

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет радиофизики и электроники

Кафедра системного анализа

**РАЗРАБОТКА WEB-ОРИЕНТИРОВАННОЙ  
ИНФОРМАЦИОННО-СПРАВОЧНОЙ СИСТЕМЫ  
ПОДДЕРЖКИ ДИСТАНЦИОННОГО ОБУЧЕНИЯ ПО  
СПЕЦИАЛЬНЫМ ДИСЦИПЛИНАМ**

Дипломная работа студента 5 курса  
ПОПЛЕТЕЕВА Андрея Михайловича

Руководитель: ст. преподаватель  
ЛУТКОВСКИЙ Владимир Михайлович

Рецензент: доцент  
МАЛЫЙ Сергей Владимирович.

Допустить к защите  
зав. кафедрой

д-р физ.-мат. наук \_\_\_\_\_ /АПАНАСОВИЧ В. В./

« \_\_\_\_ » \_\_\_\_\_ 2004 г.

МИНСК 2004

## Содержание

Содержание.....	2
Список сокращений .....	3
Введение.....	4
1. Анализ систем дистанционного обучения .....	5
1.1. Актуальность дистанционного обучения.....	5
1.2. Обзор существующих систем.....	8
1.2.1. Прометей .....	8
1.2.2. e-University .....	10
1.2.3. ANNET.....	13
1.3. Перспективы развития ДО.....	15
1.4. Выводы .....	16
2. Разработка информационно-справочной системы.....	18
2.1. Объектно-ориентированный анализ и проектирование .....	18
2.2. Методология Rational Unified Process .....	20
2.2.1. Потоки работ RUP и диаграммы UML.....	21
2.3. Выбор средств разработки.....	24
2.4. Технологии Java Servlets и Java Server Pages.....	26
2.5. Анализ требований к системе.....	27
2.6. Структура системы.....	30
2.6.1. Структура БД .....	31
2.6.2. Менеджеры.....	33
2.6.3. Процессы .....	34
2.7. Взаимодействие компонентов системы .....	35
2.8. Пользовательский интерфейс.....	36
2.8.1. Главное и контекстное меню.....	38
2.8.2. Разделы, статьи и закладки.....	39
2.8.3. Пользовательские отзывы.....	40
2.8.4. Глоссарий .....	40
2.8.5. Поиск .....	40
2.8.6. Присоединяемые файлы .....	39
2.9. Выводы .....	41
3. Учебно-методический комплекс «Нейронные сети» .....	42
3.1. Структура УМК .....	42
3.2. Демонстрационные апплеты .....	43
3.2.1. Искусственный нейрон .....	44
3.2.2. Обучение персептрона .....	44
3.2.3. Многослойный персептрон .....	45
3.2.4. Аппроксимация функций.....	45
3.2.5. Аппроксимация РБФ-сетью.....	46
3.2.6. Прогнозирование временного ряда .....	46
3.2.7. Распознавание символов.....	47
3.2.8. Ассоциативная память .....	47
3.2.9. Самоорганизующаяся сеть .....	48
3.3. Выводы .....	48
Заключение .....	49
Список использованных источников .....	50

## Список сокращений

- ДО – дистанционное обучение
- СДО – система дистанционного обучения
- ИКТ – информационно-коммуникационные технологии
- ИНС – искусственные нейронные сети
- УМК – учебно-методический комплекс
- ЭУ – электронный учебник
- ООП (ООР) – объектно-ориентированное программирование (object-oriented programming)
- ООА – объектно-ориентированный анализ (object-oriented analysis)
- ООД – объектно-ориентированное проектирование (object-oriented design)
- RUP – Rational Unified Process
- ОС – операционная система
- JVM – виртуальная java-машина (Java Virtual Machine)
- JSP – Java Server Pages
- БД – база данных
- СУБД – система управления базами данных
- MVC – Модель-Вид-Контроллер (Model-View-Controller)

## Введение

Развитие наукоемких производств во всем мире увеличивает потребность в высококвалифицированном персонале, а быстрое моральное старение технологий требует от специалистов способности быстро осваивать новые технологии. Все эти процессы отражаются в повышении требований к системе образования. Возникает необходимость в постановке и решении задачи эффективного обучения.

Окинавская Хартия глобального информационного общества, принятая на заседании лидеров восьми наиболее развитых стран мира в 2002 г., в своем первом пункте констатировала, что информационно-коммуникационные технологии (ИКТ) являются одним из наиболее важных факторов, влияющих на формирование общества XXI века. В Хартии отмечается, что наиболее приоритетной областью развития общества является разработка инновационных подходов в образовании, включая дистанционное обучение (ДО). [1]

Несмотря на разнообразие предлагаемых зарубежными и отечественными университетами дистанционных курсов, существуют слабо заполненные или практически пустые ниши в преподавании некоторых областей знаний. В частности, это относится к специальным дисциплинам, таким, например, как искусственные нейронные сети (ИНС).

Интернет-учебников по ИНС существует достаточно много, но подавляющее большинство из них являются простым переводом традиционных лекционных курсов в электронную форму и практически не используют расширенные интерактивные возможности новых технологий [2,3,4]. В то же время, существуют материалы, содержащие мультимедиа-данные и интерактивные элементы, но они в основном относятся к какой-то одной области применения ИНС, разрознены и не позволяют получить полного представления о разнообразии и универсальности нейросетевых технологий. [5]

Цель настоящей работы состояла в развитии существующих методов и инструментальных средств разработки учебных курсов и (дистанционного) обучения, направленных на повышение качественного уровня образования за счет более активного использования современных технологий.

Для достижения этой цели ставились следующие задачи:

- спроектировать и реализовать универсальную информационно-справочную систему поддержки дистанционного обучения по специальным дисциплинам, которая позволит в полной мере использовать при обучении преимущества новых технологий;
- адаптировать и усовершенствовать существующие материалы УМК «Нейронные сети» для использования в дистанционном обучении.

Актуальность работы обусловлена следующим. Современный уровень развития общества повышает требования к системе образования. Основными направлениями развития обучения являются индивидуализация, учет специфики личностного развития обучаемого, вовлечение его в самостоятельную целенаправленную деятельность, применение полученных знаний для решения практических задач. Эффективность такого обучения еще более возрастает при использовании информационных технологий.

Следует отметить, что дистанционное обучение не является альтернативой традиционных форм обучения. Дистанционные технологии дополняют и расширяют возможности получения образования, делают процесс обучения более информативным и гибким, обеспечивают реализацию провозглашенного в XXI веке принципа обучения через всю жизнь. [6]

# 1. Анализ систем дистанционного обучения

## 1.1. Актуальность дистанционного обучения

Значительный прогресс информационных технологий, наблюдаемый в настоящее время, во многом обусловил появление и активное развитие процессов глобализации. Стратегическим направлением развития образования в мире является внедрение систем, способных к интеграции в глобальное образовательное пространство.

Кроме того, развитие современного общества характеризуется такими явлениями, как:

- рост наукоемких производств, требующий работу высококвалифицированного персонала;
- постоянное увеличение объема научно-технической информации, требующее от квалифицированного специалиста способностей и навыков самообразования, умения включаться в непрерывный процесс повышения квалификации;
- быстрая смена технологий, вызывающая достаточно быстрое моральное старение производственных мощностей, требует от специалиста хорошей фундаментальной подготовки, способности быстро осваивать новые технологии;
- рост числа людей, вовлеченных в научную и наукоемкую деятельность, что требует от специалистов знания методологии научной и практической деятельности. [7]

В основе современной парадигмы образования лежит личностно-ориентированная педагогика, базирующаяся на моральном и интеллектуальном развитии человека, вовлечении его в разнообразную самостоятельную целенаправленную деятельность. Эффективность приемов такой педагогики еще более возрастает с использованием информационно-коммуникационных технологий (ИКТ), учащиеся получают возможность совместной работы над проектами (локализация партнера при этом не имеет значения), доступа к информационным банкам не только своей школы или ВУЗа, но и к другим источникам в стране и за рубежом. Применение ИКТ в учебном процессе стимулирует мотивацию к самостоятельной познавательной деятельности учащихся в группах и индивидуально. [8]

Дистанционное обучение (ДО) – это способ обучения на расстоянии, при котором преподаватель и обучаемые физически находятся в различных местах. Исторически, дистанционное обучение означало заочное обучение. Однако сейчас – это средство обучения, использующее аудио, видео и компьютерные системы, связанные через каналы связи. Дистанционное обучение является формой получения образования, наряду с очной и заочной, при которой в образовательном процессе используются лучшие традиционные и инновационные методы, средства и формы обучения, основанные на компьютерных и телекоммуникационных технологиях. [9]

Дистанционное образование тесно связано с дистанционным обучением. Принято считать, что дистанционное образование – это процесс передачи знаний (за него ответственен преподаватель и учебный центр), а дистанционное обучение – это процесс получения знаний (за него ответственен обучаемый). [9]

К достоинствам дистанционного образования можно отнести:

- Гибкость. Каждый может учиться столько, сколько ему лично необходимо для освоения курса, дисциплины и получения необходимых знаний по выбранной специальности. Кроме того, больше времени уделяется самостоятельной работе, а, следовательно, приобретаемые знания и навыки усваиваются лучше.
- Модульность. В основу программ дистанционного образования закладывается модульный принцип. Это позволяет из набора независимых учебных курсов формировать учебный план, отвечающий индивидуальным или групповым потребностям.

- Параллельность. Обучение может проводиться при совмещении с основной профессиональной деятельностью, т.е. «без отрыва от производства».
- Дальнодействие. Расстояние от места нахождения обучающегося до образовательного учреждения не является препятствием для эффективного образовательного процесса.
- Оперативный контроль над качеством полученных знаний сразу после освоения материала (online-тесты).
- Асинхронность. В процессе обучения обучающий и обучаемый могут участвовать по удобному для каждого расписанию и в удобном темпе.
- Охват. Эту особенность иногда называют «массовостью». Количество обучающихся в СДО не является критичным параметром.
- Рентабельность. Средняя оценка зарубежных и отечественных образовательных СДО показывает, что они обходятся приблизительно на 10-50% дешевле. [9]

Необходимость и возможность организации СДО обусловлена следующими факторами:

- Ограничения по получению профессионального образования (в частности, ограниченная доступность, высокая стоимость обучения, ограничения по времени обучения и др.);
- Ограничения по пропускной способности вузов, факультетов повышения квалификации и образовательных учреждений других типов;
- Увеличение количества желающих получить профессиональное образование из-за повышения престижа образования и необходимости переподготовки специалистов в связи с внедрением на сетях современного телекоммуникационного оборудования;
- Появление и развитие качественно новых средств информационных технологий и ярко выраженный процесс информатизации телекоммуникационных технологий. [9]

Появление дистанционного обучения не было обусловлено возникновением Интернет. Сам термин «Дистанционное обучение» возник на десятки лет раньше, во времена распространения радио, телевизионных и телефонных систем. Они позволили большему количеству людей слушать лекции. Студенты и преподаватели могли общаться по телефону. Телефонные конференции также применялись для ведения лекций и дискуссий. Дальнейшее развитие коммуникационных технологий привело к созданию web-ориентированных курсов, которые содержат чаты, форумы, аудио- и видеоконференции и которые позволили приблизить уровень взаимодействия «студент–преподаватель» к наблюдаемому в обычной аудитории. Асинхронный доступ к информации посредством Интернет позволяет получать образование когда угодно, где угодно. [11]

Схема развития образовательных технологий согласно [6] изображена на рис. 1.

Сегодня многие зарубежные университеты предоставляют возможность дистанционного обучения. Некоторые из них позволяют студенту пройти полный цикл обучения, сдать экзамены и получить сертификат или диплом, практически не вставая из-за своего домашнего компьютера. [10]

Британский Открытый Университет был пионером в этой области и до сих пор занимает видное место по исследованиям в области ДО. Австралийские университеты, например, Deacon University, используют Интернет, чтобы вывести свои учебные программы на мировой рынок.

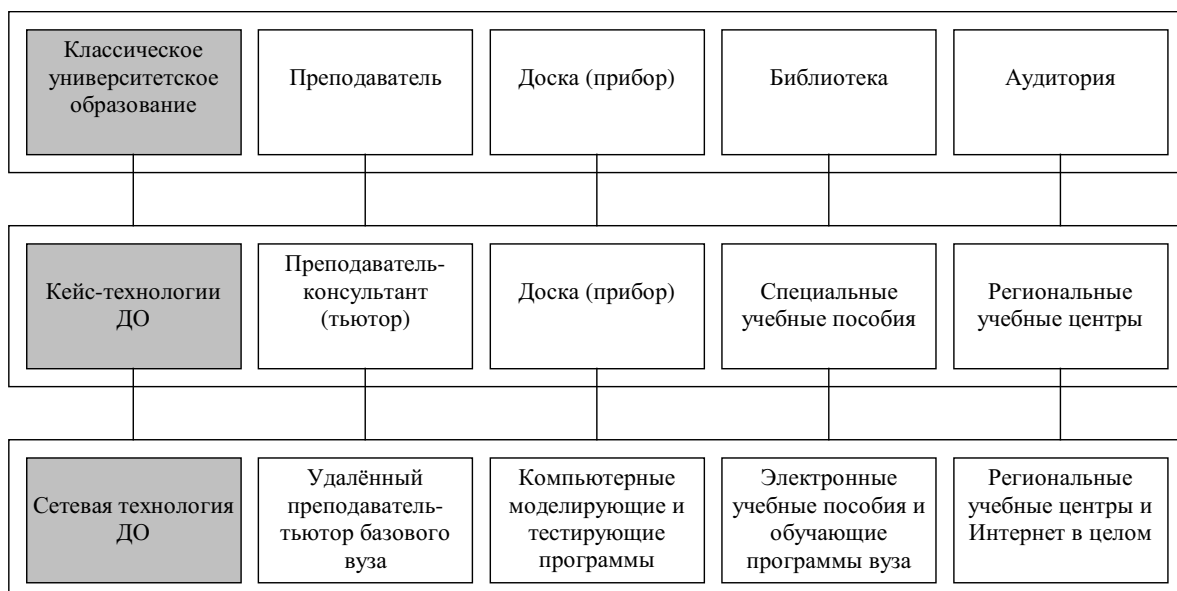


Рисунок 1. Эволюция образовательных технологий

В США с начала 90-х гг., фондом Sloan Foundation было выдано 136 грантов суммой \$30 миллионов на развитие асинхронных образовательных сетей (Asynchronous Learning Networks, ALN). [11]

Европейские университеты также предлагают студентам большое количество учебных программ ДО. Так, в Упсальском университете (Швеция) в 1999г. дистанционно обучались 5 тыс. студентов, им было предложено 115 различных курсов. [12]

В Республике Беларусь технология ДО получила распространение лишь в конце 90-х гг. на уровне инициатив отдельных вузов, образовательных и научных центров. Вместе с тем нельзя не отметить, что за последние 5–7 лет в РБ наметился определенный прорыв в рассматриваемом направлении. Белорусскими вузами предпринят ряд шагов (организационных, технологических, научно-методических) по развитию дистанционного обучения в стенах высшей школы. [6]

В 2002 году Белорусский государственный университет информатики и радиоэлектроники выступил с инициативой об открытии в вузе в качестве эксперимента дистанционной формы обучения. Инициатива была поддержана Министерством образования.

Проведению эксперимента предшествовала разработка ряда документов, организационные мероприятия. Был проведен анализ специальностей, по которым возможна организация дистанционной формы обучения.

Оптимальной была признана организация ДО для специальностей, имеющих меньшее количество сложных лабораторных практикумов, к которым можно отнести специальности информационного и экономического направления. Поэтому в БГУИР в 2002 г. было принято решение об организации дистанционного обучения на специальностях «Программное обеспечение информационных технологий» и «Информационные технологии и системы (в экономике)», а с 2003 г. ведется обучение еще по пяти специальностям: «информационные технологии и управление в технических системах», «информатика», «автоматизированные системы обработки информации», «искусственный интеллект», «маркетинг». [1]

Важным шагом в улучшении телекоммуникационной связи стало создание корпоративной телекоммуникационной сети для учреждений образования и науки UNIBEL, а также научно-исследовательской компьютерной сети (НИКС).

## **1.2. Обзор существующих систем**

### **1.2.1. Прометей**

Система «Прометей» - это программная оболочка, которая не только обеспечивает дистанционное обучение и тестирование слушателей, но и позволяет управлять всей деятельностью виртуального учебного заведения, что способствует быстрому внедрению дистанционного обучения и переходу к широкому коммерческому использованию.

В настоящее время система дистанционного обучения «Прометей» используется различными учебными заведениями и корпорациями из России и стран СНГ. В Беларуси система используется в БГУИР, в рамках эксперимента по внедрению ДО. Интерфейс переведен на несколько национальных языков, среди которых русский, украинский, казахский, узбекский (латиница и кириллица) и английский.

Система дистанционного обучения «Прометей» имеет сертификат Министерства образования РФ о соответствии требованиям, предъявляемым к системам дистанционного обучения.

К достоинствам СДО «Прометей» 4.0 можно отнести следующее [10].

#### ***Общие***

- 1) универсальная оболочка, позволяет проводить дистанционное обучение по любым областям знаний;
- 2) простота освоения, невысокие требования к клиентским компьютерам;
- 3) весь функционал системы доступен из браузера;
- 4) на компьютер пользователя никакого дополнительного ПО устанавливать не нужно;
- 5) позволяет использовать без доработок электронные курсы любых форматов;
- 6) неограниченное количество клиентских мест;
- 7) высокая производительность и масштабируемость по мере увеличения числа пользователей;
- 8) возможность объединения нескольких систем в единую образовательную среду.

#### ***Организация обучения***

- 1) регистрация на курсы по типу электронного магазина;
- 2) календарные планы изучения курсов;
- 3) новый принцип организации учебно-методических материалов - к курсу можно прикрепить любое число электронных книг;
- 4) гибкая подсистема учета платежей (расходов);
- 5) подсистема регистрации/выдачи сертификатов;
- 6) студент может входить в любое количество групп с одним логином;
- 7) возможность сочетания ролей (тьютор может одновременно быть и организатором);
- 8) история взаимодействия со слушателем, заполняемая организатором;
- 9) программы обучения, объединяющие несколько курсов;
- 10) тотальный контроль деятельности участников учебного процесса;
- 11) автоматизация выполнения административных операций через веб-интерфейс.

#### ***Учебный процесс***

- 1) календарные планы изучения курсов;
- 2) мощнейшая подсистема тестирования:
  - тесты для самопроверки и экзаменационные;
  - 10 типов вопросов;
  - коэффициенты сложности вопросов и правильности ответов;
  - произвольный порядок прохождения вопросов в тестовом задании;
  - работа над ошибками.



- 3) развитые средства общения:
- объявления;
  - обмен файлами;
  - обмен почтовыми сообщениями;
  - форумы;
  - чат;
  - книга отзывов;
- дизайнер тестов - верх удобства и функциональности;
  - организация работы в малых группах (In-Team Works);
  - отчеты об активности слушателей позволяют тьютору строить взаимоотношения с ними на индивидуальной основе;

Система «Прометей» имеет модульную архитектуру. Это обеспечивает легкое расширение, модернизацию и масштабирование системы. Ниже перечислены основные модули системы и их назначение.

1. **Подсистема регистрации.** Позволяет просматривать подробную информацию по предлагаемым курсам и программам обучения, формировать заказ и отправлять на обработку;
2. **Подсистема заказов.** Позволяет просматривать информацию о заказе, зачислять слушателей в группы;
3. **Подсистема платежей.** Позволяет регистрировать и обрабатывать поступающие платежные документы;
4. **Подсистема управления группами.** Позволяет выполнять административные операции на уровне групп, что существенно облегчает управление учебным процессом, в который вовлечены большие потоки слушателей;
5. **Подсистема библиотеки.** Позволяет хранить учебные пособия в любом файловом формате, закреплять их за определенными курсами, производить полнотекстовый поиск, собирать статистику обращений слушателей;
6. **Подсистема календарного плана.** Позволяет создавать план-график изучения курса, включая сроки проведения мероприятий, тип, возможность оценки и другие параметры;
7. **Подсистема тестирования.** Реализует проверку учебных достижений слушателей, предусматривает режим работы над ошибками;
8. **Подсистема обмена информацией.** Обеспечивает общение между участниками учебного процесса;
9. **Подсистема администрирования.** Автоматизирует создание и сопровождение объектов системы дистанционного обучения «Прометей»;
10. **Подсистема Мультимедиа-сервер.** Позволяет передавать по сети потоковое видео/аудио в «прямой» трансляции или по запросу;
11. **Подсистема отчетов.** Накапливает статистику по учебному процессу и отображает ее в виде специализированных отчетов;
12. **Подсистема Дизайнер тестов.** Позволяет создавать и редактировать тесты.
13. **Подсистема контроля.** Позволяет преподавателю отслеживать обращения слушателей к учебным материалам.

Внешний вид интерфейса системы приведен на рис 2.

Для реализации системы использована технология Active Server Pages, разработанная компанией Microsoft. К сожалению, эта технология работает только под ОС Windows, что накладывает дополнительные ограничения на серверную часть.

Требуемые аппаратные и программные ресурсы приведены в таблице 1.

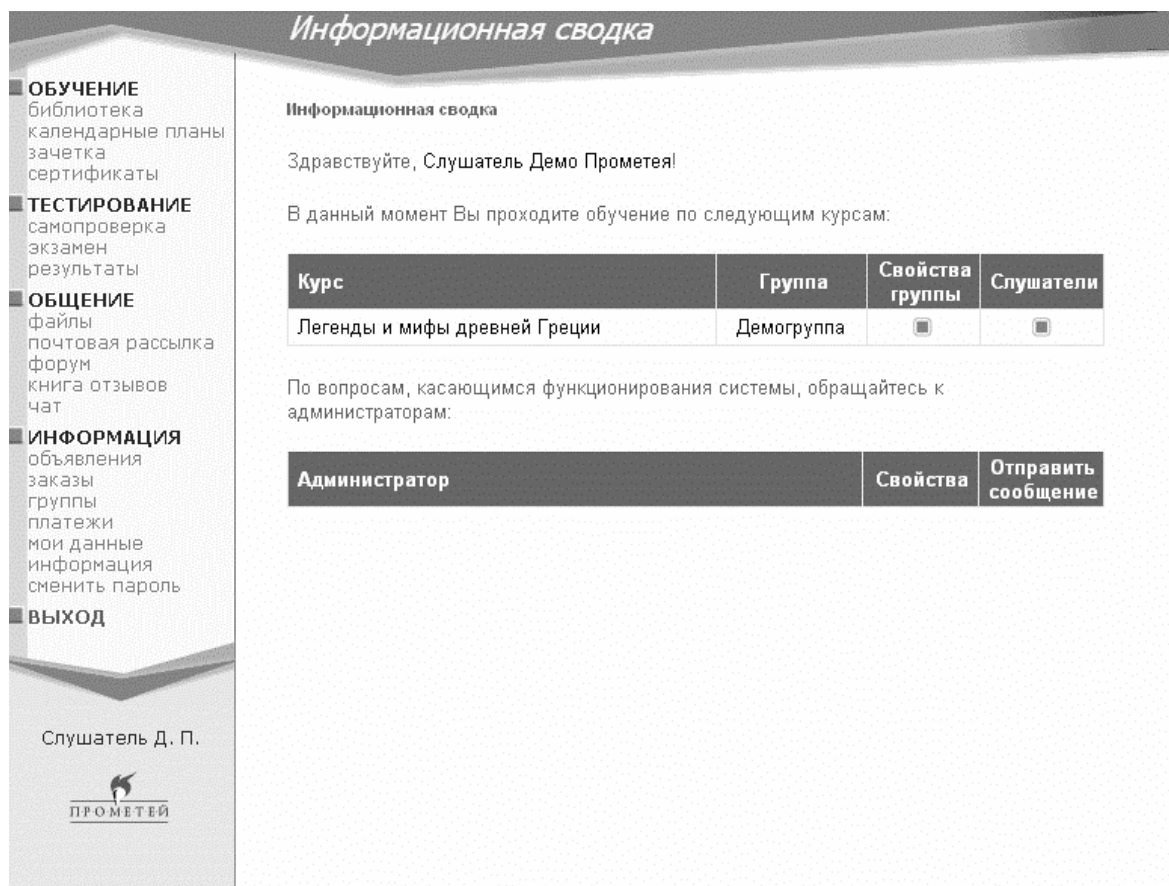


Рисунок 2. Интерфейс студента (слушателя) СДО «Прометей»

Таблица 1. Системные требования СДО «Прометей» 4.0

Ресурс	Сервер		Клиент	
	Минимум	Рекомендуется	Минимум	Рекомендуется
Процессор	Pentium 166 МГц	от Pentium III 800 МГц	от Pentium 100	от Celeron 600
ОЗУ	128 Мб	от 512 Мб	16 МБ	от 64 МБ
Видео	-	-	SVGA, 1МБ	от AGP 8 МБ
Диск	4 Гб	от 18 Гб, SCSI	-	-
CD-ROM	4x	от 40x	4x	от 4x
Интернет	от 64 кбит/с	от 128 кбит/с	от 14.4 кбит/с	от 64 кбит/с
ПО	<ul style="list-style-type: none"> <li>MS Windows 2000 Server</li> <li>MS SQL Server 2000</li> </ul>		MS Internet Explorer 4.0 или выше	

### 1.2.2. e-University

СДО «e-University» была спроектирована и реализуется на кафедре информационного и программно-математического обеспечения автоматизированных производств БГУ. К работе над графическим дизайном, тестированием, написанием документации (руководств пользователей), администрированием системы привлекаются специалисты соответствующих секторов отделения Интернет-технологий компании ИВА. [13, 14]

СДО «e-University» предназначена для решения следующих задач:

- Дистанционное образование: организация обучения, консультирования и тестирования неограниченного числа слушателей посредством Internet; сочетание традиционных педагогических методов с новейшими коммуникационными и мультимедийными технологиями.

- Организация учебного процесса традиционного образования: эффективное взаимодействие преподавателей и учащихся в удобное для каждого время; самостоятельная подготовка студентов; тестирование и автоматическая оценка знаний; контроль организации обучения и его эффективности.
- Корпоративное образование: обучение и тестирование квалификационных знаний персонала и соискателей на работу; поддержка концепции непрерывного образования и повышения квалификации; эффективное взаимодействие удаленных филиалов.[15]

СДО «e-University» предоставляет необходимый набор средств для реализации концепции дистанционного обучения, взаимодействия управления, преподавания и обучения. Система может использоваться как в аудитории, так и удаленно.

В первой реализации приложение включает в себя модули по формированию и изменению групп, для которых определяются изучаемые дисциплины и контрольные работы для выполнения. Включены модули, обеспечивающие организацию консультаций с преподавателем, рейтингового голосования для определения отношения обучаемых к постановке и проведению того или иного курса. [13]

Предметами и курсами системы дистанционного обучения управляет администратор, он же назначает преподавателей и студентов на соответствующие предметы курсов. Когда преподаватель назначен на предмет курса, он получает право на наполнение последнего тестами и пособиями. Когда студент назначен на предмет, он получает доступ к тестам и пособиям предмета. При регистрации предмета также надо указать систему оценки по предмету (n-балльную), т.е. зачет/незачет – двухбалльная и т.д. Оценка по предмету вычисляется автоматически в зависимости от веса тестов и системы оценки. [13]

Тестированием закрепляется пройденный материал и осуществляется текущий рубежный итоговый контроль. Тесты могут иметь различные параметры: предварительно сформированную преподавателем систему оценок теста, время выполнения теста, порядок появления вопросов при прохождении теста (случайный или последовательный), способ доступа к вопросам (последовательный – вопросы появляются один за другим при невозможности возврата назад; произвольный – все вопросы доступны в любой момент времени). При прохождении теста можно прерваться, но при этом фиксируется ответ на последний открытый вопрос, т.е. при возобновлении тестирования тест начнется со следующего вопроса. Имеется возможность копирования и вывода на печать. [13]

Статистика позволяет познакомиться с результатами тестирования: индивидуальными, групповыми, по курсу, по предмету, по тесту, по вопросу и т.д. Незарегистрированный пользователь не имеет доступа в раздел. Обучающийся может смотреть подробную статистику своих результатов тестирования и общую статистику тех, кто обучается вместе с ним по одному предмету. Преподаватель может просматривать подробную статистику обучающихся по его предмету и общую статистику по учебному курсу, которому принадлежит его предмет. Администратор может просматривать подробную статистику всех преподавателей и учащихся на всех курсах.

Основными структурными единицами учебного материала является любой заархивированный файл: документ с гипертекстом и графическими иллюстрациями, аудио-видео материалы. Представление материалов лежит на преподавателе, который самостоятельно решает, как его организовывать.

Кроме того, имеется возможность помещать дополнительный материал и исправлять уже существующий. Есть справочная информация для преподавателя, обучаемого и администратора. Существует возможность сопровождать текстовые фрагменты аудио- или видеоинформацией: преподаватель сам готовит пособие, включающее аудио- или видеоинформацию, а система предоставляет обучаемому его скопировать и изучить самостоятельно. Можно получить электронный вариант статей или книг, они доступны для копирования.

Система не создает ограничений к структуре курса и учебных материалов к нему. Рекомендуется разделять курс по разделам и создавать к ним контрольные тесты, чтобы после изучения каждого учебного пособия (раздела, главы), заявленного как необходимые, обучаемый проходил тест. В определенных пределах возможно изменение структуры курса. В то же время, при создании учебной программы необходим опыт преподавания конкретного предмета.

Обучаемые должны владеть компьютером на уровне обычного пользователя. Система дистанционного обучения «e-University» снабжена руководствами пользователя (студента, преподавателя, администратора), где пошагово описана последовательность действий для работы в системе. Обучаемые должны иметь доступ в Internet или локальную сеть, в которую входит сервер системы.

Как и в большинстве СДО, для работы с системой пользователю достаточно обычного интернет-браузера (Internet Explorer 5.0 и выше). Программа поддерживает спецификацию IMS, т.е. обучение и подачу материала в зависимости от способностей, результатов обучаемого. Существует возможность работать с системой в многооконном режиме, т.е. привычным для web-пользователей образом.

К достоинствам системы можно отнести следующее.

- Цена ниже российских аналогов при включении широкого перечня бесплатных услуг;
- Соответствует спецификациям по стандартизации образовательной деятельности международного образовательного консорциума (IMS, модель LTSA);
- Для функционирования не требуется закупка и установка лицензионных программных продуктов;
- Учтены требования и пожелания крупнейших ВУЗов Беларуси;
- Дружественный русскоязычный web-интерфейс.

Для реализации системы использован язык программирования Java, технологии Apache Jakarta Struts 1.1, JSP 1.2, Java Servlet 2.3 и бесплатная СУБД MySQL.

Внешний вид интерфейса системы приведен на рис 3, системные требования – в таблице 2.



Рисунок 3. Интерфейс администратора СДО «e-University»

Таблица 2. Системные требования СДО «e-University»

	Сервер	Клиент
Процессор	Intel 400Mhz и выше	Intel 133Mhz и выше
ОЗУ	не менее 128Mb (рекомендуется 256Mb)	не менее 32Mb
Операционная система	Windows NT/2000/XP или Linux	Windows 95/98/NT/2000/XP
ПО	<ul style="list-style-type: none"> <li>• Microsoft Internet Explorer 5 и выше;</li> <li>• MySQL 2.23 и выше;</li> <li>• Apache Jakarta Tomcat 4.1.18 и выше;</li> <li>• JDK 1.3.1 и выше.</li> </ul>	<ul style="list-style-type: none"> <li>• Microsoft Internet Explorer 5 и выше</li> </ul>

### 1.2.3. ANNET

Дистанционный курс ANNET (Artificial Neural NETWORKs), разработанный профессором Дж. Рингвудом (J.V. Ringwood) из Дублинского университета (Ирландия), предназначен для изучения теории искусственных нейронных сетей (ИНС). [16]

Для использования ИНС необходимо проводить довольно объемные вычисления, что делает первые идеально подходящим предметом для изучения с помощью компьютера.

Курс ANNET содержит следующие части.

- Введение в ИНС (Introduction to ANNs);
- Простые нейронные сети (Simple neural networks);
- Многослойные перцептроны (Multilayer perceptron networks);
- Радиально-базисные сети (Radial basis function networks);
- Динамические нейронные сети (Dynamic neural networks);
- Самоорганизующиеся карты (Self-organizing feature maps);
- Сети Хопфилда (Hopfield networks);
- Предобработка данных (Data preprocessing);
- Применения ИНС (Application of ANNs).

Курс структурирован таким образом, чтобы обеспечить студенту удобство и простоту использования. Материал разбит на главы. При просмотре главы, слева отображается ее оглавление, справа – основной материал. Сверху и снизу находятся кнопки «Назад» и «Далее», которые позволяют последовательно продвигаться по материалу. Для произвольного доступа к параграфам можно использовать оглавление.

В конце каждой главы студенту предлагается тест с несколькими вариантами ответа на каждый вопрос. Тест состоит из кратких вопросов и/или математических выражений, изученных в этой главе. Студентам рекомендуется попробовать пройти эти тесты, т.к. это позволит определить меру своего понимания прочитанного материала.

Для обсуждения вопросов, связанных с курсом, организована почтовая рассылка (mailing list). Всем студентам, проходящим обучение, предлагается подписаться на эту рассылку. Предполагается, что рассылка используется студентами для обсуждения между собой проблем или сложностей, которые могут возникнуть при изучении материала. Рассылка также просматривается преподавателем, задача которого – оказывать помощь и давать пояснения при возникновении общих затруднений.

Курс содержит практические примеры использования ИНС. Большинство примеров предназначены для использования в среде MATLAB (что требует от студента наличия таковой на своем компьютере). После того, как примеры загружены из сети на локальный

компьютер, студент может произвольно изменять значения параметров и наблюдать, как это отражается на работе ИНС. MATLAB-примеры также используются в некоторых тестах.

Навигация по курсу характеризуется следующими элементами.

1. Единообразное расположение элементов, кнопки «Назад/Далее» на каждой странице.
2. Гиперссылки, которые ведут пользователя к
  - предыдущим записям для «освежения» знаний;
  - более подробным объяснениям различных деталей;
  - другим сайтам, непосредственно связанным с изучаемым пунктом. Например, в разделе по самоорганизующимся картам Кохонена существуют ссылки на сайт Тамперского университета (Tampere University), Финляндия [17], который содержит множество различных примеров использования ИНС этого типа.
3. Оглавление с гиперссылками, всегда видимое в левой части окна браузера.

Внешний вид пользовательского интерфейса системы изображен на рис. 4. Вместо просмотра HTML-статей, пользователи могут скачать материалы в формате PostScript.

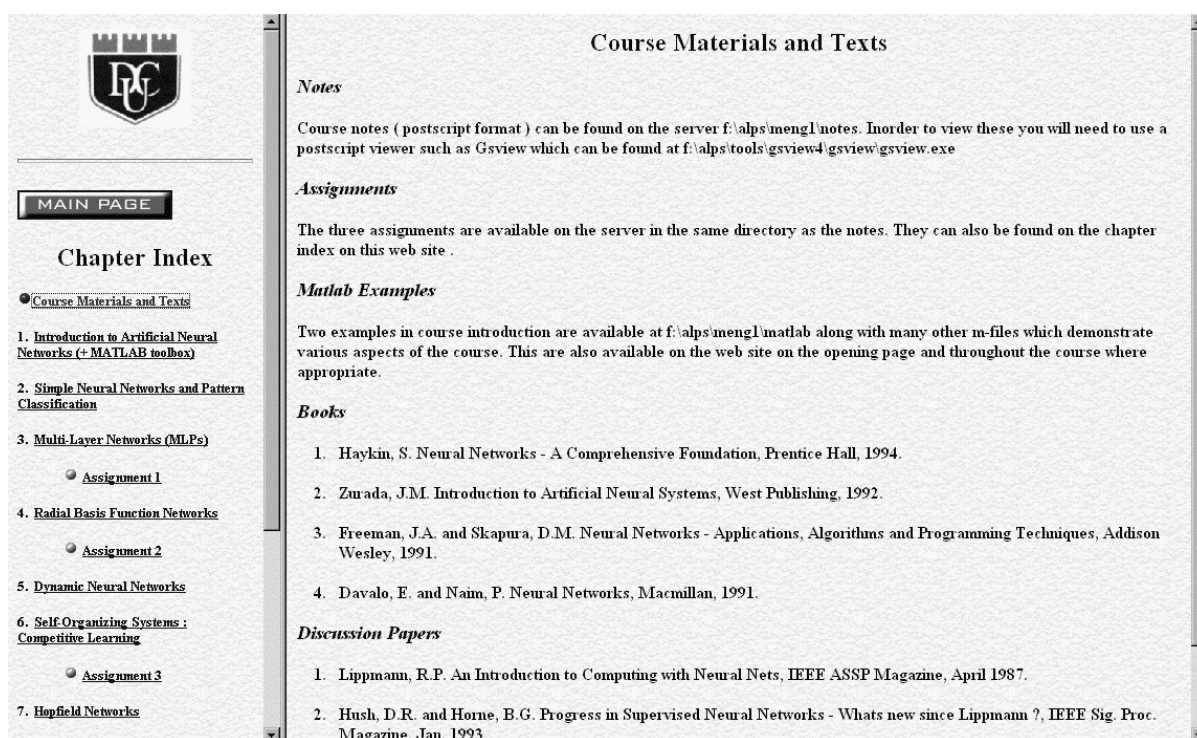


Рисунок 4. Интерфейс курса по ИНС Дж. Рингвуда

Кроме ссылок на другие сайты, части этого курса используются в других курсах, в том числе образовательном курсе по ИНС в США.

Существующие в курсе Java-апплеты располагаются в начале главы, их назначение – предоставлять основную информацию об этой главе. Они не требуют специальных знаний, а предназначены скорее для реализации концепции «делая – видеть» («see by doing»).

Апплеты содержатся в четырех разделах:

- многослойные перцептроны;
- радиально-базисные функции;
- сети Хопфилда;
- самоорганизующиеся карты.

После усвоения материала, студенты должны выполнить контрольное задание. Задания программируются на макроязыке MATLAB. Таким образом, каждый студент должен приобрести экземпляр пакета MATLAB, что накладывает определенные ограничения на доступность курса и может стать причиной уменьшения возможного

количества студентов. Отчеты о контрольных работах отправляются электронной почтой, обычно в формате Microsoft Word или LaTeX (postscript).

В конце каждой главы студенту предлагается пройти тест, состоящий из 10 вопросов, случайно выбранных из общего набора (30 и более вопросов). Ответы на тестовые задания немедленно обрабатываются сервером, и результаты отправляются студенту посредством браузера. Результаты тестов никоим образом не используются для оценки знаний студентов, но статистика прохождения тестов доступна для просмотра преподавателем.

При регистрации каждый студент подписывается на e-mail рассылку. Архивы рассылки хранятся на web-сервере. Таким образом, каждое сообщение, отправленное в рассылку, может быть просмотрено любым участником курса. Эта функция выполняет три предназначения:

1. Архив рассылки служит доской объявлений для публикации новостей о курсе.
2. Общение через e-mail проходит через рассылку, позволяя, таким образом, любому студенту (или преподавателю) почерпнуть полезную информацию из вопросов и ответов.
3. С целью помочь удаленным студентам, в рассылке публикуются еженедельные отчеты об основных пунктах, изученных на лекциях.

Преподавателям также доступна для просмотра статистика по использованию курса и прохождения тестов. Автоматическое ведение статистики реализовано с помощью Perl-скриптов и технологии SSI (server-side includes).

Для отображения статистических данных в удобном табличном виде существует специальная программа. Полученная информация является основой для постоянного изменения материала.

Хотя полезность статистики самими создателями представляется сомнительной, она, тем не менее, свидетельствует, что студенты, которые успешно прошли онлайн-тесты (даже не с первой попытки), показывают лучшие результаты на финальных экзаменах.

К достоинствам системы можно отнести также и то, что курс легко может быть адаптирован к использованию с CD-ROM. В этом случае, доступ к Интернет нужен только для прохождения тестов и чтения почтовой рассылки.

### **1.3. Перспективы развития ДО**

Стремительное увеличение популярности ДО, являющееся следствием быстрого роста Интернета, требует от образовательных учреждений переосмысления векового опыта образования.

На Западе отмечается усиление конкуренции университетов за иностранных студентов [11]. Несомненно, ДО является одной из основных причин этого явления.

Образовательные учреждения постоянно ищут более эффективные пути для оплаты труда преподавателей и развертывания соответствующей инфраструктуры. Один из наиболее популярных способов – максимизация количества студентов в расчете на одного преподавателя. Университеты, следящие за своей репутацией, пользуются консультациями экспертов для того, чтобы небольшие семинары были сбалансированы с большими лекционными курсами.

Это происходило и задолго до возникновения Интернета, до того, как появилось дистанционное обучение. Размер класса, нагрузка преподавателей, учебники, роль библиотеки в поддержке образования – все это предоставляет широкие возможности для организации экономически эффективного учебного процесса. Онлайн-доставка курсов и материалов – это еще одна такая возможность.

Многие специалисты отмечают, что компьютерные технологии имеют особые возможности объяснения понятий посредством устройств, таких, например, как апплеты. Предоставлением студентам непосредственной обратной связи в виде ответов на вопросы, компьютеры также могут заменить преподавателей, которые тратили время в аудитории на тренировку и практику.

Многие компании уже имеют собственные «корпоративные университеты», которые содержат тренинги и учебную информацию для работников. Многие подключаются к таким ресурсам из-за необходимости постоянного обучения на рабочем месте.

Более того, некоторые крупные компании, например, Motorola, предлагают часть своих курсов онлайн. Хотя почти все корпоративные университеты создавались для внутренних пользователей, компании пытаются компенсировать свои инвестиции путем привлечения внешних клиентов. Некоторые эксперты считают, что корпоративные университеты способны стать хорошим дополнением к традиционным образовательным программам обычных университетов, которые не всегда отвечают современным запросам промышленности. [11]

Что касается Республики Беларусь, то в качестве причин, сдерживающих развитие ДО, можно отметить:

- отсутствие достаточного материально-технического обеспечения учреждений образования;
- ограниченные возможности каналов связи;
- отсутствие полноценного правового регулирования процесса развития ДО;
- низкое материальное стимулирование труда преподавателей. [6]

К основным тенденциям дальнейшего развития ДО в Республике Беларусь можно отнести:

- широкое применение Интернет-технологий, сетевых информационных ресурсов;
- создание корпоративных виртуальных библиотек (совместный проект VirLib – БГУИР, БГУ, БГЭУ);
- использование технологий ДО в традиционных формах обучения;
- создание открытых межгосударственных университетов (образовательных порталов). [6]

Перспективным направлением развития систем дистанционного обучения представляется создание их на основе экспертных систем. Такие системы могут аккумулировать знания преподавателей в конкретных предметных областях и тиражировать этот опыт для консультаций обучаемых. База знаний экспертной системы, то есть совокупность знаний определенной предметной области формируется преподавателями, имеющими соответственно достаточно высокую квалификацию [18]. Наличие таких систем позволит восполнить дефицит высококвалифицированных педагогических кадров, а также повысить эффективность обучения за счет более наглядного и простого доступа к знаниям.

Однако следует учитывать, что система дистанционного обучения сможет дать необходимый социальный и экономический эффект лишь при условии, если создаваемые и внедряемые информационные технологии станут не инородным элементом в традиционной системе образования, а будут естественным образом интегрированы в нее. [18]

#### **1.4. Выводы**

В представленном обзоре рассмотрены три СДО, различные по своей сложности, имеющие различную стоимость. Такая дифференциация позволяет четко определить общие требования к СДО. В общем случае, каждая СДО должна иметь функциональность для предоставления удаленным студентам учебного материала (посредством электронного учебника, ЭУ), а также иметь подсистему тестирования знаний.

Способ предоставления студентам учебных материалов для сетевых СДО, в общем, только один – в виде информационного текста, возможно, с включением активных элементов. Поэтому на первое место выходят такие характеристики, как удобство навигации по материалам курса, наличие удобной системы поиска и т.п. Следует заметить, что среди рассмотренных систем поиск по материалам курса реализован только в СДО «Прометей».



Объяснить такое упущение можно, пожалуй, тем фактом, что наибольшее внимание при разработке СДО уделяется подсистеме тестирования. В пользу этого предположения говорит хотя бы то, что тестирование в той или иной мере реализовано в каждой из рассмотренных СДО.

Действительно, задача контроля качества знаний студентов является одной из важнейших задач ДО. Наличие нерешенных проблем в области тестирования знаний удаленных студентов позволяет рассматривать эту задачу как самостоятельную, требующую отдельного особого решения.

В частности, при проведении удаленного тестирования невозможно идентифицировать, кто отвечает на вопросы теста – обучаемый или его более опытный коллега. В настоящее время эта проблема решается путем введения обязательного очного экзамена. Одна из вариаций такого решения – очное прохождение тестов посредством СДО, при этом преподаватель присутствует в аудитории, что гарантирует точную идентификацию студентов. [19]

Основной целью данной работы являлось восполнение описанных недостатков в реализации ЭУ, проявляющихся в неудобстве навигации по курсу, и примитивности или полном отсутствии функций поиска учебных материалов.

Задача тестирования знаний студентов является самостоятельной сложной проблемой, требующей отдельного решения, и в данной работе не рассматривается.

## 2. Разработка информационно-справочной системы

В данной главе описываются технические детали реализации информационно-справочной системы поддержки дистанционного образования: выбор технологий, обоснование выбора, описание структуры системы, назначение и функциональность отдельных ее модулей.

### 2.1. Объектно-ориентированный анализ и проектирование

По исследованиям Standish Group (CHAOS Report 1998), только 26% проектов создания информационных систем заканчиваются успешно. [20]

Причина такого плачевного положения, по мнению некоторых аналитиков, заключается во все возрастающей сложности создаваемых систем. Проявлениями этой сложности являются не только увеличение размеров и функциональности. Не меньше, если не больше, проблем порождается частой сменой потребностей пользователей и ростом требований к качеству систем.

Традиционные способы разрешения проблемы сложности, такие как увеличение коллективов разработчиков, их специализация, распределение работ в чистом виде, ведут к еще большим трудностям согласования результатов и сборки готовых систем.

Серьезным шагом к победе над сложностью явилось появление объектно-ориентированного программирования (ООП, object-oriented programming). [21]

Объектно-ориентированная технология основывается на так называемой объектной модели. Основными ее принципами являются: абстрагирование, инкапсуляция, модульность, иерархичность, типизация, параллелизм и сохраняемость. Каждый из этих принципов сам по себе не нов, но в объектной модели они впервые применены в совокупности.

Объектно-ориентированный анализ и проектирование принципиально отличаются от традиционных подходов структурного проектирования: здесь нужно по-другому представлять себе процесс декомпозиции, а архитектура получаемого программного продукта в значительной степени выходит за рамки представлений, традиционных для структурного программирования. Отличия обусловлены тем, что структурное проектирование основано на структурном программировании, тогда как в основе объектно-ориентированного проектирования лежит методология объектно-ориентированного программирования. [22]

Основная идея объектно-ориентированного анализа и проектирования (OOA & OOD, object-oriented analysis and design) состоит в рассмотрении предметной области и логического решения задачи с точки зрения объектов (понятий и сущностей). В процессе объектно-ориентированного анализа основное внимание уделяется определению и описанию объектов (или понятий) в терминах предметной области. В процессе объектно-ориентированного проектирования определяются логические программные объекты, которые будут реализованы средствами объектно-ориентированного языка программирования. Эти программные объекты включают в себя атрибуты и методы. И, наконец, в процессе конструирования (construction) или объектно-ориентированного программирования (object-oriented programming) обеспечивается реализация разработанных компонентов и классов.

Рассмотрим некоторые основные принципы объектно-ориентированного анализа и проектирования. [23]

- Сначала производится так называемый анализ требований (requirements analysis) во время которого выделяются основные процессы, происходящие в моделируемой системе и их формулировка в виде прецедентов. Прецедент (precedent), или вариант использования (use case) – это текстовое описание процессов, происходящих в предметной области.
- Шаг второй. Объектно-ориентированный анализ предметной области (object-oriented domain analysis). Задача этого шага в определении видов деятельности

участников процесса и составлении концептуальной модели (conceptual model), которая отражает различные категории элементов предметной области. Причем не только виды деятельности участников, но и все относящиеся к делу понятия.

- Шаг третий. На этом шаге основное внимание сосредоточено на распределении обязанностей. Это и называется объектно-ориентированным проектированием (object-oriented design). Распределение обязанностей (responsibility assignment) означает выделение задач и обязанностей различных программных объектов в приложении.

Наиболее важным моментом объектно-ориентированного анализа и проектирования является квалифицированное распределение обязанностей между компонентами программной системы. Это единственный вид деятельности, без которого невозможно обойтись. К тому же он оказывает определяющее влияние на качество системы, ее масштабируемость, расширяемость и возможность повторного использования компонентов. Обязанности объектов и их взаимодействия изображаются с использованием диаграмм классов (design class diagram) и диаграмм взаимодействий (collaboration diagram).

Гради Буч, ставший, фактически, классиком ООА & ООД, объясняет термины объектно-ориентированного подхода следующим образом. [22]

Объектно-ориентированное программирование – это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

В данном определении можно выделить три части:

- 1) ООР использует в качестве базовых элементов объекты;
- 2) каждый объект является экземпляром какого-либо определенного класса;
- 3) классы организованы иерархически.

Программа будет объектно-ориентированной только при соблюдении всех трех указанных требований. В частности, программирование, не основанное на иерархических отношениях, не относится к ООР, а называется программированием на основе абстрактных типов данных.

Программирование прежде всего подразумевает правильное и эффективное использование механизмов конкретных языков программирования. Проектирование, напротив, основное внимание уделяет правильному и эффективному структурированию сложных систем.

Объектно-ориентированное проектирование – это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы. [22]

В данном определении содержатся две важные части: объектно-ориентированное проектирование

- 1) основывается на объектно-ориентированной декомпозиции;
- 2) использует многообразие приемов представления моделей, отражающих логическую (классы и объекты) и физическую (модули и процессы) структуру системы, а также ее статические и динамические аспекты.

Именно объектно-ориентированная декомпозиция отличает объектно-ориентированное проектирование от структурного; в первом случае логическая структура системы отражается абстракциями в виде классов и объектов, во втором – алгоритмами.

Объектно-ориентированный анализ (ООА, object-oriented analysis) направлен на создание моделей реальной действительности на основе объектно-ориентированного мировоззрения. ООА – это методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области.

На результатах ООА формируются модели, на которых основывается OOD; OOD в свою очередь создает фундамент для окончательной реализации системы с использованием методологии OOP. [22]

К сожалению, появление OOP не устранило проблем недостаточного взаимопонимания разработчиков и пользователей, неэффективного управления разработкой в условиях изменяющихся требований, неконтролируемости изменений в процессе выполнения работ, субъективности в оценке качества продуктов разработки и т. д. [21]. Наилучшим решением данной проблемы в настоящее время является разработанная компанией Rational Software методология Rational Unified Process.

## **2.2. Методология Rational Unified Process**

Rational Unified Process (RUP) – одна из лучших методологий разработки программного обеспечения, созданная в компании Rational Software. Основываясь на опыте многих успешных программных проектов, RUP позволяет создавать сложные программные системы, основываясь на индустриальных методах разработки. Одним из основных столпов, на которые опирается RUP, является процесс создания моделей при помощи унифицированного языка моделирования (UML). [24]

RUP ведет свою историю от продуктов Rational Approach и Objectory Process 3.8, объединение которых произошло после слияния в 1995 г. корпораций Rational Software и Objectory AB.

RUP представлен в виде структурированной базы знаний [25, 26], которая представляет собой набор исчерпывающих рекомендаций для создания практически любых программных продуктов [24]. RUP поддерживает технологию разработки программных продуктов для различных платформ (Microsoft .NET, Java2 Enterprise Edition и др.).

Применение унифицированного процесса разработки позволяет уменьшить затраты проекта, уложиться в заданные сроки и повысить качество создаваемого программного продукта [27]. Сейчас больше тысячи компаний воспользовались преимуществами разработки на основе RUP. Унифицированный процесс разработки используется в различных прикладных областях, в больших и малых проектах, что доказывает его универсальность и широкую применимость.

Основными понятиями RUP являются артефакт (artifact) и прецедент (precedent). Артефакты — это некоторые продукты проекта, порождаемые или используемые в нем при работе над окончательным продуктом. Прецеденты — это последовательности действий, выполняемых системой для получения наблюдаемого результата.

Весь процесс разработки программной системы рассматривается в RUP как процесс создания артефактов. Причем то, что попадает в руки конечного пользователя, будь то программный модуль или программная документация, — это один из подклассов всех артефактов проекта.

Основной упор в RUP делается на моделирование разрабатываемой системы. Модели помогают очерчивать как проблему, так и пути ее решения, и создаются они при помощи унифицированного языка Unified Modeling Language (UML), предложенного компанией Rational и впоследствии утвержденного OMG как стандарт. Но еще до этого UML стал стандартом де-факто для описания сложных систем и позволяет разработчикам определять, визуализировать, конструировать и документировать артефакты программных систем.

Это дает возможность пройти через всю последовательность необходимых работ — от общего взгляда на них «с высоты птичьего полета» до создания конкретных артефактов. [27]

Модели позволяют рассмотреть будущую систему, ее объекты и их взаимодействие еще до вкладывания значительных средств в разработку, позволяют увидеть ее глазами будущих пользователей снаружи и разработчиков изнутри еще до создания первой строки исходного кода. Большинство моделей представляются UML диаграммами. [24]

### 2.2.1. Поток работ RUP и диаграммы UML

RUP разделяется на фазы, которые состоят из итераций, поэтому процесс еще называют итеративным и инкрементным. Каждая итерация проходит цикл основных работ и подводит разработчиков к конечной цели: созданию программной системы. В ходе итераций создаются промежуточные артефакты и вариант программной системы, который реализует некоторый набор функций, увеличивающийся от итерации к итерации. Фазы и основные потоки работ процесса показаны на рис. 5, там же приведены примерные трудозатраты работ по фазам. [24]

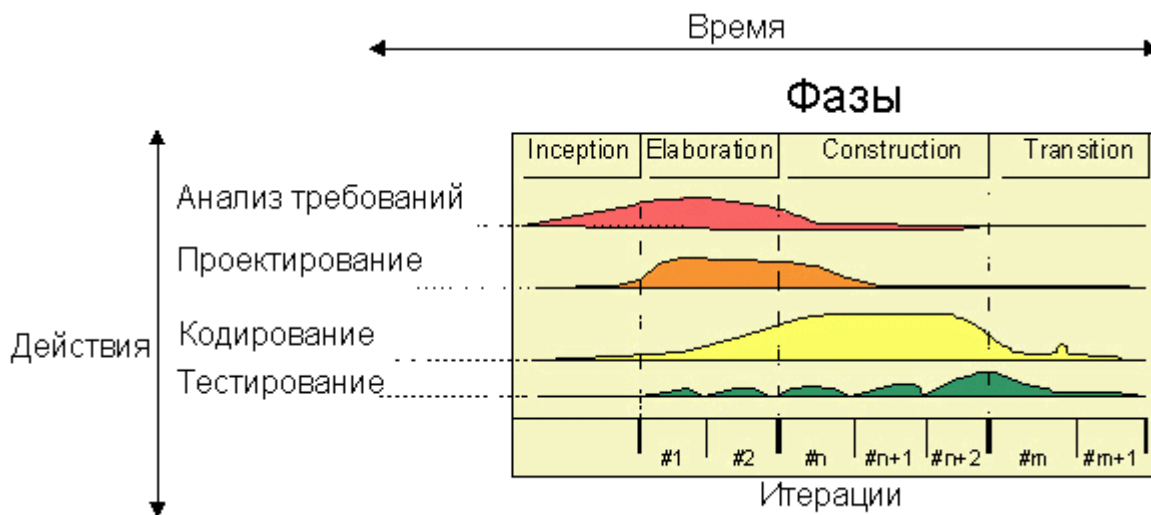


Рисунок 5. Фазы и потоки работ RUP

RUP включает в себя следующие основные потоки:

1. определение требований;
2. анализ;
3. проектирование;
4. реализация;
5. тестирование.

#### 1. Определение требований

Унифицированный процесс – это процесс, управляемый прецедентами, которые отражают сценарии взаимодействия пользователей. Фактически, это взгляд пользователей на программную систему снаружи.

Таким образом, одним из важнейших этапов разработки, согласно RUP, является этап определения требований, который заключается в сборе всех возможных пожеланий к работе системы, которые только могут прийти в голову пользователям и аналитикам. Позднее эти данные должны будут систематизированы и структурированы, но на данном этапе в ходе интервью с пользователями и изучения документов, аналитики должны собрать как можно больше требований к будущей системе, что не так просто, как кажется на первый взгляд. Пользователи часто сами не представляют, что они должны получить в конечном итоге. Для облегчения этого процесса используются диаграммы прецедентов (use case diagram) (рис. 6). [24]



Рисунок 6. Пример диаграммы прецедентов (use case diagram)

Диаграмма представляет собой отражение действующих лиц (актеров, actors), которые взаимодействуют с системой, и реакцию программных объектов на их действия. Актерами могут быть как пользователи, так и внешние агенты, которым необходимо передать или получить информацию. Значок варианта использования отражает реакцию системы на внешнее воздействие и показывает, что должно быть сделано для актера.

Между актерами и вариантами использования могут быть различные виды взаимодействия, которые отображаются различными видами линий и стрелок. Основные виды взаимодействия следующие: простая ассоциация, направленная ассоциация, наследование. Существуют и другие виды взаимодействия, но они применяются реже.

Для детализации конкретного прецедента используется диаграмма Активности (Activity Diagram), пример которой дан на рис 7.

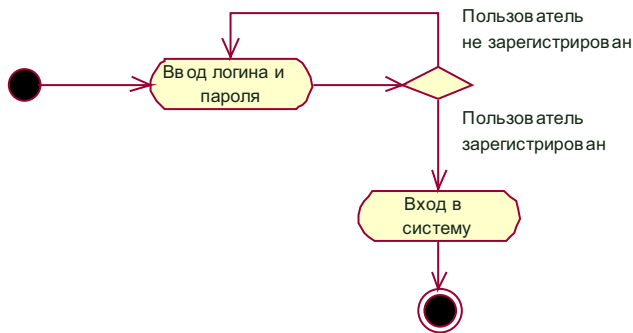


Рисунок 7. Пример диаграммы активности

Простота диаграммы прецедентов позволяет аналитикам легко общаться с заказчиками в процессе определения требований, выявлять ограничения, налагаемые на систему.

Также диаграмма прецедентов может использоваться для создания сценариев тестирования, поскольку все взаимодействие пользователей и системы уже определено.

Для того чтобы верно определить требования, разработчики

должны понимать контекст (часть предметной области) в котором будет работать будущая система. Для этого создается модель предметной области, которая описывает важные понятия, с которыми будет работать система и связи их между собой. Для создания модели предметной области используется обычная диаграмма классов (рис. 10).

## 2. Анализ

После определения требований и контекста, в котором будет работать система, наступает черед анализа полученных данных. В процессе анализа создается аналитическая модель, которая подводит разработчиков к архитектуре будущей системы. Аналитическая модель – это взгляд на систему изнутри, в отличие от модели прецедентов, которая показывает, как система будет выглядеть снаружи. [24]

Эта модель позволяет понять, как система должна быть спроектирована, какие в ней должны быть классы и как они должны взаимодействовать между собой. Основное ее назначение – определить направление реализации функциональности, выявленной на этапе сбора требований и сделать набросок архитектуры системы.

Для детализации модели анализа при помощи UML используются диаграммы сотрудничества (collaboration diagrams) (рис 8).

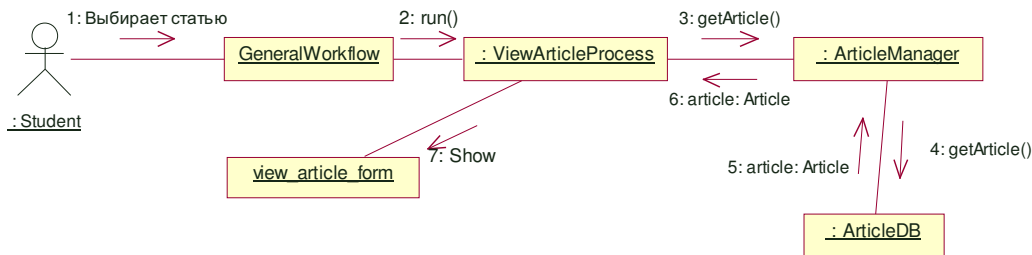


Рисунок 8. Пример диаграммы сотрудничества

Эта диаграмма позволяет наглядно показать связи классов друг с другом. Если же важно акцентировать внимание на порядке взаимодействия, используется диаграмма последовательности (sequence diagrams) (рис. 9). Эта диаграмма позволяет взглянуть на обмен сообщениями во времени, наглядно отобразить последовательность процесса. При использовании такого инструмента для создания моделей как Rational Rose, эти два вида диаграмм могут быть созданы друг из друга автоматически.

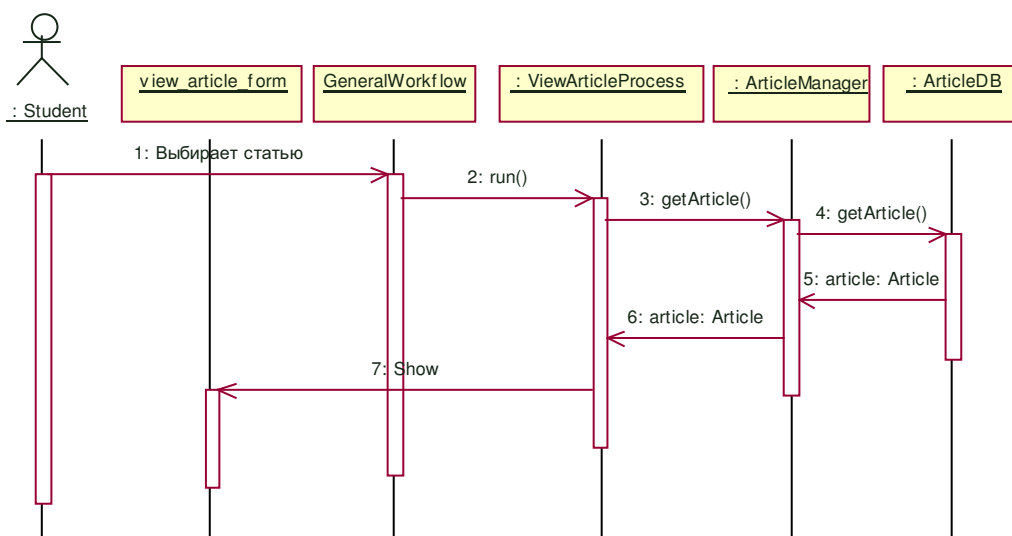


Рисунок 9. Пример диаграммы последовательностей «Просмотр статьи»

### 3. Проектирование

Следующим этапом в процессе создания системы является проектирование, в ходе которого на основе созданных ранее моделей, создается модель проектирования, которая описывает создаваемый продукт на уровне классов и компонентов.

В отличие от модели анализа, модель проектирования сильно зависит от условий реализации, применяемых языков программирования и компонентов. Эта модель должна быть максимально формализована, и поддерживаться в актуальном состоянии на протяжении всего жизненного цикла разработки системы. [24]

Для создания модели проектирования используются целый набор UML диаграмм: диаграммы классов (рис. 10), диаграммы кооперации, диаграммы взаимодействия, диаграммы активности.

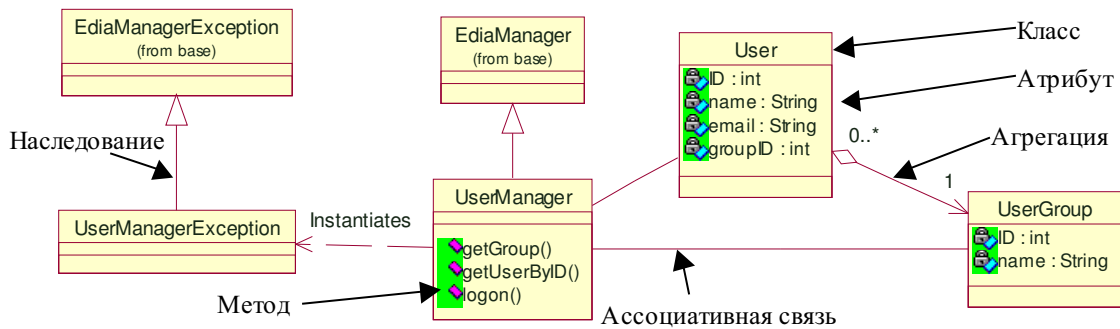


Рисунок 10. Пример диаграммы классов

#### 4. Реализация

Основная задача процесса реализации – создание системы в виде компонентов – исходных текстов программ, сценариев, двоичных файлов, исполняемых модулей и т.д. На этом этапе создается модель реализации, которая описывает то, как реализуются элементы модели проектирования, какие классы будут включены в конкретные компоненты. Данная модель описывает способ организации этих компонентов в соответствии с механизмами структурирования и разбиения на модули, принятыми в выбранной среде программирования и представляется диаграммой компонентов (рис. 11). [24]

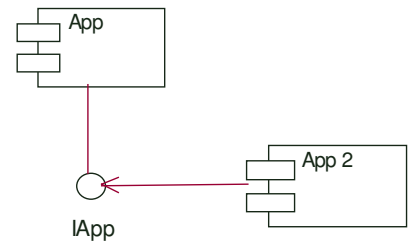


Рисунок 11. Пример диаграммы компонентов

#### 5. Тестирование

В процессе тестирования проверяются результаты реализации. Для данного процесса создается модель тестирования, которая состоит из тестовых примеров, процедур тестирования, тестовых компонентов. [24]

### 2.3. Выбор средств разработки

Проектирование сложных программных систем требует разработки и применения моделей высокого уровня абстракции и эффективных схем организации межмодульного взаимодействия. Необходимыми свойствами для создания сложных многокомпонентных информационных систем обладают объектно-ориентированные средства разработки.

При реализации информационной системы поддержки ДО за основу были приняты современные Java-технологии, базирующиеся на объектно-ориентированном языке программирования Java. Этот подход обладает следующими достоинствами.

- Переносимость, расширяемость и модульность языка Java. Java является мощным объектно-ориентированным языком и позволяет создавать платформо-независимые, расширяемые приложения. Независимость от платформы (на уровне исходного текста и на уровне исполняемого кода), дает возможность удовлетворить такому критическому требованию к программам для World Wide Web, как возможность запуска этих программ во многих системах. Существуют версии виртуальной Java-машины (Java Virtual Machine, JVM) для всех наиболее распространенных ОС (Windows, Linux, MacOS и т.п.)
- Преимущества платформы Java 2. Новые возможности, связанные с поддержкой технологии клиент-сервер, сервлеты (Servlet), получившие широкое распространение в последнее время, позволяют существенно сократить время



разработки приложений для web-серверов. Все функции взаимодействия web-сервера с Java-приложением берет на себя среда выполнения сервлетов.

- Поддержка технологии сервлетов (Servlets). В настоящее время существует широкий выбор открытых профессиональных платформ HTTP-доступа, поддерживающих Java 2. Среди них Oracle Application Server, IBM WebSphere, Apache и др.
- Использование Java Server Pages (JSP). Применение JSP, как развития технологии Servlet, позволяет разделить задачу построения web-интерфейса к базе данных, содержащей, например, электронный каталог библиотеки, на две изолированные подзадачи: программирование логики работы и внешний дизайн приложения. Такое разделение позволяет использовать ядро доступа к базе данных как основу для создания множества интерфейсов, каждый из которых максимально адаптирован к задачам конкретной организации.

Широкие возможности Java, простота применения, независимость от платформы и встроенные функции защиты делают этот язык программирования одним из лучших для создания решений для Internet.

Для построения системы была принята классическая трехуровневая архитектура:

1. Уровень представления (presentation) – окна, формы (JSP).
2. Уровень логики приложения (application logic) – бизнес-логика, правила управления процессом (сервлет).
3. Уровень данных (storage) – постоянное хранилище данных (БД).

Основной характеристикой трехуровневой архитектуры является вынесение логики приложения на отдельный уровень.

Объекты уровня представления направляют запросы на средний уровень бизнес-логики, а средний уровень взаимодействует в свою очередь с самым нижним уровнем хранения данных (рис. 12).

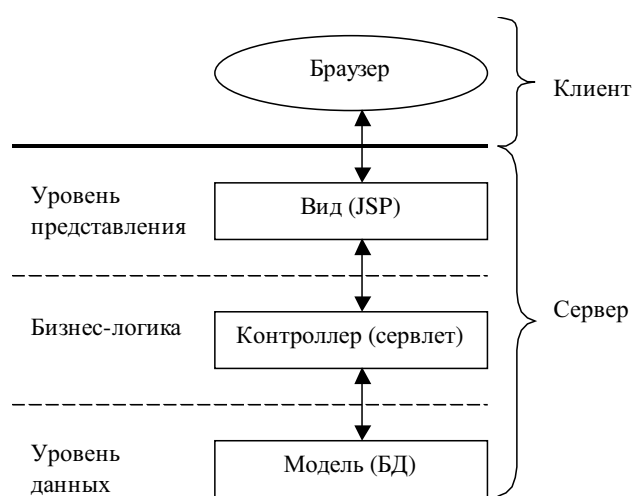


Рисунок 12. Трехуровневая архитектура приложения

Фактически, представленная архитектура является воплощением популярного шаблона проектирования «Модель-Вид-Контроллер» (Model-View-Controller, MVC). В данном случае, за представление и обработку данных отвечают отдельные компоненты. Компоненты представления (представители) – это JSP-страницы, создающие HTML/XML ответ, который определяет вид пользовательского интерфейса при обработке браузером. Компоненты обработки данных (контроллеры) не участвуют в представлении данных, они только обрабатывают запросы пользователя и формируют объекты (beans), которые используются

компонентами представления; они также определяют, в зависимости от действий пользователя, какому представителю перенаправить запрос. Контроллеры могут быть реализованы как посредством сервлетов, так и в виде JSP.

Преимущество этой архитектуры заключается в том, что компоненты представления не содержат никакой логики обработки данных, они ответственны только за прием объектов, которые были созданы контроллером, извлечение их динамического содержимого и вставку извлеченных данных в статический шаблон представления. Следовательно, такое разделение представления от содержания ведет к четкому разделению ролей разработчиков-программистов и дизайнеров. Другое преимущество

этого подхода заключается в том, что контроллеры являются единственной точкой входа в приложение, делая, таким образом, управление состоянием приложения и безопасностью единообразным и более простым для сопровождения.

## **2.4. Технологии Java Servlets и Java Server Pages**

Сервлеты являются значительным улучшением «тяжеловесной» модели программирования применяемой в сценариях CGI, которая использует отдельные серверные процессы для обработки каждого нового запроса. Модель обработки запросов, применяемая в сервлетах, использующая легковесные потоки Java (один поток на один клиентский запрос), намного более эффективна. Она также позволяет реализовать сохранение объектов между запросами, без использования базы данных или дисковой памяти. Сервлеты являются основной технологией для обработки клиентских запросов на сервере Java.

Однако замена CGI-скриптов на Java-сервлеты никак не затрагивает проблему HTML-разметки, глубоко внедренной внутрь кода, вне зависимости от языка, генерирующего страницу. Для достижения необходимого результата либо Web-дизайнер должен уметь программировать на Java, либо Java-программист должен освоить Web-дизайн.

Большинство существующих методов решения этой проблемы полагаются на добавление некоторых механизмов создания шаблонов к стандартным сервлетам. Использование шаблонов – методика, которая расширяет статические документы HTML, добавляя сценарии или заказные теги, либо автоматизируя создание классов Java, представляющих документы. Шаблон, в этом случае, представляет собой блок расширенного HTML, который благодаря Java-манипуляциям или внедренным сценариям, облегчает включение динамически сгенерированного содержания на этапе выполнения.

Java Server Pages (JSP) – ответ Sun Microsystems на технологию Active Server Pages (ASP), разработанную Microsoft. JSP и ASP – сходные технологии: обе добавляют к HTML специальные теги, язык создания сценариев, и способность обращаться к внешним программным компонентам. Sun продвигает JSP как стандартный внешний интерфейс к серверам на базе Java, включая средства Enterprise Java (J2EE). JSP можно использовать как шаблонный механизм, но особенности этой технологии позволяют рассматривать ее и отдельно.

В двух словах, файл JSP - это HTML (или XML) с включением любых комбинаций следующих элементов:

- Java-скриптлеты (scriptlets) – в HTML можно внедрять любые правильные инструкции Java;
- JSP-теги для управления и организации запросов к JavaBeans;
- JSP-теги для включения и перенаправления запросов другим страницам JSP;
- Определяемые пользователем теги.

JSP-контейнер выполняет следующие шаги, при обработке запроса на доступ к JSP-странице:

1. Страница JSP преобразовывается в исходный текст Java-сервлета
2. Исходный текст сервлета компилируется в class-файл
3. Класс сервлета загружается JVM сервера
4. Сервлет выполняется в новом потоке

Шаги с 1-го по 3-й выполняются только один раз. JSP-страница может быть заранее преобразована и откомпилирована; в противном случае, она конвертируется и компилируется в момент первого обращения к ней. После первоначальной загрузки class-файл остается в памяти пока запущен сервлет-контейнер.

Разработчик страницы имеет возможность использовать весь арсенал средств Java внутри файла страницы, сочетая скриптлеты, HTML и специальные теги JSP. Его потенциал ограничен только границами платформы Java. В то же время, дизайнер может

использовать привычные для него инструменты для задания внешнего вида страницы, и при этом не обязан знать Java. [28]

Кроме того, связка сервлет+JSP отлично подходит для реализации описанной выше концепции MVC, со всеми ее достоинствами.

## 2.5. Анализ требований к системе

Прежде всего, СДО должна быть доступна по сети (Intranet/Internet), и включать в себя как учебный материал, так и интерфейс для удаленного доступа к последнему.

Информация, хранимая в системе, должна быть хорошо структурирована, поэтому статьи хранятся в разделах, которые образуют многоуровневую иерархическую структуру. Кроме того, для улучшения восприятия самой статьи, логически самостоятельные части ее могут образовывать еще один уровень иерархии – внутри самой статьи. Для этого информация, содержащаяся в статье, разделяется по так называемым закладкам. Это позволяет более точно отразить структуру материала, отделить громоздкие выкладки и доказательства от основного материала и пр.

Важным требованием к современной информационной системе является поддержка работы с мультимедиа-данными. Это требование особенно важно для ДО, т.к. использование мультимедиа и активных элементов (java-апплеты, flash-анимация) позволяет существенно улучшить восприятие учебного материала. Это достигается за счет увеличения наглядности материала, а также возможности самостоятельного проведения студентами виртуальных экспериментов.

Основное требование к интерфейсу – он должен предоставлять удобный доступ к учебным материалам, возможность их модификации удаленным пользователем. Также существенным требованием является разграничение доступа к системе для различных пользователей – это повышает безопасность системы, ограничивает вероятность несанкционированного доступа, считывания и модификации хранимых данных.

При разработке системы учитывалось, что пользоваться ей будут четыре различных категории (группы) пользователей. Каждая группа имеет различные возможности по взаимодействию с системой. Краткое описание прав пользователей, принадлежащих различным группам, приведено в табл. 3.

Таблица 3. Группы пользователей системы

Группа	Описание
Гость (Guest)	Незарегистрированный пользователь. Имеет возможность зарегистрироваться и/или войти в систему (идентифицировать себя)
Студент (Student)	Зарегистрированный пользователь системы. Имеет возможность изменения регистрационных данных, поиска и просмотра статей, размещения отзывов и др.
Преподаватель (Tutor)	Зарегистрированный пользователь системы. Имеет возможность добавления новых статей и редактирования уже существующих.
Администратор (Admin)	Администратор системы. Осуществляет поддержку системы, управление пользователями, регистрирует новых преподавателей. Контролирует все разделы и статьями.

Каждая последующая группа пользователей имеет расширенные по сравнению с предыдущей возможности (кроме группы «Гость», к которой относятся незарегистрированные пользователи).

Для облегчения поиска нужной информации, каждая закладка статьи включает в себя набор ключевых слов, которые задаются автором материала. Все ключевые слова собраны в глоссарии, или, иными словами, предметном указателе. Таким образом,

пользователю достаточно указать в глоссарии интересующее его ключевое слово, а затем выбрать из списка содержащих это ключевое слово статей нужную.

Важным требованием к СДО является также возможность общения преподавателей и студентов. В рассматриваемой системе любой зарегистрированный пользователь может оставить свой отзыв о статье, а также просмотреть отзывы других пользователей. Это позволяет наладить обратную связь между преподавателями и студентами, и, таким образом, дает возможность преподавателям улучшать качество материала соответственно потребностям аудитории.

Возможности пользователей при работе с системой удобно представить в виде описанных выше диаграмм прецедентов. Диаграммы прецедентов для групп «Гость», «Студент», «Преподаватель» и «Администратор» представлены на рис. 13–16 соответственно.

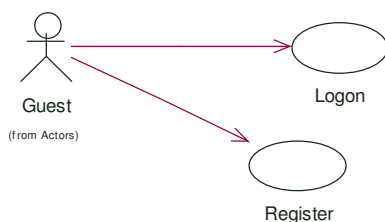


Рисунок 13. Диаграмма прецедентов группы пользователей «Гость»

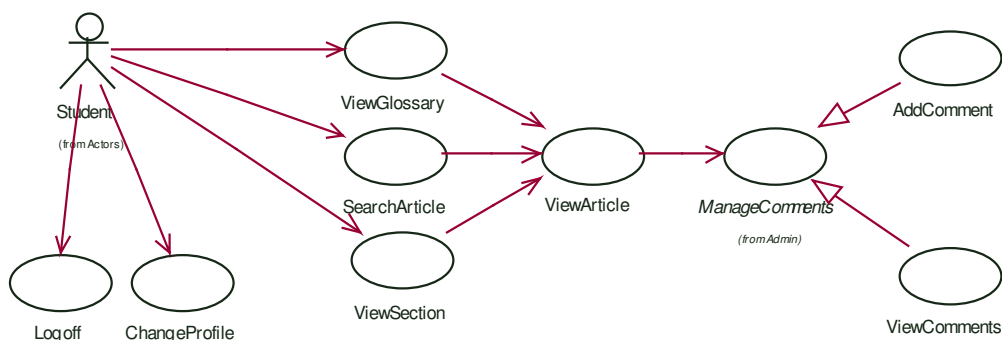


Рисунок 14. Диаграмма прецедентов группы пользователей «Студент»

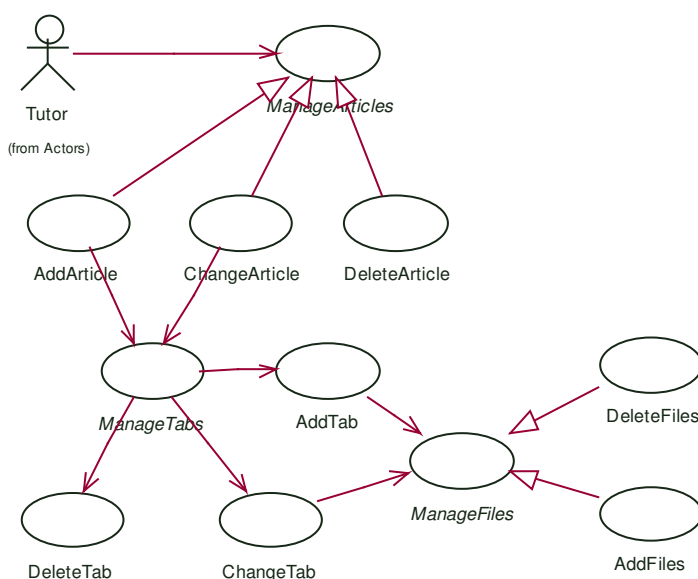


Рисунок 15. Диаграмма прецедентов группы пользователей «Преподаватель»

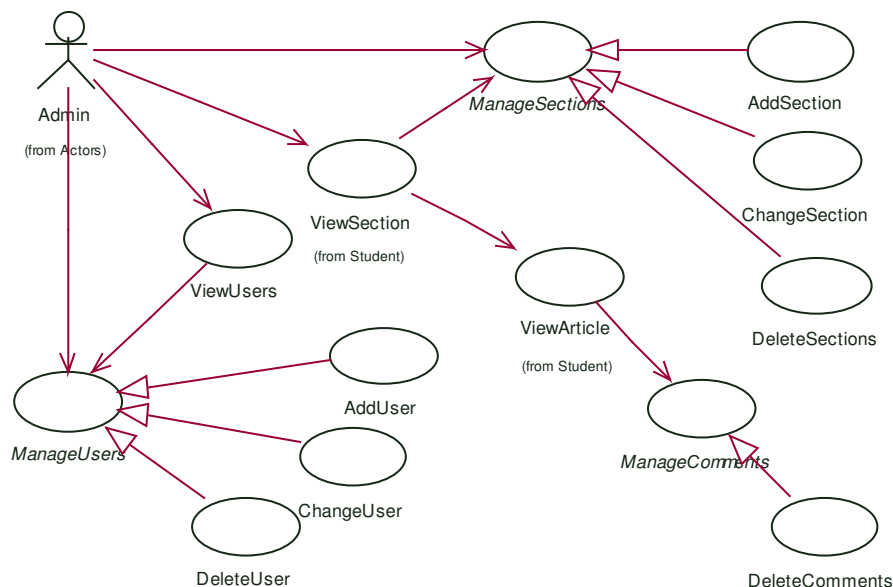


Рисунок 16. Диаграмма прецедентов группы пользователей «Администратор»

После анализа диаграмм прецедентов, каждый из вариантов использования описывается подробно. В табл. 4 приведен пример такого полного описания одного варианта использования.

Таблица 4. Детальное описание прецедента «Вход в систему»

**Общая характеристика:**

1.	<b>Действующее лицо (Actor):</b>	Гость
2.	<b>Цель в контексте (Goal in context):</b>	Идентифицировать себя и начать работу с системой в качестве зарегистрированного пользователя.
3.	<b>Иницилирующее событие (Trigger event):</b>	Пользователю необходимо воспользоваться возможностями системы
4.	<b>Предварительные условия (Pre-condition):</b>	Пользователь предварительно успешно зарегистрирован в системе
5.	<b>Критерий успешного завершения (Success End Condition):</b>	Пользователь идентифицирован системой. В зависимости от его роли в системе, он может воспользоваться различными ее возможностями.
6.	<b>Критерий неуспешного завершения (Failed End Condition):</b>	Идентификация не состоялась с указанием причин.

**Основной успешный сценарий (Main success scenario):**

Шаг	Действующее лицо	Описание действия
1.	Гость	Заходит на сайт системы.
2.	Система	Выдает <i>Входную форму (Logon form)</i> с полями для ввода логина и пароля, кнопкой « <i>Войти</i> ».
3.	Гость	Вводит: 6. Логин *; 7. Пароль *; Нажимает кнопку « <i>Войти</i> ».
4.	Система	Производит поиск и идентификацию пользователя по базе данных. Выдает <i>Главную форму</i> в зависимости от роли пользователя в системе.

### Вариации сценария (Scenario variations):

Шаг	Вариант	Описание действия
1.	Пользователь идентифицирован как <i>Студент</i> .	Система выдает <i>Главную форму Студента</i> .
2.	Пользователь идентифицирован как <i>Преподаватель</i> .	Система выдает <i>Главную форму Преподавателя</i> .
3.	Пользователь идентифицирован как <i>Администратор</i> .	Система выдает <i>Главную форму Администратора</i> .
4.	Обязательные поля формы не заполнены.	Система выдает <i>Входную форму (Logon form)</i> с сообщением, что не существует зарегистрированного пользователя с таким логином и паролем.
5.	Пароль или логин неверен.	Система сообщает пользователю, какое поле не заполнено и предлагает его заполнить.

Очевидно, что детальное описание всех прецедентов – процесс довольно трудоемкий. Но также очевидна и получаемая от таких описаний польза.

Во-первых, полное описание прецедентов позволяет уточнить и зафиксировать документально требования/пожелания пользователя. Во-вторых, полное описание варианта использования дает четкие указания программисту, что помогает при реализации функциональности системы и существенно повышает вероятность того, что последняя будет вести себя так, как того хотел пользователь.

### 2.6. Структура системы

За основу системы принята односервлетная модель. В этой модели все запросы пользователя получает единственный сервлет, который проводит обработку запроса и перенаправляет вызов соответствующему представителю (JSP) для отображения пользователю результатов запроса.

По сравнению с многосервлетной моделью, в которой различные категории пользовательских запросов должны направляться различным сервлетам, односервлетная модель имеет ряд преимуществ. В частности, существование единственной точки входа упрощает поддержку системы (программисту не нужно вспоминать, какие сервлеты отвечают за обработку того или иного вида запроса); и кроме того – и это более важно – контроль корректности поступающих запросов может быть реализован централизованно, что упрощает реализацию и поддержку подсистемы безопасности.

Состав и назначение основных компонентов представлен в табл. 5.

Таблица 5. Основные компоненты системы и их функциональное назначение

Подсистема	Назначение
Основной сервлет (MainServlet)	Получает запросы, передает их в подсистему обработки запросов, после чего перенаправляет полученный результат для отображения соответствующему представителю.
Подсистема безопасности	Контролирует права пользователя на выполнение тех или иных действий.
Подсистема работы с БД	Обеспечивает работу с БД на низком уровне (SQL-запросы), оптимизацию быстродействия при обращениях к БД и т. п. (см. 2.6.1).

Подсистема	Назначение
Подсистема обработки запросов	Осуществляет предобработку и преобразование «сырых» http-запросов, запуск соответствующих им процессов, формирование пользовательского меню и т. п. Делится на две основных части: <ul style="list-style-type: none"> <li>• Подсистема управления пользователями;</li> <li>• Подсистема управления содержанием.</li> </ul>
Сервлет для работы с файлами (FileServlet)	Обеспечивает получение пользователем присоединенных к статье файлов (см. 2.8.3).
Подсистема журналирования	Ведет и сохраняет протокол действий пользователя, системные и служебные сообщения, сообщения для отладки и т. п.
Менеджеры	Осуществляют взаимодействие между высокоуровневыми компонентами системы и подсистемой работы с БД, обеспечивают работу с БД на высоком уровне абстракции (см. 2.6.2).

Схема взаимодействия компонентов системы описана в разделе 2.7.

### 2.6.1. Структура БД

Для хранения служебной и пользовательской информации в системе используется СУБД MySQL 3.23. Выбор именно этой СУБД обусловлен тем, что она имеет хорошее быстродействие, широко распространена и бесплатна для некоммерческого использования.

Структура БД системы была спроектирована в САПР ErWin 3.52, и представлена на рис. 17. Описание отдельных таблиц БД приведено в табл. 6.

Таблица 6. Описание таблиц БД

Таблица	Описание
SECTIONS	Разделы, в которых находятся статьи. Для раздела задается его название и дата создания.
ARTICLES	Информационные статьи. Для статьи задается: раздел, автор, название и дата создания. С каждой статьей связана одна или несколько закладок (TABS), а также список отзывов на статью (COMMENTS).
TABS	Закладки статьи. Каждая закладка имеет название и содержание (body) – собственно, информационный текст. С закладкой связываются ключевые слова (KEYWORDS) и присоединенные файлы (FILES).
COMMENTS	Отзывы о статье. Каждый отзыв, кроме непосредственно текста, содержит информацию о пользователе, его оставившем, а также о дате размещения.
FILES	Прикрепленные к закладке файлы. Каждый файл имеет имя и собственно данные.
KEYWORDS	Ключевые слова, назначенные закладкам.
KEYWORDS2TABS	Вспомогательная таблица для осуществления связи «многие-ко-многим» между таблицами KEYWORDS и TABS.
USERS	Пользователи системы. Для каждого пользователя указывается принадлежность к одной из групп, уникальный логин, пароль, полное имя (ФИО), адрес e-mail.
GROUPS	Группы пользователей. Права каждой группы определяются подсистемой безопасности на основе таблицы прав (RIGHTS).
PROCESS	Процессы. Записи этой таблицы связывают имя процесса с именем соответствующего класса, реализующего этот процесс.

Таблица	Описание
RIGHTS	Права групп пользователей. Устанавливает, какие процессы могут запускаться пользователями определенной группы.
MENUNAME	Названия пунктов меню. Определяет названия процессов, как они должны отображаться в меню пользователя.
MAIN_MENU	Главное меню. Определяет отображаемые в главном меню элементы для каждой группы пользователей.
CONTEXT_MENU	Контекстное меню. Устанавливает, какие процессы будут отображаться в контекстном меню определенного «процесса-хозяина».
PRESENTORS	Представители. Устанавливает связь имен представителей с реальными именами jsp-файлов.

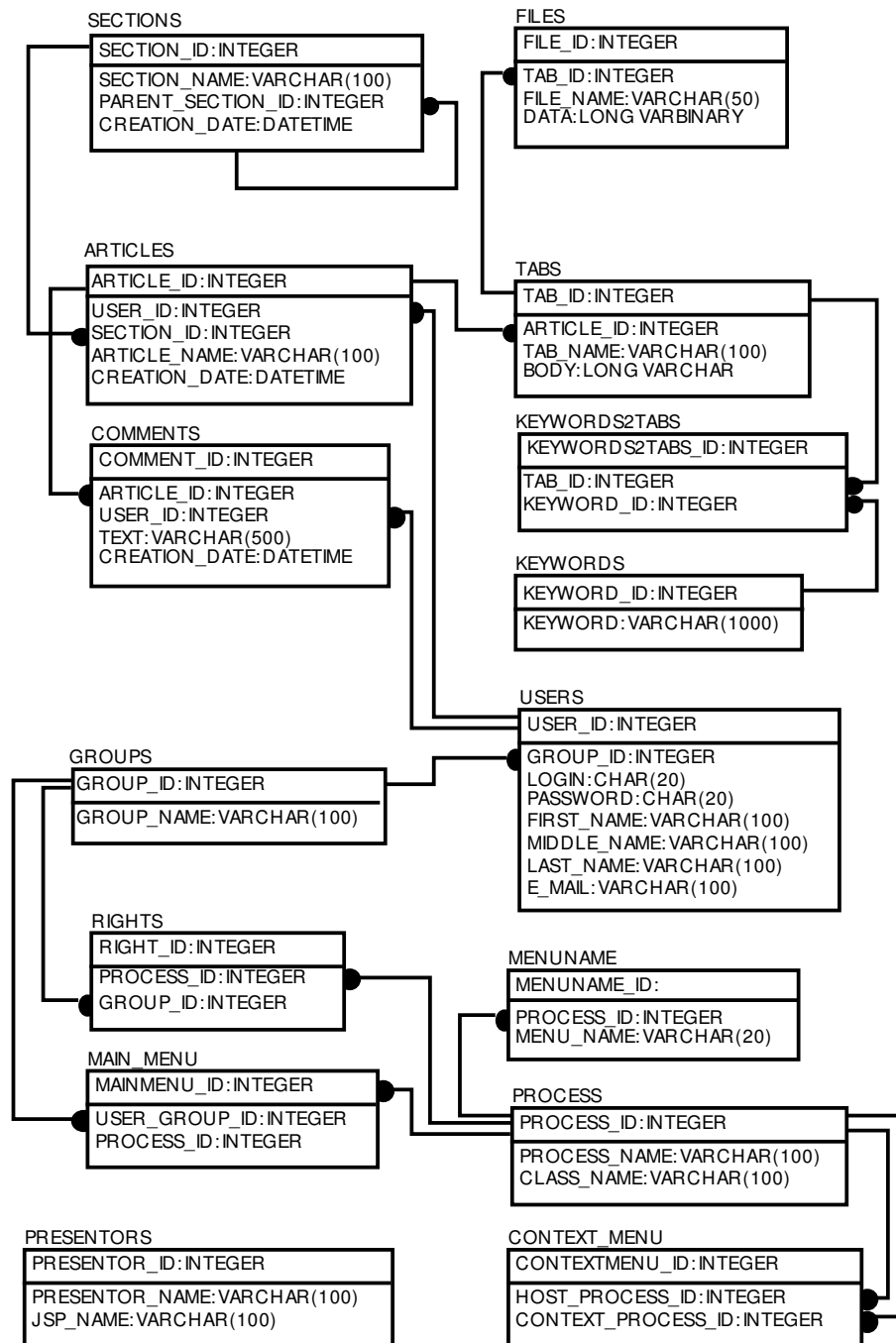


Рисунок 17. Структура БД системы



## 2.6.2. Менеджеры

Базовая функциональность системы основывается на нескольких управляющих классах – менеджерах (это классы-наследники `com.edia.base.EdiaManager`).

Менеджеры обеспечивают выполнение основных высокоуровневых операций с БД (Например: «Получить список статей указанного раздела», «Изменить данные пользователя» и т. п.)

По выполняемым функциям, менеджеров можно разделить на три вида:

- Ядро системы: подсистема безопасности (`SecurityManager`) и подсистема обработки запросов (`ProcessManager`);
- Управление пользователями (`UserManager`);
- Управление содержанием (`ContentManager`, `SectionManager`, `ArticleManager`, `GlossaryManager`, `CommentManager`, `FileManager`).

Список всех менеджеров системы и их назначение приведены в табл. 7.

Таблица 7. Назначение управляющих классов (менеджеров) системы

Класс	Описание
<code>com.edia.base.SecurityManager</code>	Основа подсистемы безопасности. Проверяет, имеет ли текущий пользователь право на запуск того или иного процесса, на изменение/удаление разделов, статей и т. п.
<code>com.edia.process.ProcessManager</code>	Основа подсистемы обработки запросов. Обеспечивает запуск процесса, соответствующего пришедшему запросу; обеспечивает возможность использования результатов работы одного процесса другим (последовательная обработка одного запроса несколькими процессами); отвечает за своевременное обновление пользовательского меню.
<code>com.edia.user.UserManager</code>	Обеспечивает базовые функции управления пользователями: <ul style="list-style-type: none"><li>• проверка, зарегистрирован ли пользователь с указанным логином/паролем;</li><li>• добавление, изменение, удаление пользователей;</li><li>• проверка корректности данных о пользователе;</li><li>• загрузка меню, соответствующего правам пользователя.</li></ul>
<code>com.edia.content.ContentManager</code>	Родительский класс для всех менеджеров, управляющих содержанием (приведены ниже).
<code>com.edia.content.SectionManager</code>	Предоставляет базовые функции для работы с разделами: <ul style="list-style-type: none"><li>• добавление, удаление, изменение раздела;</li><li>• получение списка дочерних разделов.</li><li>• поиск раздела.</li></ul>
<code>com.edia.content.ArticleManager</code>	Предоставляет базовые функции для работы со статьями и закладками: <ul style="list-style-type: none"><li>• добавление, удаление, изменение статей/закладок;</li><li>• получение списка статей указанного раздела;</li><li>• получение списка закладок указанной статьи;</li><li>• поиск статей/закладок.</li></ul>
<code>com.edia.content.GlossaryManager</code>	Предоставляет базовые функции для работы с ключевыми словами и глоссарием.
<code>com.edia.content.CommentManager</code>	Предоставляет базовые функции для работы с пользовательскими отзывами о статьях.
<code>com.edia.content.FileManager</code>	Предоставляет базовые функции для работы с прикрепленными файлами: добавление, удаление, получение из БД.

### 2.6.3. Процессы

Каждый прецедент, в общем случае, предполагает отправку некоего запроса к системе и получение от нее результата обработки этого запроса. Обработка запроса для каждого конкретного варианта использования называется процессом. Процесс может быть одноэтапным (получение запроса – вывод результата) или двухэтапным (получение запроса – формирование страницы с формой для ввода данных – получение введенных данных – вывод результата).

Таким образом, прецеденты реализуются посредством отдельных самостоятельных процессов (в программе – это все классы-наследники от `com.edia.process.Process`). В случае необходимости, функциональность системы может быть легко расширена путем добавления новых процессов. Для этого достаточно реализовать класс-наследник от `com.edia.process.Process`, и зарегистрировать новый процесс в БД:

- указать связь «имя процесса» – «имя класса»;
- указать, какие группы имеет право запускать этот процесс;
- указать связь «имя представителя» – «реальный путь к jsp-файлу»;
- при необходимости, сформировать контекстное меню для нового процесса.

Следует заметить, что добавление новых процессов не требует перекомпиляции исходного кода системы – подключение новых процессов осуществляется «на лету», посредством регистрации их в БД. Таким образом, возможности системы можно расширять, не прерывая ее работы.

С точки зрения ООП, каждый процесс представлен классом-наследником `com.edia.process.Process`; также существует ряд дополнительных классов, которые реализуют вспомогательные функции. Описание этих классов приведено в табл. 8.

Таблица 8. Базовые классы поддержки работы процессов

Класс	Описание
Process	Реализует главную функциональность процесса. Наследники могут перекрыть следующие методы: <ul style="list-style-type: none"><li>• <code>run()</code> – «входная точка» в процесс, этот метод вызывается при запуске процесса;</li><li>• <code>isFirstCall()</code> – по параметрам запроса определяет, определяет, какой шаг двухэтапного процесса должен быть выполнен;</li><li>• <code>processFirstCall()</code> – формирует результат первого шага двухэтапного процесса;</li><li>• <code>processMain()</code> – формирует результат второго шага двухэтапного процесса, или результат одноэтапного процесса;</li><li>• <code>processInvalidParams()</code> – формирует результат при некорректном значении параметров запроса (обязательное поле формы не заполнено или имеет некорректный формат);</li><li>• <code>restoreState()</code> – восстанавливает недостающие параметры из предыдущих значений;</li><li>• <code>saveState()</code> – сохраняет параметры запроса для того, чтобы их можно было восстановить методом <code>restoreState()</code>.</li></ul>
ProcessRequest	Содержит все параметры запроса к процессу.

Класс	Описание
ProcessResult	Содержит результат работы процесса. Объект этого класса возвращается методом <code>Process.run()</code> , и содержит все данные, которые необходимы для отображения jsp-страницы.
ProcessValidator	Проверяет корректность запроса к процессу: все ли необходимые параметры заданы; допустимы ли указанные значения параметров; имеет ли право текущий пользователь на выполнение процесса с данными параметрами.
ProcessState	Отвечает за сохранение и восстановление параметров запроса к данному процессу. В дальнейшем это позволяет вызывать процесс с неполным списком параметров – недостающие значения будут извлечены из пользовательской сессии с помощью <code>ProcessState</code> .
ProcessException	Исключения, которые могут возникнуть в процессе. В основном, исключения этого класса генерируются во время проверки корректности запроса в <code>ProcessValidator</code> , когда продолжение работы с указанными невозможно (например, при попытке обращения к несуществующей статье; или в случае, когда пользователь не имеет прав на выполнение запрошенных действий).

## 2.7. Взаимодействие компонентов системы

Рассмотрим, как взаимодействуют компоненты системы при обработке типичного входного запроса (*general workflow*). Для этого будем использовать диаграммы последовательностей языка UML, которые являются удобным средством для отображения порядка взаимодействия компонентов и подсистем.

В качестве примера возьмем двухэтапный процесс регистрации администратором нового пользователя. Диаграмма последовательностей для этого процесса приведена на рис. 19.

Рассмотрим эту диаграмму подробнее.

Прецедент «Добавление пользователя» инициируется администратором посредством выбора в меню пункта «Добавить пользователя». (1)

Запрос, содержащий только имя процесса («`add_user`»), поступает на основной сервлет (`MainServlet`), а затем – в подсистему обработки запросов (*General workflow*).

Подсистема обработки запросов (см. табл. 5) выполняет общие для всех запросов действия:

- Проверяет, имеет ли пользователь право на запуск запрошенного процесса;
- запускает запрошенный процесс;
- перенаправляет результат процесса соответствующему представителю для отображения.

Диаграмма последовательностей для подсистемы обработки запросов изображена на рис. 18.

Подсистема обработки запросов создает объект класса `AddUserProcess`, и запускает его на выполнение методом `run()`. (1.1)

`AddUserProcess` создает объект класса `AddUserProcessValidator`, который проверяет корректность входного запроса. (1.1.1) Т.к. на первом этапе процесса «Добавление пользователя» не требуется никаких дополнительных параметров, запрос признается корректным.

Осуществляется обработка первого этапа процесса добавления пользователя. (1.1.2) Как правило, это извлечение из БД данных, необходимых для отображения формы, в которую пользователь будет вводить параметры для второго этапа процесса. В данном

случае, из БД извлекается список всех существующих групп пользователей (1.1.2.1, 1.1.2.1.1).

Результат процесса передается представителю *add\_user\_form*, который отображает администратору *Форму добавления пользователя* с полями для ввода данных о пользователе. (1.1.2.2)

Пользователь заполняет необходимые поля, нажимает кнопку «Отправить». (2)

Снова запускается *AddUserProcess*, снова происходит проверка корректности параметров запроса. (2.1, 2.1.1) В случае, если какой-либо из обязательных параметров не задан, или имеет некорректный формат (адрес e-mail), пользователю выводится та же самая *Форма добавления пользователя*, вместе с указанием причин ошибки. (2.1.1.1, 2.1.1.1.1)

Если все параметры заданы корректно, методом *processMain()* осуществляется обработка второго этапа процесса – непосредственно добавление нового пользователя в систему. (2.1.2, 2.1.2.1, 2.1.2.1.1), после чего система должна отобразить администратору список зарегистрированных пользователей. Для этого производится передача управления процессу *ViewUsersProcess* (2.1.2.2, 2.1.2.2.1). Для упрощения диаграммы, подробности работы *ViewUsersProcess* опущены.

Таким образом, в результате обработки запроса администратору выводится список зарегистрированных пользователей системы (2.1.2.2.1.1), содержащий и только что добавленного пользователя.

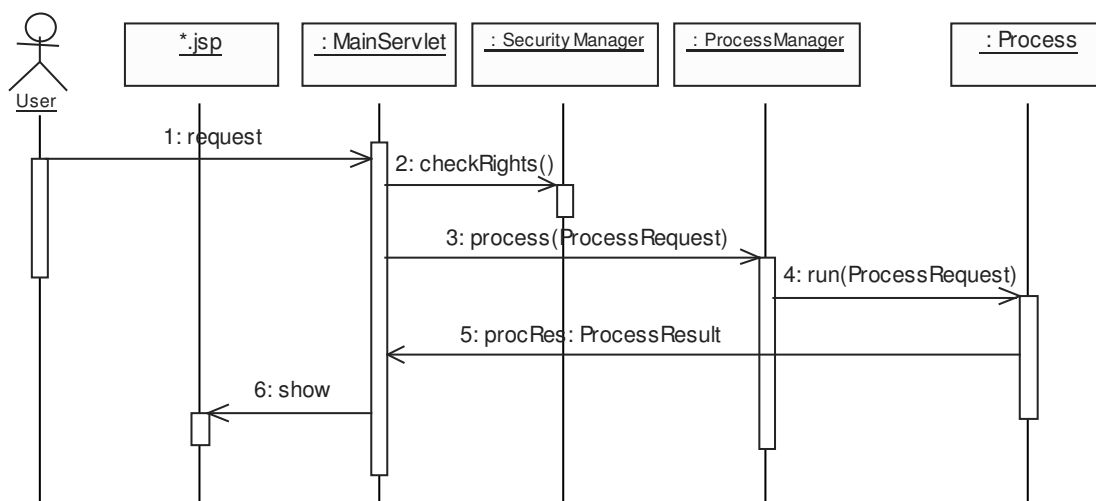


Рисунок 18. Диаграмма последовательностей для подсистемы обработки запросов

## 2.8. Пользовательский интерфейс

В этом разделе содержится описание пользовательского интерфейса созданной системы поддержки ДО, а также общие рекомендации по работе ней.

Вид пользовательского интерфейса системы представлен на рис. 20.

В верхней части страницы находится логотип (который одновременно является ссылкой на главную страницу) и строка главного меню. Под логотипом расположено контекстное меню. Оставшаяся часть страницы является основной – здесь отображаются результаты работы процесса.

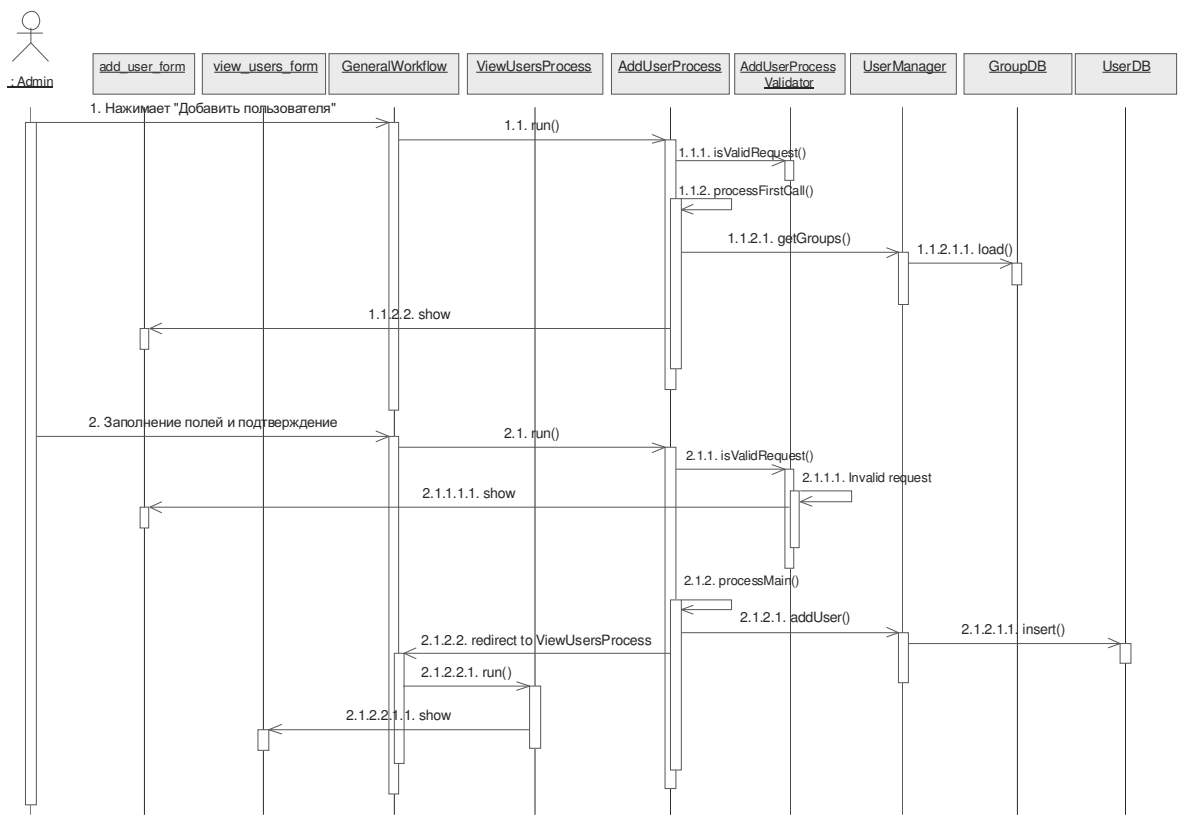


Рисунок 19. Диаграмма последовательностей для процесса «Добавление пользователя»

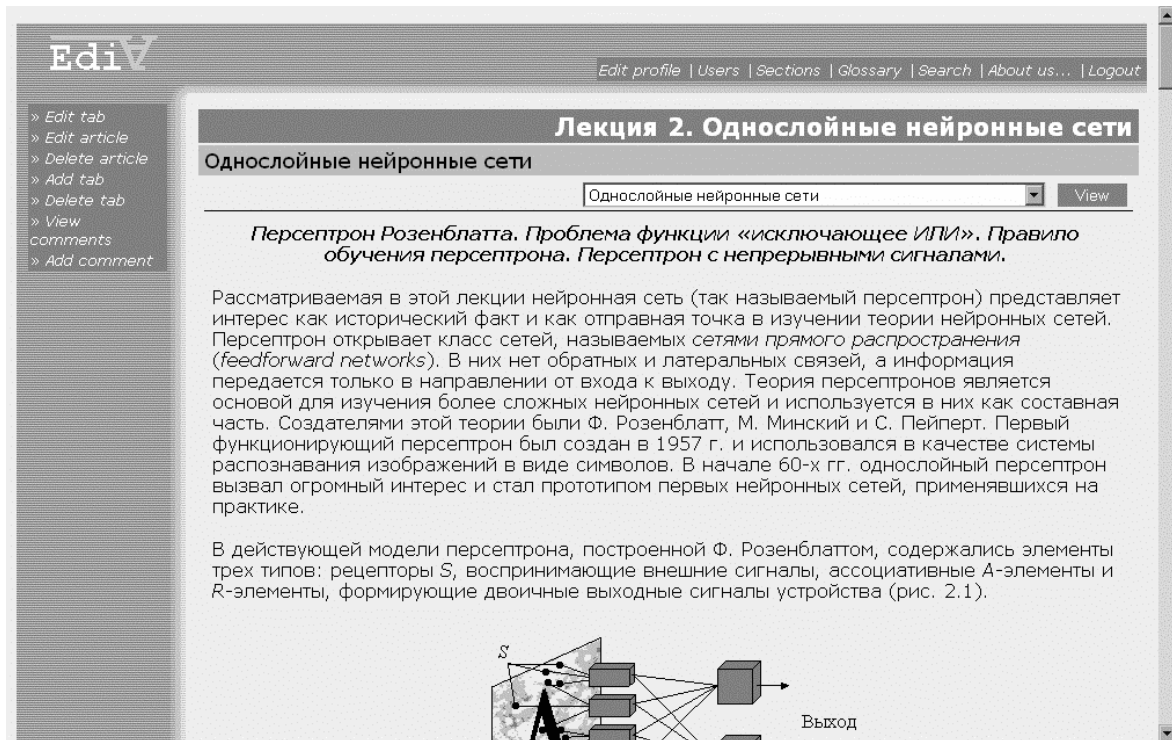


Рисунок 20. Вид пользовательского интерфейса системы при просмотре статьи

### 2.8.1. Главное и контекстное меню

Меню являются одним из средств взаимодействия с системой, они предоставляют быстрый доступ к нужным процессам.

Интерфейс предоставляет два типа меню: главное и контекстное.

Главное меню располагается в верхней части страницы. Набор его элементов зависит только от группы, к которой принадлежит текущий пользователь (от его прав), и не зависит от текущего состояния системы.

Для всех зарегистрированных пользователей в главном меню находятся пункты:

- Изменить личные данные (Change profile);
- Просмотр разделов (Sections);
- Просмотр глоссария (Glossary);
- Поиск (Search);
- О системе (About us);
- Выход (Logout).

Контекстное меню располагается слева в левой части страницы. Набор его элементов зависит не только от прав текущего пользователя, но также и от контекста выполнения, т. е. от выполняемого в настоящее время процесса. Например, при просмотре преподавателем раздела, контекстное меню предоставляет ему возможность разместить новую статью; а при просмотре статьи – просмотреть отзывы о ней и т. д.

При выборе пользователем какого-либо пункта меню, Сервету посылается запрос, содержащий только имя соответствующего процесса, без параметров. Необходимые для корректной работы процесса параметры восстанавливаются по сохраненным в пользовательской сессии значениям (см. табл. 8, класс `ProcessState`). Таким образом учитывается и влияние контекста.

Например, администратор при просмотре разделов (запрос содержит только параметр `section_id` – номер просматриваемого раздела в БД) выбирает в контекстном меню пункт «Удалить раздел». Вызывается процесс удаления раздела, но т. к. параметра `section_id` (какой раздел удалять) ему передано не было – восстанавливается последнее полученное значение `section_id`, которое в данном случае соответствует

просматриваемому разделу. Таким образом, система предложит администратору подтвердить удаление просматриваемого раздела и после того, как пользователь подтвердит свое намерение удалить раздел, это будет сделано.

## **2.8.2. Разделы, статьи и закладки**

Основными информационными элементами в системе являются статьи. Соответственно своей тематике, статьи размещены в разделах, которые образуют многоуровневую иерархическую структуру.

Каждая статья имеет также еще один, внутренний уровень иерархии – так называемые закладки. Это позволяет лучшим образом представить структуру статьи, и, следовательно, улучшает ее восприятие. Например, на первой закладке может располагаться основная информация, на второй – демонстрационные материалы (апплеты), на третьей – доказательства сложных теорем, на четвертой – список литературных источников или web-ссылок по теме статьи.

Создавать новые и изменять существующие разделы может только пользователь, принадлежащий к группе администраторов. При создании/изменении раздела нужно указать его название, а также родительский раздел.

Преподаватели могут размещать новые статьи в любом из существующих разделов. Изменить статью может только ее автор или администратор. Для каждой статьи указывается ее название, а также раздел, в котором она должна располагаться.

С каждой статьей связан набор ее закладок. Преподаватель может изменить порядок отображения закладок в списке с помощью соответствующих кнопок (☛).

При добавлении/изменении закладки преподаватель задает ее название, содержание (html-текст), список ключевых слов, описывающих содержание. Текст закладки можно ввести непосредственно в соответствующее поле, а можно указать html-файл, содержимое которого будет загружено системой и использовано в качестве текста закладки. Размер этого файла не должен превышать указанного в настройках (файл `edia.properties`) значения `MAX_FILE_SIZE` (по умолчанию, 1 Мб). В противном случае пользователю выводится сообщение о том, что указанный файл имеет слишком большой размер.

Ключевые слова, назначенные закладке, автоматически добавляются в глоссарий (см. 2.8.5).

К каждой закладке можно присоединить несколько файлов, которые будут доступны из html-текста закладки (см. 2.8.3).

## **2.8.3. Присоединяемые файлы**

Разработанная система поддерживает возможность включения в html-текст закладок мультимедиа и активных элементов. Это могут быть изображения, аудио- и видеоданные, flash-анимация, java-апплеты, а также файлы любого другого типа.

Разработанная система поддерживает возможность присоединения файлов к закладкам. Таким образом, преподаватель может загрузить не только текстовую информацию, но и набор изображений, аудио- и видеоданных, или файл любого другого типа. На присоединенные файлы можно ссылаться в html-тексте закладки, что позволяет отображать статьи со встроенными изображениями, видео- и аудиоклипами, и таким образом существенно улучшить восприятие материала.

Все присоединенные файлы сохраняются в БД и доступ к ним можно получить через сервлет (`FileServlet`). Сервер Jakarta Tomcat может быть настроен на перенаправление запросы по определенным адресам на сервлет. В данном случае, все запросы на адрес вида `“files/*”` перенаправляются сервлету `FileServlet`, который в качестве результата возвращает из БД файл с указанным после `“files/”` именем. Таким образом, ссылки на присоединенные файлы имеют простой унифицированный формат `“files/поддиректории/имя_файла”`.

При обработке запроса, `FileServlet` восстанавливает из контекста идентификационный номер закладки, к которой присоединен запрашиваемый файл, а также осуществляет проверку прав пользователя на доступ к указанному файлу.

Рассмотрим следующий пример. Допустим, в тексте одной закладки находится ссылка вида `“files/somefile.jpg”`, в тексте другой закладки – аналогичная ссылка, но на файл с другим содержимым. Несмотря на то, что ссылки (т.е. имена файлов) полностью совпадают, в каждую закладку будет загружен именно соответствующий ей файл. В данном случае решающую роль снова играет механизм сохранения предыдущих параметров запросов – именно он позволяет использовать унифицированный формат ссылки на файл, без явной привязки к определенной закладке.

Чтобы присоединить к закладке один или несколько файлов, пользователю нужно в режиме редактирования закладки выбрать в контекстном меню пункт «Добавить файлы». С помощью появившейся формы можно одновременно загрузить до пяти файлов (это количество задается параметром `MAX_UPLOAD_FILE_COUNT` в файле настроек `edia.properties`). Суммарный объем загружаемых одновременно файлов не должен превышать указанного в настройках значения `MAX_FILE_SIZE` (по умолчанию, 1 Мб).

Для каждого файла можно задать псевдодиректорию, в которой он будет располагаться. Это позволяет структурировать доступ к файлам различного типа из текста закладки (например, изображения – в `“files/images”`, апплеты – в `“files/applets”` и т. п.)

#### **2.8.4. Пользовательские отзывы**

Все зарегистрированные пользователи системы могут просматривать и размещать свои отзывы о прочитанных статьях. Это позволяет наладить связь между преподавателями и студентами, что, в свою очередь, дает возможность преподавателям адаптировать свои статьи соответственно потребностям аудитории. Подсистема пользовательских отзывов может также рассматриваться как средство организации набора вопросов и ответов, относящихся к данной статье.

Для просмотра отзывов о статье, пользователь выбирает в контекстном меню пункт «Отзывы». Чтобы поместить отзыв, следует выбрать в контекстном меню «Добавить отзыв», затем ввести нужный текст в соответствующее поле и нажать кнопку «Добавить».

#### **2.8.5. Глоссарий**

Наличие поддержки работы с ключевыми словами является отличительной особенностью разработанной системы по сравнению с рассмотренными выше СДО. Ключевые слова позволяют проиндексировать текстовую информацию, и значительно упрощают поиск.

Глоссарий является удобным средством поиска статей по нужной тематике. В нем в алфавитном порядке отображаются все ключевые слова всех закладок всех статей, находящихся в БД.

При добавлении или изменении закладки, изменения связанных с ней ключевых слов учитываются автоматически. Таким образом, динамическое обновление глоссария гарантирует, что он всегда содержит корректную информацию.

Чтобы найти нужную информацию, пользователь выбирает в глоссарии нужное ключевое слово. Система выводит список всех закладок, содержащих это слово, с указанием полного пути к каждой из них (т. е. название статьи и всех родительских разделов, вплоть до корневого). Из этого списка пользователь выбирает нужную закладку (статью, раздел). Таким образом, поиск по глоссарию может быть осуществлен посредством всего двух кликов мышью.

#### **2.8.6. Поиск**

Система обладает следующими возможностями поиска нужной информации:

- Поиск в названии раздела;



- Поиск в названии статьи;
- Поиск в названии закладки;
- Поиск по имени автора;
- Поиск по ключевым словам.

Пользователь может использовать любые комбинации вышеприведенных вариантов.

Результат поиска выводится аналогично результатам поиска по глоссарию: раздел(ы)/статья(если есть)/закладка(если есть).

Поисковая строка должна содержать не менее трех символов (задается параметром `MIN_QUERY_LENGTH` в файле `edia.properties`).

#### ***Формат поисковой строки***

Для сравнения строк при поиске используются встроенные средства языка SQL. Это позволяет минимизировать время поиска, т. к. поиск на уровне БД оптимизирован наилучшим образом самими ее разработчиками.

В частности, для сравнения строк используется SQL-оператор `LIKE`. Специальные символы, понимаемые этим оператором, позволяют сделать поиск более гибким.

Оператор сравнения строк `LIKE` позволяет использовать в поисковой строке знак “\_” (подчеркивание) для обозначения любого одиночного символа, и символ “%” для обозначения любого количества символов (в том числе, и равного нулю). При сравнении символы регистр символов игнорируется.

## **2.9. Выводы**

В данной главе были рассмотрены детали проектирования и реализации информационно-справочной системы; обоснован выбор используемых технологий, приведено их краткое описание; проведен анализ требований к функциональным возможностям системы; представлены основные UML-диаграммы, использовавшиеся на этапе проектирования. Также были рассмотрены подробности функционирования разработанной системы, ее структура и порядок взаимодействия отдельных компонентов и подсистем.

В конце главы приведено руководство пользователя (раздел 2.8), а также некоторые рекомендации по работе с системой.

### 3. Учебно-методический комплекс «Нейронные сети»

В данной главе будет приведен пример применения рассмотренной в главе 2 информационно-справочной системы – для дистанционного обучения по нейронным сетям.

Искусственные нейронные сети – одна из наиболее интенсивно развивающихся областей информационных технологий, поэтому соответствующие курсы все чаще включаются в учебные программы технических университетов. [16]

За основу курса ДО принят учебно-методический комплекс (УМК) «Нейронные сети», созданный и уже несколько лет используемый на факультете радиофизики и электроники БГУ. [29]

Основной характеристикой комплекса является ориентация на развитие у студентов практических навыков применения и использования нейронных сетей для обработки данных. [30]

#### 3.1. Структура УМК

УМК «Нейронные сети» включает в себя:

- рабочую программу курса,
- методические указания к лабораторным работам
- конспект лекций [30].

УМК состоит из трех учебных модулей, по четыре лекции в каждом:

- 1 Биологический прототип и теоретические основы функционирования ИНС.
  - 1) Биологические и искусственные нейроны (Историческая справка. Обработка информации человеком и компьютером. Биологический нейрон. Модель нейрона. Биологические и искусственные нейронные сети).
  - 2) Однослойные нейронные сети (Персептрон Розенблатта. Проблема функции «исключающее ИЛИ». Правило обучения персептрона. Персептрон с непрерывными сигналами).
  - 3) Многослойный персептрон (Структура многослойной сети. Активационные функции. Архитектура. Преодоление ограничения линейной разделимости. Примеры. Сеть с линейными нейронами и ее сведение к однослойной. РБФ-сеть).
  - 4) Контролируемое обучение нейронных сетей (Структура многослойной сети. Активационные функции. Архитектура. Преодоление ограничения линейной разделимости. Примеры. Сеть с линейными нейронами и ее сведение к однослойной. РБФ-сеть).
- 2 Типовые применения однонаправленных ИНС.
  - 1) Задача классификации (Типовые задачи. Этапы применения искусственных нейронных сетей. Предварительная обработка данных. Примеры).
  - 2) Распознавание образов (Распознавание как статистическая задача. Понижение размерности входов. Метод главных компонент. Примеры задач. Распознавание дефектов изделий).
  - 3) Нейросетевая аппроксимация и моделирование (Пример и постановка задачи. Аппроксимация персептроном и РБФ-сетью. Особенности обучения. Пример применения. Нейросетевое моделирование).
  - 4) Анализ данных (Задача прогнозирования временного ряда. Метод скользящего окна. Пример линейной нейронной сети. Подготовка данных. Прогнозирование нагрузок электроэнергетической системы. Финансовые временные ряды и методика их предсказания).
- 3 Архитектура, алгоритмы обучения и применения ИНС с латеральными и обратными связями.

- 1) Нейронные сети с обратными связями и алгоритмы их обучения (Устойчивость и особенности сетей с обратными связями. Рекуррентные и релаксационные сети. Сети Хопфилда и Хэмминга. Ассоциативная память).
- 2) Сети с латеральными связями и их обучение (Недостатки детерминированных методов обучения. Алгоритм стохастического обучения и его варианты. Машина Больцмана и примеры ее применения. Методы статистической физики в теории нейронных сетей).
- 3) Стохастические сети и алгоритмы их обучения (Сеть Кохонена. Конкуренционное обучение. Особенности неконтролируемого обучения. Метод обучения Хебба и его модификации. Самоорганизующиеся нейронные сети).
- 4) Сети сложной архитектуры и их применение (Недостатки простых сетей. Архитектура АРТ. Когнитрон. Неокогнитрон).

Одна из проблем, возникающих в процессе преподавания курса, заключается в высокой стоимости лицензионных программных средств для моделирования нейронных сетей, необходимых для выполнения лабораторных работ. [31]

В адаптированной для дистанционного обучения версии УМК, совместно с лекциями студентам предоставляются Java-апплеты, позволяющие проводить моделирование различных аспектов функционирования ИНС в режиме реального времени. Это позволяет существенно улучшить качество восприятия материала за счет высокой степени наглядности. Студенты могут проводить собственные эксперименты с различными типами нейронных сетей, что, безусловно, положительно сказывается на качестве понимания и усвоения материала.

### **3.2. Демонстрационные апплеты**

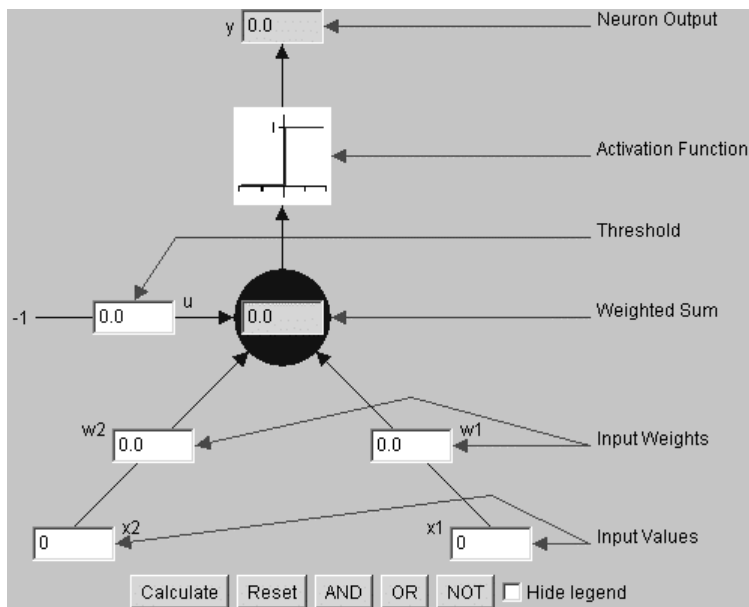
Адаптированный УМК включает в себя свободно распространяемые апплеты, большая часть из которых поставляется с открытыми исходными кодами. Этот факт не только решает проблему лицензирования программных средств моделирования ИНС, но также дает студентам возможность развивать апплеты, приспособивать их для решения собственных практических задач.

В настоящее время курс содержит следующие апплеты:

- Искусственный нейрон [32];
- Обучение персептрона [33];
- Многослойный персептрон [34];
- Аппроксимация функций [35];
- Прогнозирование временного ряда [34];
- Распознавание символов [36];
- Аппроксимация РБФ-сетью [37];
- Самоорганизующиеся сети [38];
- Ассоциативная память [34].

Ниже приведено краткое описание этих апплетов.

### 3.2.1. Искусственный нейрон



Данный апплет моделирует работу классического искусственного нейрона (рис. 21). Пользователь может задать величины весов, порога, указать вид активационной функции, а затем рассчитать выходное значение нейрона при указанных значениях на входах.

Рисунок 21. Апплет «Искусственный нейрон»

### 3.2.2. Обучение персептрона

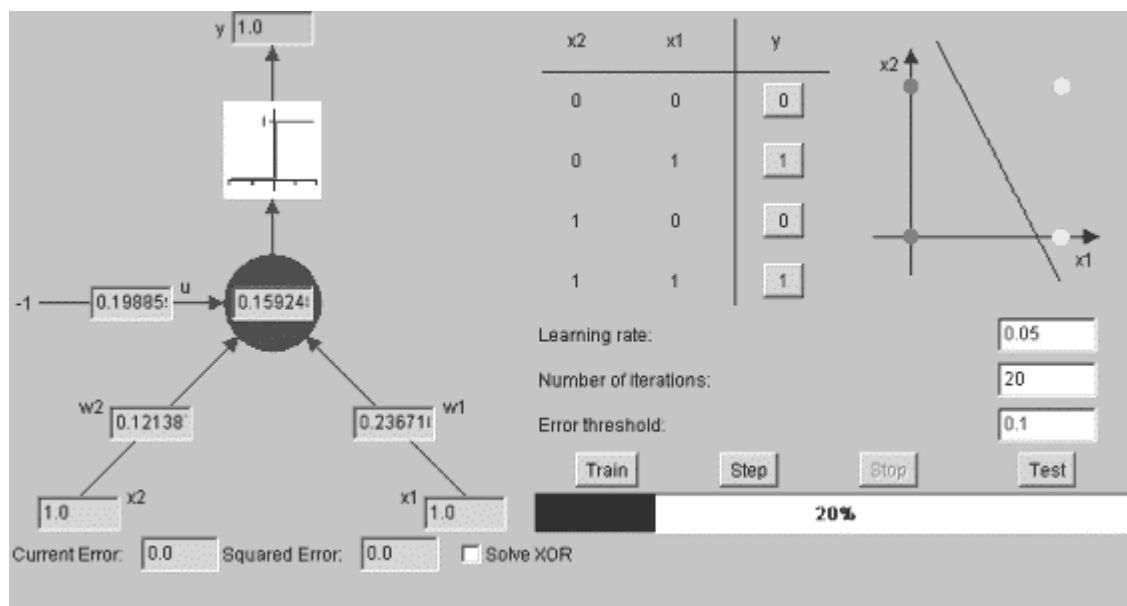
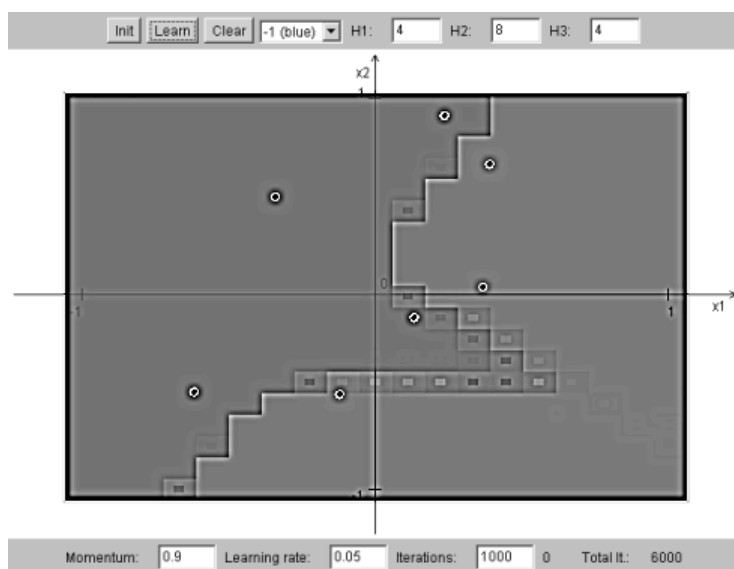


Рисунок 22. Апплет «Обучение персептрона»

Данный апплет является развитием предыдущего, в нем реализовано правило обучения однослойного персептрона (рис. 22). С его помощью можно обучить персептрон, чтобы тот действовал подобно логическому элементу. Персептрон может вычислять или аппроксимировать большинство логических функций двух аргументов. Апплет наглядно демонстрирует «Проблему исключаящего ИЛИ», а также возможный способ ее решения.

### 3.2.3. Многослойный персептрон



Данный апплет моделирует работу многослойного персептрона. Пользователь может задать структуру сети, параметры обучения (момент, скорость, количество итераций), обучающее множество. Выходные значения нейронов лежат в диапазоне  $[-1; 1]$ . После обучения сети пользователь может видеть, как разделяются точки, относящиеся к различным классам (рис. 23).

Рисунок 23. Апплет «Многослойный персептрон»

### 3.2.4. Аппроксимация функций

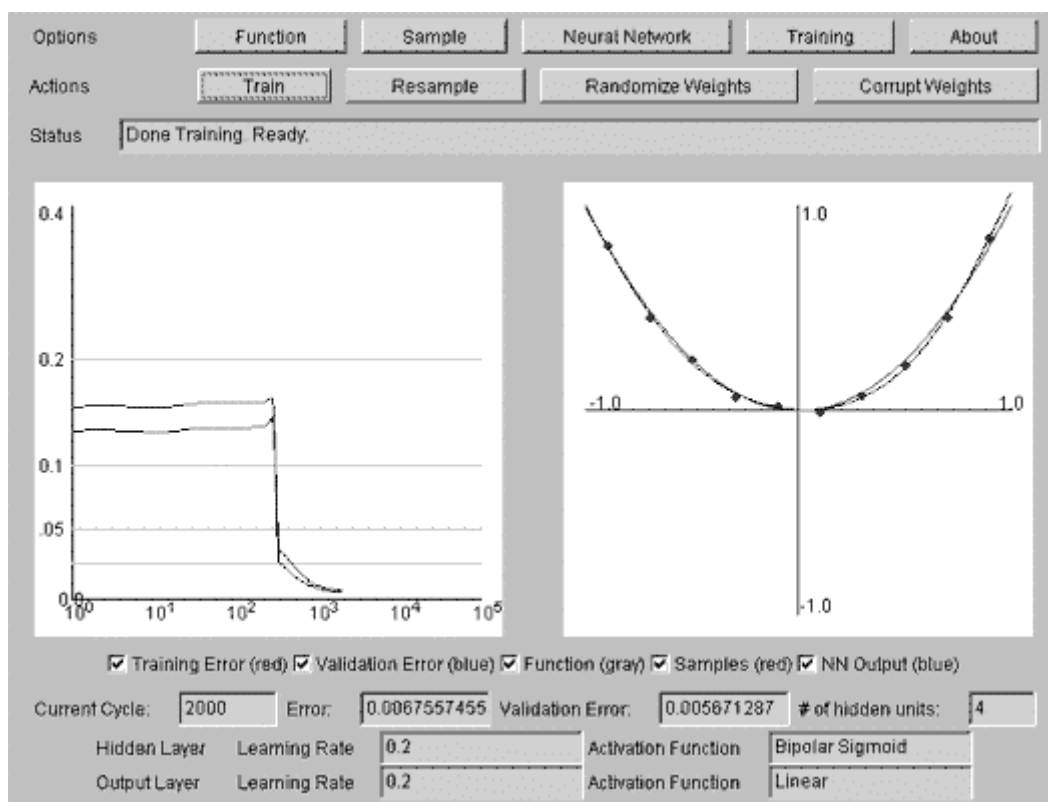


Рисунок 24. Апплет «Аппроксимация функций»

Данный апплет может быть использован для демонстрации использования многослойных персептронов, обучаемых методом обратного распространения ошибки, для аппроксимации функций (рис. 24). Пользователь может выбрать функцию, которую нужно аппроксимировать, выбрать количество обучающих пар, тип активационных функций нейронов, размер сети и скорость обучения.

Апплет является хорошей демонстрацией способностей нейронных сетей к обобщению.

### 3.2.5. Аппроксимация РБФ-сетью

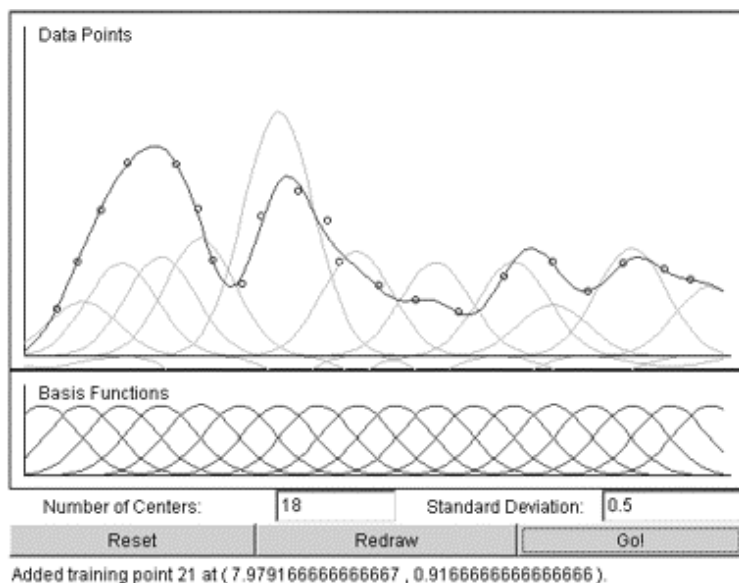


Рисунок 25. Апплет «Аппроксимация РБФ-сетью»

Данный апплет демонстрирует способности радиально-базисных ИНС к аппроксимации функций.

Сначала пользователь должен задать набор точек-примеров для обучения, которые являются реализациями некоторой кривой. Затем пользователь задает количество равномерно распределенных центров (гауссианы).

После обучения сеть становится способна аппроксимировать функцию между указанными точками-примерами (рис. 25).

### 3.2.6. Прогнозирование временного ряда

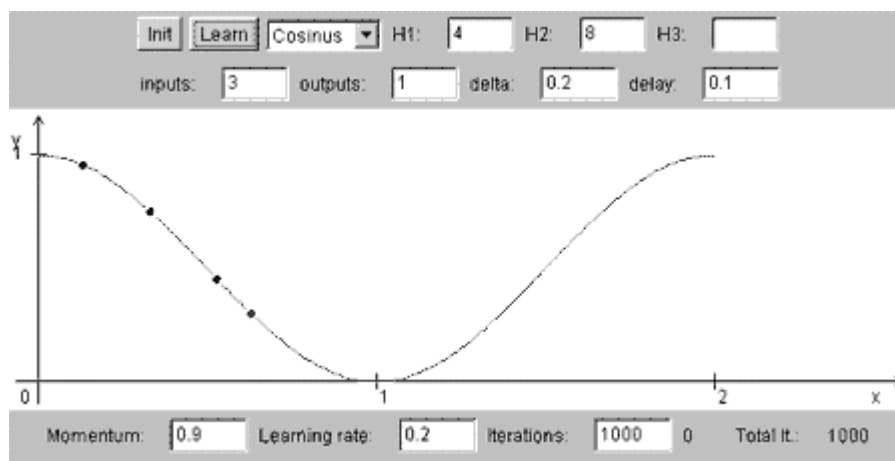


Рисунок 26. Апплет «Прогнозирование временного ряда»

Данный апплет демонстрирует способности многослойных персептронов к прогнозированию (рис. 26). Он позволяет задать входной сигнал, который будет использован для прогнозирования. Пользователь может выбрать количество входов сети, количество нейронов в каждом слое, а также интервал между входными точками и прогнозируемыми выходными значениями. Апплет позволяет наблюдать интересные свойства нейросетевого прогнозирования.

### 3.2.7. Распознавание символов

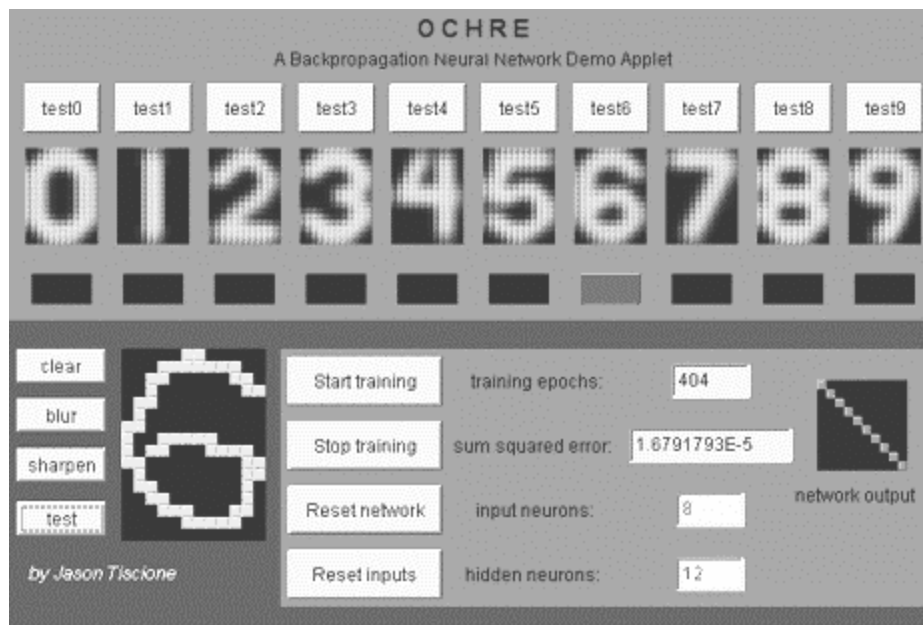


Рисунок 27. Апплет «Распознавание символов»

Данный апплет может быть использован для решения одной из классических задач искусственного интеллекта – распознавания образов. Пользователь может воспользоваться как предустановленным обучающим набором (цифры от 0 до 9), так изменить набор на собственный. После обучения сети, можно проверить ее способности к распознаванию: пользователь рисует некоторый символ в специальном окне, а апплет отображает, к какому классу сеть относит этот символ (рис. 27).

### 3.2.8. Ассоциативная память

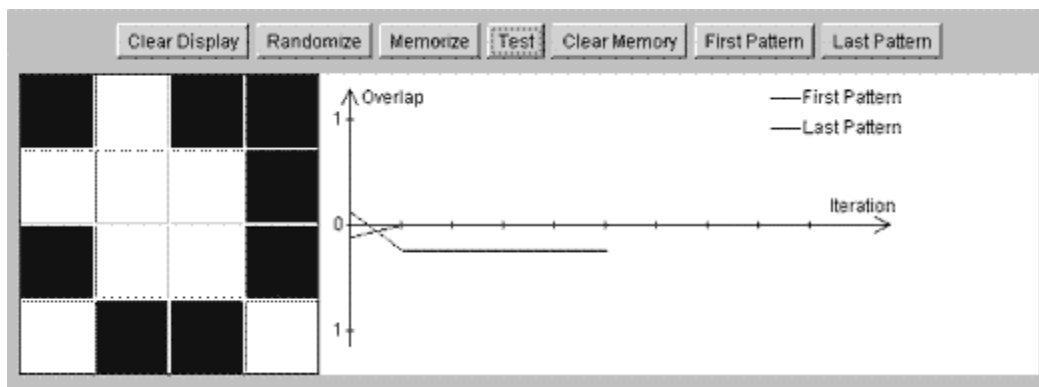


Рисунок 28. Апплет «Ассоциативная память»

Данный апплет является примером реализации ассоциативной памяти по модели Хопфилда (рис. 28). Нейроны сопоставлены пикселям, и принимают значения  $-1$  (выключен) и  $+1$  (включен). Сеть хранит некоторое количество пиксельных узоров (pixel patterns). В процессе эволюции, сеть из произвольного начального значения переходит в одно из сохраненных состояний, которое ближе всего к начальному.

### 3.2.9. Самоорганизующаяся сеть

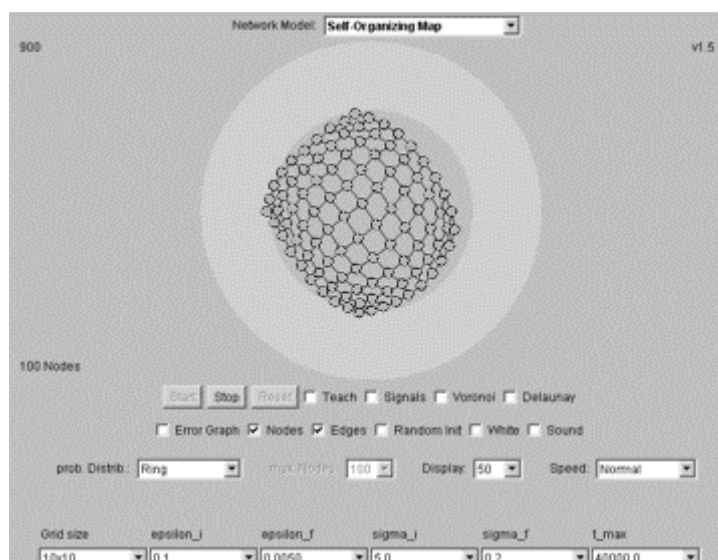


Рисунок 29. Апплет «Самоорганизующаяся сеть»

Данный апплет реализует несколько методов, относящихся к конкурентному обучению. Пользователь может поэкспериментировать с этими методами, используя различные варианты исходных данных, и наблюдая процесс обучения (рис. 29).

Апплет позволяет моделировать следующие алгоритмы:

- Расширяющийся нейронный газ (Growing Neural Gas – Fritzke);
- Жесткое конкурентное обучение (Hard Competitive Learning – стандартный алгоритм);
- Нейронный газ (Neural Gas – Martinetz и Schulten);
- Нейронный газ с конкурентным обучением Хебба (Neural Gas with Competitive Hebbian Learning – Martinetz and Schulten);
- Конкурентное обучение Хебба (Competitive Hebbian Learning – Martinetz and Schulten);
- LBG (Linde, Buzo, Gray);
- Расширяющаяся сетка (Growing Grid – Fritzke);
- Самоорганизующаяся карта (Self-Organizing Map – Kohonen);

### 3.3. Выводы

В данной главе был рассмотрен один из курсов, внесенных в разработанную информационно-справочную систему – адаптированный для использования в ДО специальный курс «Нейронные сети».

Частью процесса адаптации стало включение в курс демонстрационных апплетов, которые покрывают большинство рассматриваемых в лекциях тем. В настоящей главе были приведены описания основных возможностей использованных апплетов.



## Заклучение

Итогом работы явилось создание универсальной информационно-справочной системы поддержки дистанционного образования, а также адаптированного для использования в дистанционном обучении специального курса «Нейронные сети», который был создан на основе традиционного УМК.

Разработанная информационная система может использоваться как в локальных, так и в глобальных сетях (Intranet/Internet); она предоставляет преподавателям и студентам широкие возможности по размещению и поиску материалов. Система поддерживает работу с ключевыми словами, включает в себя динамический глоссарий, существенно упрощающий поиск нужных материалов.

Созданная система поддерживает возможность внедрения в учебные статьи интерактивных элементов, которые повышают наглядность материала, позволяют студентам проводить самостоятельные вычислительные эксперименты. Все эти факторы, безусловно, положительно влияют на восприятие материала и, следовательно, повышают эффективность обучения.

В рамках работы также был развит и адаптирован для дистанционного обучения специальный курс «Нейронные сети», читаемый на факультете радиофизики и электроники БГУ. За счет использования интерактивных элементов (java-апплетов) курс стал более наглядным и информативным.

Следует отметить, что созданная система, в отличие от большинства аналогов, имеет более низкую стоимость и не требует от пользователей приобретения какого-либо специального оборудования или программного обеспечения. Для использования системы достаточно интернет-браузера, установленного на любом современном компьютере (например, Internet Explorer или Opera).

Перспективным направлением развития работы представляется включение в ее состав экспертной системы с базой знаний на основе семантических сетей. [39]

Результаты дипломной работы докладывались на научных и научно-методических конференциях.

- Международная научно-методическая конференция «Высшее техническое образование: проблемы и пути развития». 17–18 марта 2004, БГУИР, Минск.
- 61-я конференция студентов и аспирантов БГУ. 18–19 мая 2004, Минск.
- XIV Symposium on Photonics and Web Engineering. 26–30 May 2004, Wilga, Poland.

## Список использованных источников

---

- 1 Красовский В. И. Дистанционное обучение: от эксперимента до внедрения. //Высшая школа. 2004. № 1. С. 45-46.
- 2 Neural Nets by Kevin Gurney  
(<http://www.shef.ac.uk/psychology/gurney/notes/contents.html>)
- 3 Терехов С. А., Лекции по теории и приложениям искусственных нейронных сетей.  
(<http://alife.narod.ru>)
- 4 Миркес Е. М. Учебное пособие по курсу «Нейроинформатика». //Красноярск, 2002.  
(<http://www.softcraft.ru/neuro/ni/p01.shtml>)
- 5 Java Demonstrations of Neural Net Concepts.  
(<http://neuron.eng.wayne.edu/software.html>)
- 6 Кундас С. П., Красовский В. И., Тавгень И. А. Состояние и тенденции развития дистанционного обучения в Республике Беларусь. //Материалы международной научно-методической конференции «Высшее техническое образование: проблемы и пути развития», БГУИР, Минск, 2004. С. 267–270.
- 7 Смирнов С. Д. Педагогика и психология высшего образования: от деятельности к личности. //Учеб. пособие для слушателей фак-тов и ин-тов повышения квалификации, преподавателей вузов и аспирантов. – М.: АспектПресс, 1995. – С.33-36.
- 8 Румянцев В. В., Милославский А. Г., Каргин А. А. Роль информационных технологий в развитии современного образования. //Конгресс конференций «Информационные технологии в образовании». (<http://www.ito.su/2002/I/2/I-2-76.html>)
- 9 Основные понятия системы дистанционного образования. //Центр дистанционного обучения при БГЭУ. (<http://cdo.bseu.by/about/DA.htm>)
- 10 Система дистанционного обучения «Прометей». (<http://www.prometeus.ru>)
- 11 Wiesner P. Distance Education: Rebottling or a New Brew? //Proceedings of the IEEE, Vol. 88, No. 7, July 2000.
- 12 Лазько Л. Я., Тригубова Л. А. Пути совершенствования технологии образования. //Материалы международной научно-методической конференции «Высшее техническое образование: проблемы и пути развития», БГУИР, Минск, 2004. С. 189–190.
- 13 Стрелкова И. Б. Использование технологий дистанционного обучения в системе повышения квалификации библиотечных кадров Республики Беларусь: возможности и перспективы. //Материалы международной научно-практической конференции «Современные информационные технологии и социальная значимость библиотек». ВГУ, Витебск, 2003. (<http://lib.vsu.by/by/konf/str.htm>)
- 14 Блинов И. Н., Таранчук В. Б. О системах дистанционного обучения для локальной и глобальной сетей //Материалы I Международной конференции «Информационные системы и технологии (IST/2002)», – Минск, 2002. – С. 16–17.
- 15 IBA – International Business Alliance. ([www.iba.by](http://www.iba.by))
- 16 Ringwood J. V., Galvin G.. Computer-aided learning in artificial neural networks. //IEEE Trans on Education. Vol. 45, No. 4, 2002. PP. 380–387.
- 17 Neural Networks Research Centre, Helsinki Univ. of Technology, Espoo, Finland.  
(<http://www.cis.hut.fi/research/demos.shtml>)
- 18 Долинер Л. И., Супрун С. В. Проблемы внедрения современных систем дистанционного обучения. //Конгресс конференций «Информационные технологии в образовании». (<http://ito.edu.ru/2003/III/2/III-2-1529.html>)
- 19 Блинов И. Н. Опыт применения СДО в учебном процессе ВУЗа. //Материалы международной научно-методической конференции «Высшее техническое образование: проблемы и пути развития», БГУИР, Минск, 2004. С. 165–166.
- 20 CHAOS: A Recipe for Success. ([www.velocitystorm.com/servicessolutions/chaos.pdf](http://www.velocitystorm.com/servicessolutions/chaos.pdf))

- 
- 21 Новиков Л. Введение в Rational Unified Process. (<http://rusdoc.ru/material/raznoe/rup-0.zip>)
  - 22 Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. //Второе издание. Rational Санта-Клара, Калифорния. Перевод с англ. под редакцией И. Романовского и Ф. Андреева
  - 23 Змеев О. А. Разработка и стандартизация программных средств и информационных технологий. //Конспект лекций. Анжеро-Судженский филиал КемГУ. 2002г. (<http://www.nkfi.ru/doc/risp/lec012.asp.htm>)
  - 24 Трофимов С. Рабочие процессы RUP и диаграммы UML. ([http://www.caseclub.ru/articles/rup\\_uml.html](http://www.caseclub.ru/articles/rup_uml.html))
  - 25 Крачтен Ф. Введение в Rational Unified Process. //Изд. 2-е.- М.: Издательский дом «Вильямс», 2002. - 240 с.: ил.
  - 26 Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения //СПб.: Питер, 2002. - 496 с.: ил.
  - 27 Трофимов С. Унифицированный процесс разработки от Rational Software. //PCWeek/RE, №4/2003. (<http://www.caseclub.ru/articles/rup2.html>)
  - 28 Унгер К. Решите проблемы представления в сервлет-ориентированных приложениях. ([http://www.javable.com/javaworld/11\\_00/02/](http://www.javable.com/javaworld/11_00/02/))
  - 29 Лутковский В. М., Поплетеев А. М., Харкевич В. М.. Разработка учебно-методических комплексов с использованием метода проектов. //Тез. докл. научно-методической конференции. Мн.: БНТУ. 2004.
  - 30 Лутковский В. М. Нейронные сети. //Конспект лекций. Мн.: БГУ. 2003
  - 31 Асаула Р., Евтюхин А., Маковецкий А., Поплетеев А., Лутковский В. Пакет программ для моделирования нейронных сетей. //Материалы международной научно-методической конференции «Высшее техническое образование: проблемы и пути развития», БГУИР, Минск, 2004. С. 158.
  - 32 Corbett F. (<http://home.cc.umanitoba.ca/~umcorbe9/neuron.html>)
  - 33 Corbett F. (<http://home.cc.umanitoba.ca/~umcorbe9/perceptron.html>)
  - 34 Michel O. (<http://diwww.epfl.ch/lami/team/michel/>)
  - 35 Watta P., Hassoun M., Dannug N. (<http://neuron.eng.wayne.edu/bpFunctionApprox/bpFunctionApprox.html>)
  - 36 Tiscione J. (<http://www.geocities.com/SiliconValley/2548/ochre.html>)
  - 37 Wong J. W. (<http://exodus.mit.edu/~jesse/>)
  - 38 Loos H. S. , Fritzke B. (<http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html>)
  - 39 Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. //Учебник. СПб: Питер, 2000. – 384 с.: ил.