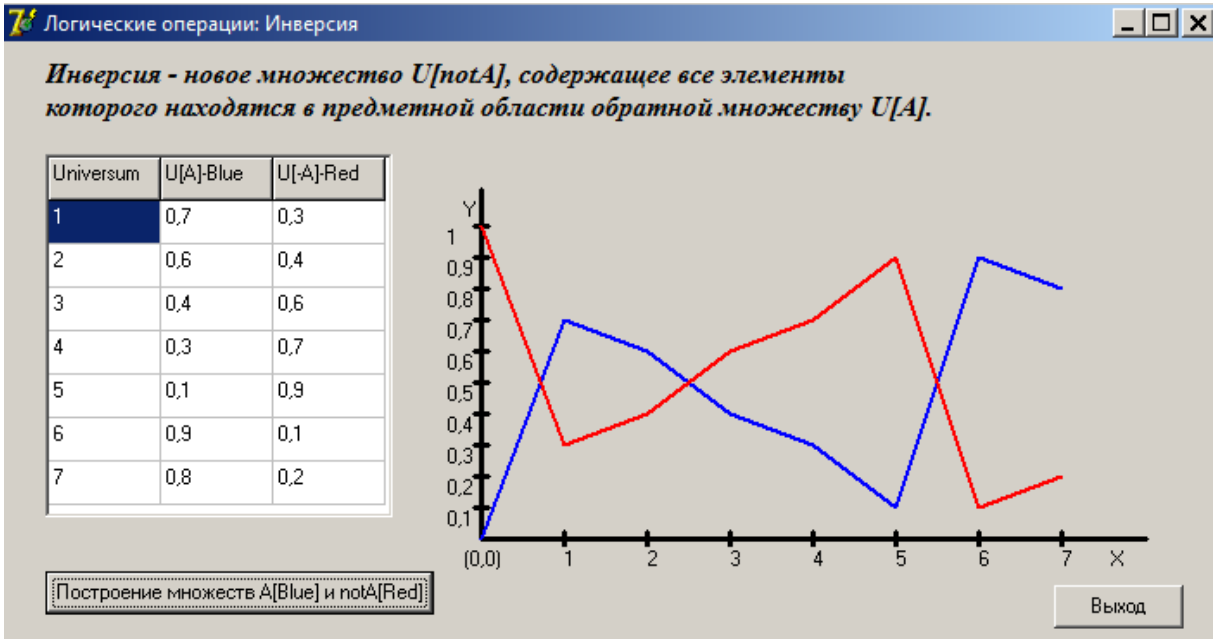

Рисунок № 2.6 - Раздела “Логические операции: Инверсия”.

В разделе “Логические операции: Конъюнкция” разработана демонстрация двух универсумов по случайно генерирующимся значениям, расчет следующего универсума и его построение графически. Студенту предлагается построить график и ознакомится с конъюнкцией нечетких множеств. Скриншот раздела “Логические операции: Конъюнкция” представлен на рисунке 2.7.

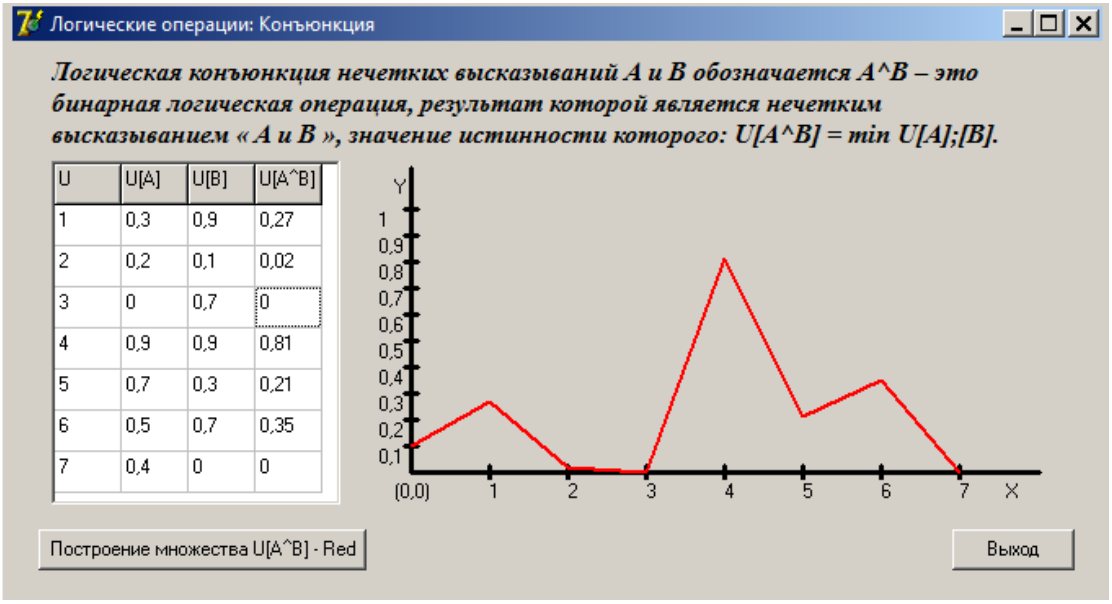

Рисунок № 2.7 - Раздел “Логические операции: Конъюнкция”

В разделе “Логические операции: Дизъюнкция” разработана демонстрация двух универсумов по случайно генерирующимся значениям, расчет следующего универсума и его построение графически. Студенту предлагается построить график и ознакомится с дизъюнкцией нечетких множеств. Скриншот раздела “Логические операции: Дизъюнкция” представлен на рисунке 2.8.

Рисунок № 2.9 - Раздел “Логические операции: Дизъюнкция”

2.3 Разработка блога по теме: «Нечеткие множества»

Блог (англ. blog, от web log) — интернет ресурс, личный журнал либо дневник блогера, включающий в себя такие функции как, добавление текста, мультимедиа, графических средств и так далее. Блоги содержать короткие записи, упорядоченные в обратном временном порядке (последняя запись сверху). Отличия блога от традиционного дневника обусловливаются средой: блоги обычно публичны и предполагают сторонних читателей, которые могут вступить в публичную полемику с автором.

Для блогов характерна возможность публикации отзывов (комментариев) посетителями. Она делает блоги средой сетевого общения, имеющей ряд преимуществ перед электронной почтой, группами новостей, веб-форумами и чатами. Под блогами также понимаются персональные сайты, которые состоят в основном из личных записей владельца блога и комментариев пользователей к этим записям.

Первым блогом считается страница Тима Бернерса Ли, где он, начиная с 1992 г., публиковал новости. Более широкое распространение блоги получили с 1996 г. В августе 1999 г. компьютерная компания Pyra Labs из Сан-Франциско открыла сайт Blogger.com, который стал первой бесплатной блоговой службой.

В настоящее время особенность блогов заключается не только в структуре записей, но и в простоте добавления новых записей. Пользователь просто обращается к веб-серверу, проходит процесс идентификации пользователя, после чего он добавляет новую запись к своей коллекции. Сервер представляет информацию как последовательность сообщений, помещая в самом верху самые свежие сообщения. Структура коллекции напоминает привычную последовательную структуру дневника или журнала.

Блог-платформа (блогохостинг, блог-служба) — это «коммунальный» сервис, предоставляющий пользователю «под ключ» движок и позволяющий

вести блог, дневник без необходимости самостоятельно заниматься обслуживанием движка и программированием. Пользователю достаточно получить навыки работы с веб-сервисами, что гораздо легче. Обратная сторона такого удобства — невозможность полноценной настройки блога, за исключением шаблонов, определяющих его дизайн. Кроме того контент пользователя с правовой точки зрения находится под контролем владельца блог-платформы и принадлежит пользователю лишь номинально. Как следствие, последний ограничен в свободе самовыражения, что обычно прямо прописано в правилах предоставления сервиса и зачастую контролируется «конфликтной командой» владельца блог-платформы.

При написании блога, но обязательно разрабатывать собственную платформу, существует возможность использования любух из «открытых» блог-платформ, которые предлагают множество различных функций, такие как WordPress.com, Blog.com, Blogger, TypePad Micro, Jux, Tumblr и др..

Основные правила написание блога:

Правило №1 Контент.

Блог должен быть уникален. Чтобы сайт индексировался, необходимо писать в блог каждый день. Для блогов это обязательное условие.

Правило №2 Навигация

Очень важен внешний вид блога, т.е. его оформление. Пользователь оценит только порядок, когда реклама находится там, где должна быть реклама, а все остальное – это контент и навигация по сайту.

Правило №3 Внутренняя перелинковка

Возможность ставить ссылки на другие статьи в самом посте. Можно под постом расположить блок с похожими статьями. Также можно расположить блок с “Популярными записями”, “Последние записи” и т.п. в сайдбаре.

Blogger — веб-сервис для ведения блогов, с помощью которого любой пользователь может завести свой блог, не прибегая к программированию и не

заботясь об установке и настройке программного обеспечения. Blogger был создан компанией Pyra Labs (англ.)русск., которой сейчас владеет Google. До 1 мая 2010 посты Blogger могли автоматически переноситься на хостинг владельца блога при помощи FTP или SFTP.

На рисунке 2.10 представлен пример титульной формы сервиса blogger, для начала работы с блогом. При создании нового блога автору предстоит заполнить титульную форму, которая состоит из трех элементов: название нового блога, адрес расположения блога и один из готовых предлагаемых шаблонов.

Рисунок № 2.10 - Титульная форма сервиса blogger.

Особенности сервиса Blogger.

- Blogger разрешает автору блога зарабатывать деньги посредством размещения рекламы с помощью принадлежащего Google сервиса AdSense.
- Blogger предлагает возможность вести блог коллективом из нескольких авторов.
- Blogger представляет возможность разместить блог на собственном домене.

В платформу Blogger в виде гаджета интегрирована технология Google Friend Connect, позволяющая добавить социальные функции к своему блогу. Это дополнение, объединяющее подписки читателей с разных сайтов, появилось после замены виджета «Постоянные читатели»

Blogger имеет множество различных шаблонов, как стандартных, так и любительских. На рисунке 2.11 представлен пример интерфейса сервиса Blogger.

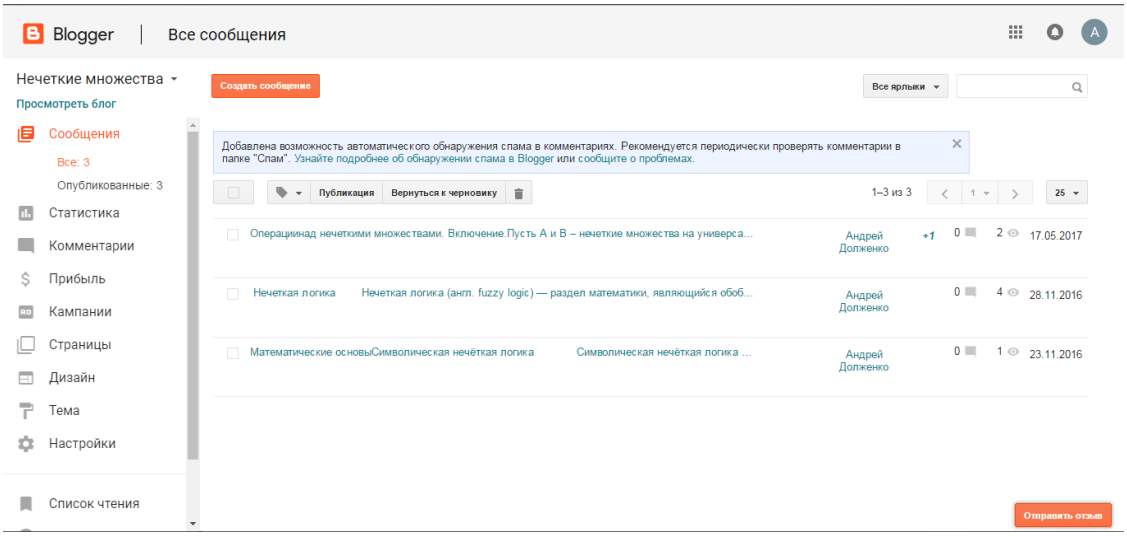

Рисунок № 2.11 – Пример интерфейса сервиса Blogger.

Окно разработки блога сервиса Blogger состоит из стандартных интуитивно понятных инструментов таких как: изменение размера и темы шрифта, его начертание, цвета, приложение к блогу гиперссылок, графических элементов, видео-(аудио-)материалов. Так же содержит дату и местоположение на момент публикации поста автора. Пример разработки блога по теме «Нечеткие множества» представлен на рисунке 2.12.

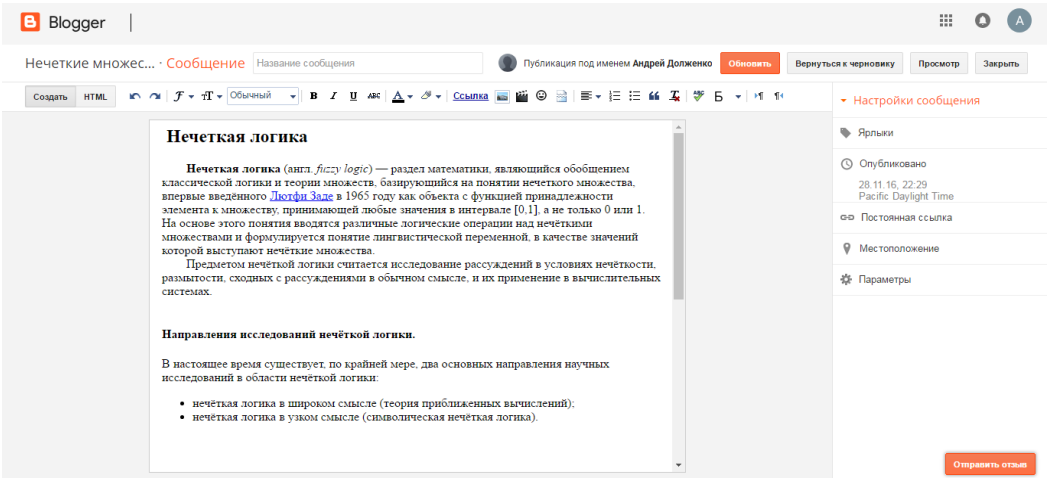

Рисунок № 2.12 - Разработка блога по теме «Нечеткие множества»

Пример разработанного и опубликованного блога по теме «Нечеткие множества» представлен на рисунке 2.13 страницы 44. Готовый блог содержит конечное форматирование текста, примененную тему для блога.

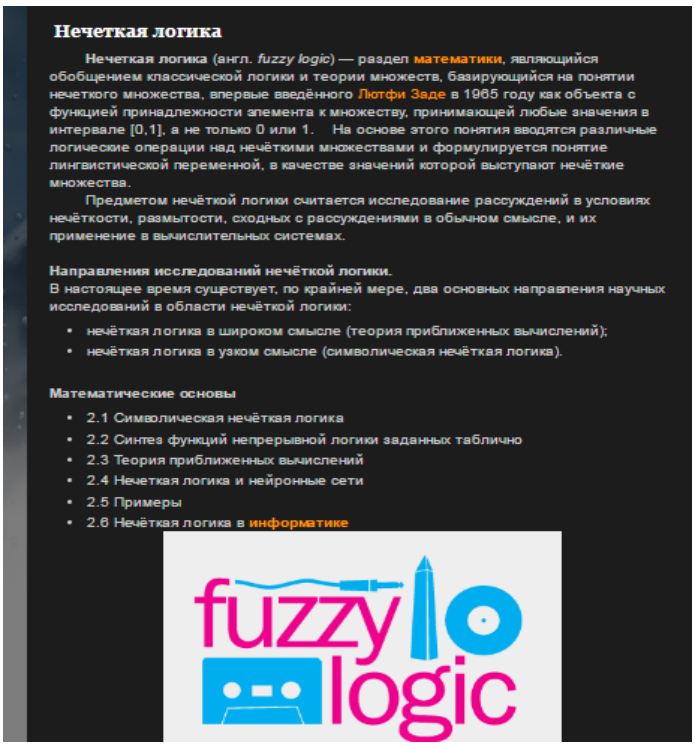

Рисунок 2.13 Разработанный и опубликованный блог по теме «Нечеткие множества».

2.4 Разработка вебинара по теме: «Нечеткие множества»

Вебинар – это современный мультимедийный инструмент для организации дистанционного обучения и делового общения с помощью интернет-технологий.

Вебинар (от англ. «webinar», сокр. от «Web-basedseminar») – онлайн-семинар, лекция, курс, презентация, организованный при помощи web-технологий в режиме прямой трансляции. Каждый участник находится у своего компьютера, вне зависимости от географии и месторасположения.

С помощью вебинара можно провести семинар, доклад, лекцию, обучающий курс, тренинг, презентацию, совещание, встречу, конференцию и проч. в режиме прямой трансляции, когда все участники находятся удалённо друг от друга. Для того, чтобы участвовать в вебинаре, достаточно иметь подключение к интернету и работающие колонки. Микрофон может понадобиться, если возникнет желание высказаться голосом.

Технически вебинар можно организовать, используя web-технологии. А именно связь в онлайн-режиме между участниками осуществляется с помощью специального сервера, куда можно попасть по пригласительной интернет-ссылке платно или бесплатно. Этот сервер – это своего рода виртуальная «аудитория», объединяющая всех участников на расстоянии. На рисунке 2.14 представлен пример распределения информации скипером при проведении вебинара.

Как правило, проведение вебинара выглядит таким образом: его участники получают от организаторов ссылку, на которую должны «кликнуть» в определенное время, подключившись к вещанию. Слушая спикера, просматривая видео и слайды на экране монитора, участники могут взаимодействовать с ведущим, задавать ему вопросы и получать ответы, выполнять задания. В целом вебинар — удобное средство для проведения интернет-конференций любого типа.

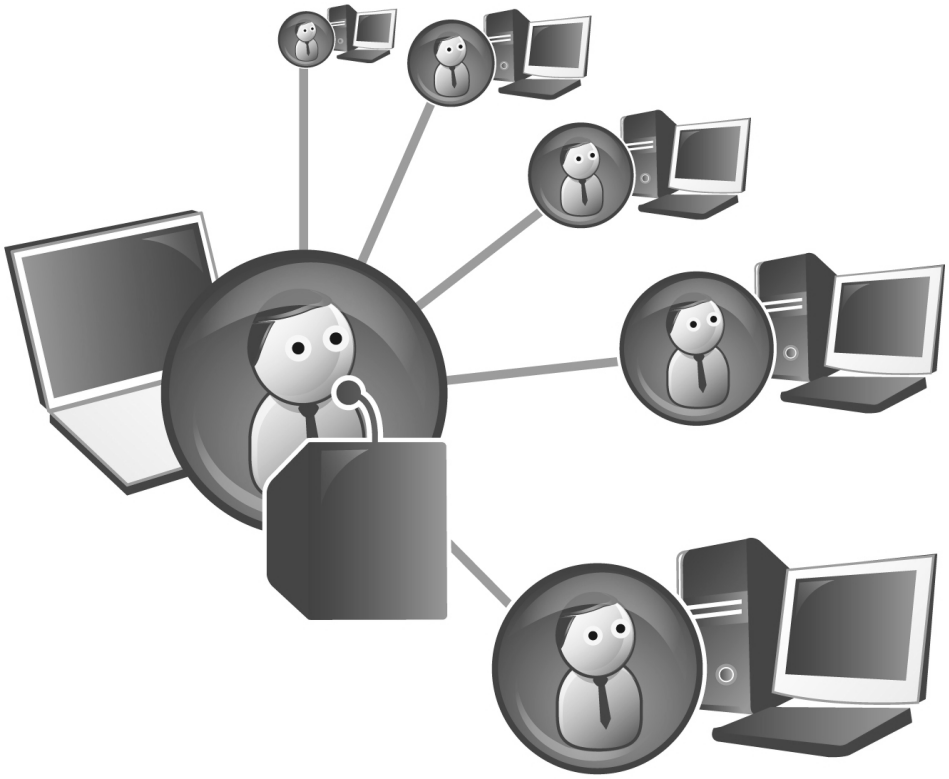

Рисунок 2.14 Распределение информации при проведении вебинара.

Преимущества вебинаров:

- Как правило, вебинары проводятся в прямом эфире. То есть это дает возможность общаться с профессором, с остальными слушателями курса, задавать интересующие вас вопросы, общаться в чате с лектором и участниками вебинара;
- Неограниченное количество учеников может присоединиться к прослушиванию лекции. То есть через час у вас будет возможность почерпнуть и знания студентов, возможно, обменяться контактами и обсудить интересующий вас вопрос или же поучаствовать в дискуссии между преподавателем и слушателями;
- Огромный плюс вебинаров в том, что вы можете слушать лекции интересующих вас профессоров из любого уголка мира. Главное, владеть языком и тогда ничего не сможет помешать вам стать участником вебинара,

который транслируется из Англии, США, Китая и так далее. Также имеется возможность просматривать его из любого уголка мира;

- После окончания урока вы можете пересмотреть видеозапись вебинара, воспроизвести какие-то моменты, если это будет необходимо;
- Вебинар предоставляет возможность не тратить время на то, чтобы добраться до места проведения курсов, вам не нужно ехать в другой конец города, чтобы прослушать лекцию. Вы можете просмотреть его в удобное для вас время;
- Вебинар существенно дешевле семинара, а часто и вовсе бесплатный. Вам не нужно тратить деньги на билет, а порой даже на путешествие и проживание в другом городе/ стране. Достаточно просто иметь подключение к интернету. Конечно, есть и платные вебинары, но плата, как правило, небольшая.

В качестве сервиса для проведения вебинара мной был выбран сервис «Webinar.ru». Запись вебинара доступна на сайте. Наличие бесплатного аккаунта. Позволяет демонстрировать документы Power Point, Word, Excel, транслировать свой экран, обмениваться сообщениями в чате. Титульная страница сервиса «Webinar.ru» представлена на рисунке 2.15 страницы 48.

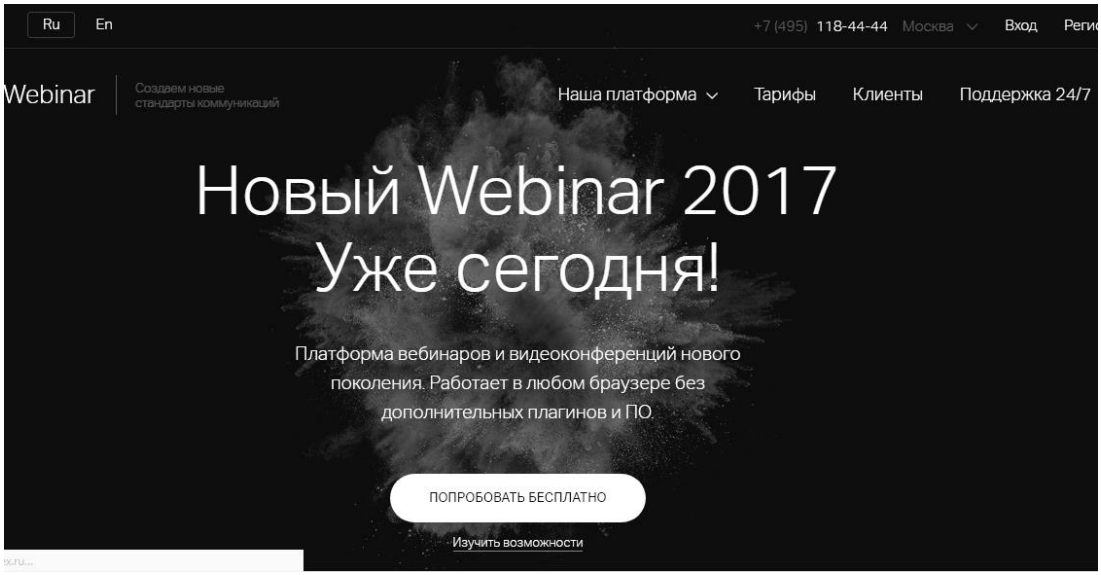

Рисунок 2.15 Титульная страница сервиса «Webinar.ru»

Функциональность вебинаров сервиса «Webinar.ru»:

– Демонстрация презентаций и видеофайлов. Имеет возможность показывать презентации, видеофайлы или демонстрируйте экран, доступно общение в общем чате или через приватные сообщения, обмен документами и ссылками. Возможность включения режима модерации комментариев;

– Расширенные возможности для проведения тестов и голосований. Проведение онлайн-тестов и голосования, чтобы узнать мнение ваших слушателей и уровень усвоения материала. Задавайте разные типы вопросов и ответов. Используйте изображения. Устанавливайте минимальный проходной балл и ограничивайте время на прохождение теста;

– Быстрое освоение и запуск без установки плагинов и ПО. Новый WEBINAR основан на современных веб-технологиях HTML5 и Web-RTC. Он не нагружает интернет-канал и не требует специальной настройки корпоративной системы безопасности;

– Быстрый запуск в браузерах. Работает на всех популярных платформах и браузерах. Доступен на мобильных устройствах. Запуск первого вебинара занимает менее 1-й минуты и не требует установки плагинов и приложений;

– Минимальные усилия для освоения. Быстрый старт без скучных инструкций. Онлайн справка по сервису в виде интерактивного помощника, который шаг за шагом подведет вас к запуску вебинара;

– Удобный файловый менеджер. Возможность облачного хранения презентаций, тестов и видео в единой файловой системе. Выстраивайте нужную структуру хранения документов, помечайте файлы тегам.

Пример, записи вебинара по теме «Нечеткие множества», представлен на рисунке 2.16, сама запись вебинара доступна по ссылке: https://events.webinar.ru/2549245/425803/record-new/435783

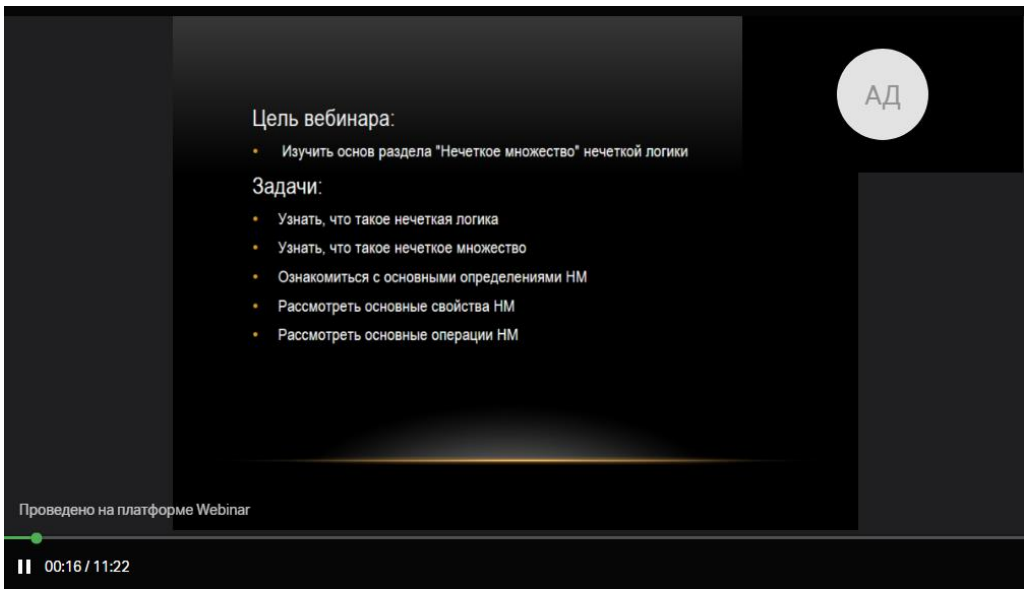

Рисунок 2.16 Запись вебинара по теме «Нечеткие множества»

На сегодняшний день вебинар — одно из наиболее прогрессивных средств дистанционного обучения, к которому прибегают люди, идущие в ногу со временем. Уже сегодня оно прочно входит в нашу жизнь и скоро станет неотъемлемой частью таких уже привычных для нас технологий, как ICQ, e-mail, Skype и т.д.

Заключение

Нечеткая логика ближе по духу к человеческому мышлению и естественным языкам, чем традиционные логические системы. Она обеспечивает эффективные средства отображения неопределённостей и неточностей реального мира.

С введением нечетких чисел оказалось возможным прогнозировать будущие значения параметров, которые меняются в установленном расчетном диапазоне. Вводится набор операций над нечеткими числами, которые сводятся к алгебраическим операциям с обычными числами при задании определенного интервала достоверности (уровня принадлежности). Применение нечетких чисел позволяет задавать расчетный коридор значений прогнозируемых параметров. Тогда ожидаемый эффект оценивается экспертом также как нечеткое число со своим расчетным разбросом (степенью нечеткости).

Системы, основанные на нечетких множествах, разработаны и успешно внедрены в таких областях, как: управление технологическими процессами, управление транспортом, медицинская диагностика, техническая диагностика, финансовый менеджмент, биржевое прогнозирование, распознавание образов. Спектр приложений очень широкий - от видеокамер и бытовых стиральных машин до средств наведения ракет ПВО и управления боевыми вертолетами.

В результате исследования получены следующие результаты:

– Проанализирована учебная литература по теме исследования;

– Раскрыта сущность понятия «Нечеткое множество»;

– Проанализированы различные логические операции над нечеткими множествами;

– Определена и разработана структура цифрового контента для раздела “нечеткие множества” нечеткой логики;

– Разработано приложение для дисциплины “Основы искусственного интеллекта” для операционной системы Windows;

– Разработан, записан и проведен вебинар по материалу раздела “нечеткие множества” нечеткой логики;

– Разработан и опубликован блог для раздела “нечеткие множества” нечеткой логики.

– Приложение апробировано при проведении занятий в группе БИ-41;

На основании вышеизложенного, можно сделать вывод о том, что цель работы достигнута, поставленные задачи решены.

Список литературы

1. Абдалова О. И. Применение интернет-курсов в дистанционном обучении / О.И. Абдалова, А.В. Гураков, С.В. Сметанин и др. // Открытое и дистанционное образование: журнал. — 2012 .— №3 .— С. 14-17.
2. Аверкин А.Н., Нечеткие множества в моделях управления и искусственного интеллекта М.: Наука, 2012. 312с.
3. Ализар А., Психология блоггера: научное исследование // Электронный ресурс Режим доступа: свободный http://www.webplanet.ru/news/internet/2004/3/23/blog_survey.html,. Дата обращения: 26.05.2017.
4. Андреев А. А. Становление и развитие дистанционного обучения в России / А.А. Андреев // Высшее образование в России: журнал. — 2012 .— №10 .— С. 106-111.
5. Андреев А. А., Солдаткин В. И. Дистанционное обучение и дистанционные образовательные технологии // Cloud of science. 2013. №1. Режим доступа: http://cyberleninka.ru/article/n/distantsionnoe-obuchenie-i-distantsionnye-obrazovatelnye-tehnologii, свободный. Дата обращения: 26.05.2017.
6. Белов В., Чистякова В. Программирование в Delphi: Процедурное, объектно-ориентированное, визуальное: учеб, пособие для высших учебных заведений. М.: Горячая линия - Телеком, 2014г.,240 с.
7. Блог [Электронный ресурс]: Электронная свободная энциклопедия. Режим доступа:: https://ru.wikipedia.org/wiki/Блог, свободный. Дата обращения: 26.05.2017.
8. Борисов С.В., Комалов С.С., Грин Д., Введение в среду визуального программирования Turbo Delphi/интернет-издание, 2012. 267 с.

9. Борисов С.В., Комалов С.С., Грин Д., Введение в среду визуального программирования Turbo Delphi М.: МГТУ им. Н.Э. Баумана, 2012. - 267 с.

10. Бьюзен Т. Руководство по развитию памяти и интеллекта // Т. Бьюзен; пер. с англ. Н.В. Жулаева. – Минск М.: Попури, 2014. – 144 с.

11. Вебинар [Электронный ресурс]: Электронная свободная энциклопедия. Режим доступа:: https://ru.wikipedia.org/wiki/Онлайн-семинар, свободный. Дата обращения: 26.05.2017.

12. ГОСТ 34.601-90 Автоматизированные системы. Стадии создания.

13. Культин Н. Программирование в Turbo Pascal 7.0 и Delphi/ Культин Н., М.: Изд-во БХВ-Екатеринбург, 2012. - 400 с.

14. Культин Н., Основы программирования в Delphi/ интернет-издание, 2015. 232 с.

15. Культин Н.Б., Delphi в задачах и примерах, 3-е изд., М.: Изд-во БХВ-Екатеринбург, 2012. 288 с.

16. Леонов В., Обучение мобильной разработке на Delphi. М.: Изд-во Embarcadero, 2016. 341 с.

17. Неудачин И.Г., Таблицы Delphi для управления базами данных. М.: Изд-во Урал. ун-та, 2016. 96 с.

18. Нечеткая логика [Электронный ресурс]: Электронные курсы. Режим доступа: http://www.intuit.ru/studies/courses/87/87/lecture/20503, свободный. Дата обращения: 26.05.2017.

19. Нечеткое множество [Электронный ресурс]: Электронная свободная энциклопедия. Режим доступа: https://ru.wikipedia.org/wiki/Нечёткое_множество, свободный. Дата обращения: 26.05.2017.

20. Онлайн учебник по Delphi 7: электрон / 2008-2017. URL: http://delphi.support.uz/index.php (дата обращения: 25.03.2017).

21. Определение и основные характеристики нечетких множеств [Электронный ресурс]: учебное пособие/ М.: БГТУ им. В.Г.Шухова. Режим доступа: http://nrsu.bstu.ru/chap21.html, свободный. Дата обращения: 26.05.2017.

22. Осипов Д., Delphi. Программирование для Windows, OS X, iOS и Android. М.: Изд-во БХВ-Екатеринбург, 2014. 464 с.

23. Основные понятия и определения теории нечетких множеств [Электронный ресурс]: Научная библиотека. Режим доступа: http://sernam.ru/book_gen.php?id=15, свободный. Дата обращения: 26.05.2017.

24. Основы теории нечетких множеств [Электронный ресурс]: Электронные курсы. Режим доступа: http://www.intuit.ru/studies/courses/87/87/lecture/20501, свободный. Дата обращения: 26.05.2017.

25. Попова Н. Е., Чикова О. А. Технологии дистанционного обучения как инновация в процессе реализации образовательных стандартов нового поколения // Вестник НГПУ. 2014. №2 (18). Режим доступа: http://cyberleninka.ru/article/n/tehnologii-distantsionnogo-obucheniya-kak-innovatsiya-v-protsesse-realizatsii-obrazovatelnyh-standartov-novogo-pokoleniya, свободный. Дата обращения: 26.05.2017.

26. Проектирование систем управления\Fuzzy Logic Toolbox [Электронный ресурс]: Электронные курсы. Режим доступа: http://matlab.exponenta.ru/fuzzylogic/book1/1.php, свободный. Дата обращения: 26.05.2017.

27. Сиббет Д., Визуализируй это! Как использовать графику, стикеры и интеллект-карты для командной работы. М.: Альпина Паблишер, 2013г. 280 с.

28. Справка по Blogger https://support.google.com/blogger.

29. Строганов Б.Г. Обучение через Web [Электронный ресурс]: учебное пособие/ Строганов Б.Г.— Электрон. текстовые данные. — М.:

Российский университет дружбы народов, 2013. — 100 с.— Режим доступа: http://www.iprbookshop.ru/22196.html. — ЭБС «IPRbooks»

30. Троичная логика [Электронный ресурс]: Электронная свободная энциклопедия. Режим доступа:: https://ru.wikipedia.org/wiki/Троичная_логика, свободный. Дата обращения: 26.05.2017.

31. Шитова И.Ю., Гончарова В.П., Ментальные карты: инструмент обработки и систематизации информации // Научные исследования: от теории к практике: Чебоксары: ЦНС «Интерактив плюс», 2015. — № 5 (6). — С. 175–176. — ISSN 2413-3957.

Приложение

Фрагмент кода программы “Нечеткие множества”:

```
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Grids;
type
  TForm5 = class(TForm)
    Button1: TButton;
    PaintBox1: TPaintBox;
    StringGrid1: TStringGrid;
    Button2: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form5: TForm5;
  x,y,j,x0,y0,x1,y1,xmax,ymax,kpx,kpy,n:integer;
  i,v:real;
implementation
{$R *.dfm}
const ofs = 20;  //константа отступа
```

```
procedure TForm5.Button1Click(Sender: TObject);
var xd,yd,t:real;
begin
PaintBox1.Refresh;
PaintBox1.Canvas.Pen.Color:=clBlack;
PaintBox1.Canvas.FillRect(PaintBox1.BoundsRect);
PaintBox1.Canvas.Pen.Width:=3;
x := PaintBox1.Clientwidth;   //ширина пейнт бокса
y := PaintBox1.ClientHeight;  //высота пейнт бокса
x0 := ofs;   // новые координаты x
y0 := y-ofs; // новые координаты y
xmax:= x-ofs ;
ymax:= ofs  ;
PaintBox1.Canvas.MoveTo(x0, y0); // ось ox
PaintBox1.Canvas.LineTo(xmax,y0);
PaintBox1.Canvas.MoveTo(x0, y0); // ось oy
PaintBox1.Canvas.LineTo(x0,ymax);
PaintBox1.Canvas.TextOut(10, y-15,'(0,0)');
PaintBox1.Canvas.TextOut(x-40, y-15,'X');
PaintBox1.Canvas.TextOut(0+10, 25,'Y');
kpx:= (xmax - x0)div 8;
kpy:= (y0 - ymax)div 11;
i:=0;
n:=1 ;
  while i<0.9    do
    begin
      i:=i+0.1;
      PaintBox1.Canvas.MoveTo((x0-4),(y0-kpy*n));
      PaintBox1.Canvas.LineTo((x0+4),(y0-kpy*n));
```

```
      PaintBox1.Canvas.TextOut(x0-20, (y0-kpy*n),FloattoStr(i));
      n:=n+1
    end;
i:=0;
n:=1;
  while i<7    do
    begin
    i:=i+1;
    PaintBox1.Canvas.MoveTo((x0+kpx*n),(y0-4));
    PaintBox1.Canvas.LineTo((x0+kpx*n),(y0+4));
    PaintBox1.Canvas.TextOut((x0+kpx*n), (y0+5),FloattoStr(i));
    n:=n+1
    end;
PaintBox1.Canvas.Pen.Width:=2;
PaintBox1.Canvas.MoveTo(x0, y0);
begin
  t:=0;
  for j := 1 to StringGrid1.RowCount-1 do
    begin
     t:=t+1;
     StringGrid1.Cells[0,j]:=floattostr(t);
    end;
for j := 1 to StringGrid1.RowCount-1 do
  begin
  Randomize();
  t:= Random(10)/10;
  StringGrid1.Cells[1,j]:=floattostr(t);
  StringGrid1.Cells[2,j]:=floattostr(1-t);
      end;
```

```
PaintBox1.Canvas.Pen.Color:=clblue;
for j := 1 to StringGrid1.RowCount-1 do
 begin
   yd:=strtofloat(StringGrid1.Cells[1,j]);
   xd:=strtofloat(StringGrid1.Cells[0,j]);
   x1:=round(xd*kpx+ofs);
   y1:=round(y0-yd*kpy*10);
   PaintBox1.Canvas.LineTo(x1,y1);
 end;
PaintBox1.Canvas.MoveTo(x0, y0-kpy*10);
PaintBox1.Canvas.Pen.Color:=clRed;
for j := 1 to StringGrid1.RowCount-1 do
 begin
   yd:=strtofloat(StringGrid1.Cells[2,j]);
   xd:=strtofloat(StringGrid1.Cells[0,j]);
   x1:=round(xd*kpx+ofs);
   y1:=round(y0-yd*kpy*10);
   PaintBox1.Canvas.LineTo(x1,y1);
 end;
end;
   end;
procedure TForm5.FormActivate(Sender: TObject);
begin
with StringGrid1 do begin
FixedCols:=0;FixedRows:=1;
ColCount:=3;
Cells[0,0]:='Universum';
Cells[1,0]:='U[A]-Blue';
Cells[2,0]:='U[-A]-Red';
```

```
end;
PaintBox1.Refresh;
Randomize();
v:= Random(3);
PaintBox1.Canvas.Brush.Color:=clBtnFace;
PaintBox1.Canvas.FillRect(PaintBox1.BoundsRect);
PaintBox1.Canvas.Pen.Width:=3;
x := PaintBox1.Clientwidth;   //ширина пейнт бокса
y := PaintBox1.ClientHeight;  //высота пейнт бокса
x0 := ofs;   // новые координаты x
y0 := y-ofs; // новые координаты y
xmax:= x-ofs ;
ymax:= ofs  ;
PaintBox1.Canvas.MoveTo(x0, y0); // ось ox
PaintBox1.Canvas.LineTo(xmax,y0);
PaintBox1.Canvas.MoveTo(x0, y0); // ось oy
PaintBox1.Canvas.LineTo(x0,ymax);
PaintBox1.Canvas.TextOut(10, y-15,'(0,0)');
PaintBox1.Canvas.TextOut(x-40, y-15,'X');
PaintBox1.Canvas.TextOut(0+10, 25,'Y');
kpx:= (xmax - x0)div 8;
kpy:= (y0 - ymax)div 11;
i:=0;
n:=1 ;
  while i<0.9    do
    begin
      i:=i+0.1;
      PaintBox1.Canvas.MoveTo((x0-4),(y0-kpy*n));
      PaintBox1.Canvas.LineTo((x0+4),(y0-kpy*n));
```

```
      PaintBox1.Canvas.TextOut(x0-20, (y0-kpy*n),FloattoStr(i));
      n:=n+1
    end;
i:=0;
n:=1;
  while i<7    do
    begin
    i:=i+1;
    PaintBox1.Canvas.MoveTo((x0+kpx*n),(y0-4));
    PaintBox1.Canvas.LineTo((x0+kpx*n),(y0+4));
    PaintBox1.Canvas.TextOut((x0+kpx*n), (y0+5),FloattoStr(i));
    n:=n+1
    end;
  end;
procedure TForm5.Button2Click(Sender: TObject);
begin
 Close;
end;
end.
```