

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

УТВЕРЖДАЮ

Зав. кафедрой АСУ, профессор



А.М. Корилов

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Лабораторные работы

Учебно-методическое пособие

направление подготовки бакалавриата: **09.03.01 — Информатика и вычислительная техника**

направленность (профиль) программы: **Программное обеспечение средств вычислительной техники и автоматизированных систем**

Разработчик

доцент кафедры АСУ

В.Г. Резник

Резник В.Г.

Распределенные вычислительные системы. Лабораторные работы по направлению подготовки бакалавриата 09.03.01. Учебно-методическое пособие. – Томск, ТУСУР, 2019. – 28 с.

Учебно-методическое пособие предназначено для проведения лабораторных работ по дисциплинам «Вычислительные системы и сети» и «Распределенные вычислительные системы» для студентов направления подготовки бакалавра: 09.03.01 «Информатика и вычислительная техника» направленности (профиля) программы - «Программное обеспечение средств вычислительной техники и автоматизированных систем».

Оглавление

Введение.....	4
1 Тестирование ПО рабочей области студента.....	5
1.1 Дистрибутив ОС УПК АСУ для проведения лабораторных работ.....	5
1.1.1 <i>Общая структура ПО для проведения лабораторных работ.</i> ..	5
1.1.2 <i>Рабочий стол и инструменты рабочей области.....</i>	6
1.2 Состав учебного материала.....	9
1.2.1 <i>Назначение обязательных учебно-методических пособий.....</i>	10
1.2.2 <i>Исходные тексты учебных проектов.....</i>	10
1.2.3 <i>Отчетность по результатам выполненных работ.....</i>	11
2 Инструментальные средства языка Java.....	12
2.1 Дистрибутив языка Java и среда разработки Eclipse EE.....	12
2.1.1 <i>Инструментальные средства командной строки.....</i>	12
2.1.2 <i>Инструментальные средства Eclipse.....</i>	13
2.2 Базовые средства и ввод-вывод языка Java.....	14
2.2.1 <i>Синтаксис и семантика базовых средств языка Java.....</i>	14
2.2.2 <i>Дополнительные аспекты использования языка Java.....</i>	15
2.3 Сокеты и сетевое ПО языка Java.....	15
2.3.1 <i>Базовые средства пакета java.net.....</i>	15
2.3.2 <i>Сети и многопоточное программирование.....</i>	16
2.4 Технология работы с базами данных.....	17
2.4.1 <i>Типовой пример выборки данных.....</i>	17
2.4.2 <i>Функции и типы данных СУБД Apache Derby.....</i>	18
3 Реализация распределенной системы средствами RMI.....	19
3.1 Технология КЛИЕНТ-СЕРВЕР.....	20
3.1.1 <i>Серверная часть приложения NotePad.....</i>	21
3.1.2 <i>Клиентская часть приложения Example12.....</i>	21
3.2 Технология RMI.....	22
3.2.1 <i>Реализация распределенной системы RmiPad.....</i>	22
3.2.2 <i>Использование утилиты rmic.....</i>	22
4 Web-технологии распределенных систем.....	23
4.1 Технология сервлетов на базе сервера Apache Tomcat.....	23
4.1.1 <i>Apache Tomcat — как контейнер сервлетов.....</i>	24
4.1.2 <i>Дополнительные классы технологии сервлетов.....</i>	24
4.2 Технология JSP для формирования динамических HTML-страниц....	25
4.2.1 <i>Основные возможности JSP-страниц.....</i>	25
4.2.2 <i>Параметры и атрибуты JSP-страниц.....</i>	25
4.3 Шаблон проектирования MVC.....	26
4.3.1 <i>Базовая архитектура шаблона MVC.....</i>	26
4.3.2 <i>Реализация проекта webpad.....</i>	26
Список использованных источников.....	27

Введение

Данное пособие содержит учебно-методический материал для проведения лабораторных работ по дисциплинам «Распределенные вычислительные системы» и «Вычислительные системы и сети», предполагающий уровень подготовки бакалавриата. Теоретический материал для этих дисциплин изложен в учебном пособии [1], содержащем также примеры на языке Java, и в учебно-методическом пособии [2], которое содержит учебный материал по проведению практических занятий. При этом предполагается, что студент ранее не изучал данный язык, но прошел успешное обучение по изучению языков C/C++.

Цель данного пособия — методическое обеспечение процессов выполнения лабораторных работ в объеме 36 академических часов.

Основная задача данного пособия — практическое закрепление навыков программирования на языке Java, которые описаны и демонстрируются в учебном пособии [1] и специализированы в плане изучаемой дисциплины. В силу ограниченности объема теоретических и практических занятий, в тексте данного пособия используются ссылки на полноценный источник по языку Java [3].

В целом, содержание данного пособия привязано к четырем главам базового учебного пособия [1]:

- Введение в теорию вычислительных сетей (глава 1).
- Инструментальные средства языка Java (глава 2).
- Объектные распределенные системы (глава 3).
- Web-технологии распределенных систем (глава 4).

Первой главе посвящена отдельная лабораторная работа «*Тестирование системного и инструментального ПО рабочей области студента*», которая ориентирована на подготовку компьютерной обучающей среды для выполнения последующих лабораторных работ. Она опирается на специальный программный комплекс ОС УПК АСУ, кратко описанный в отдельном методическом пособии [4].

В остальных главах данного пособия выделены подразделы, каждый из которых соответствует содержанию отдельной лабораторной работе, далее детализируемой отдельными пунктами. Общая нагрузка каждой работы дополняется учебным материалом методического пособия [2].

Результаты всех лабораторных работ оформляются общим отчетом, шаблон которого предоставляется студенту в его индивидуальной рабочей области. Материал этого отчета, дополненный вопросами преподавателя, учитывается в оценках промежуточной и итоговой аттестации студента.

1 Тестирование ПО рабочей области студента

Данная глава посвящена первой лабораторной работе по дисциплине «*Распределенные вычислительные системы*» (по плану 2016 года эта дисциплина называлась «*Вычислительные системы и сети*»).

Учебная цель этой работы — подготовка и изучение инструментальных программных средств, необходимых для успешного выполнения учебных заданий по изучаемой дисциплине.

Результатом выполнения данной лабораторной работы должны быть:

- знания о структуре и местоположении компонент дистрибутива ОС УПК АСУ, в ключающего ПО используемых инструментальных средств и ПО личной рабочей области студента;
- знания о составе и местоположении учебного материала, используемого в учебном процессе данной дисциплины;
- оформленные в отчетной форме результаты выполненной студентом работы.

1.1 Дистрибутив ОС УПК АСУ для проведения лабораторных работ

Общее описание дистрибутива ОС УПК АСУ опубликовано в учебно-методическом пособии [4], размещенном в электронном виде на портале ТУСУР. С этим пособием студент уже знакомился в процессе изучения дисциплины «*Операционные системы*».

В данном подразделе лабораторной работы излагается учебный материал как дополняющий источник [4], так и раскрывающий особенности применения ПО ОС УПК АСУ непосредственно к изучаемой дисциплине.

Все дополнения, касающиеся непосредственного выполнения лабораторных работ, размещения студентов по рабочим местам компьютерного класса и последовательности выполнения учебных заданий выполняются преподавателем во время проведения учебного занятия.

1.1.1 Общая структура ПО для проведения лабораторных работ

Полный комплект дистрибутива ОС УПК АСУ размещается в отдельном каталоге **asu64upk** корня файловой системы MS Windows, известном как диск **C:**. Такое размещение подробно описано в [4, подраздел 1.1] и полностью соответствует используемой архитектуре учебного ПО, установленного в учебных классах кафедры АСУ или на личные компьютеры студентов.

Студент должен убедиться в наличии на диске **C:** каталога **asu64upk**, а также его файловой структуры согласно данным таблицы 1.1.

Таблица 1.1 — Назначение компонент дистрибутива ОС УПК АСУ

Компонента	Назначение компоненты
boot	Каталог размещения ядра ОС и временной файловой системы.
opt	Каталог размещения дополнительного инструментального ПО. Следует убедиться в наличии файла eclipseEEoxygen.sfs .
themes	Каталог размещения рабочей области студента после подключения его личного архива.
upkasu	Каталог размещения ПО ОС УПК АСУ.
upk_asu.pdf	Файл учебно методического пособия, который следует использовать вместо источника [4].

Здесь следует обратить особое внимание на:

- наличие учебно-методического пособия **upk_asu.pdf**, которым следует пользоваться в обязательном порядке;
- наличие файла **eclipseEEoxygen.sfs** — являющимся дистрибутивом среды разработки приложений на языке Java.

Следующий элемент контроля связан с подготовкой и составом ПО на личном **flashUSB** студента. Общие вопросы подготовки этого инструмента изложены в источнике [4, глава 2] и освоены им во время лабораторных работ по дисциплине «Операционные системы».

Студент должен убедиться в наличии каталога **asu64upk** в корне файловой системы личного flashUSB и структуры каталогов, представленной в таблице 1.1.

Дополнительно студент должен:

- взять у преподавателя файл **rvs-home.ext4fs.gz** и поместить его на личный flashUSB в каталог **/asu64upk/themes**;
- проверить загрузку ОС УПК АСУ с личного flashUSB;
- проверить подключение темы **rvs** к домашнему каталогу пользователя **upk**;
- подключиться к сеансу пользователя **upk** и выполнить проверки, описанные в следующем пункте данной лабораторной работы.

Положительное завершение всех выполненных проверок считается конечным результатом работ по данному пункту учебно-методического пособия.

1.1.2 Рабочий стол и инструменты рабочей области

На рисунке 1.1 показано изображение рабочего стола пользователя **upk** для дисциплины «Распределенные вычислительные системы». По общей традиции методики преподавания, на рабочем столе находятся значки:

- **Учебный материал** — ссылка на каталог с файлами учебных материалов по дисциплине (см. подраздел 1.2);
- **upk_asu.pdf** — ссылка на обновленный документ [4];
- **Отчет.doc** — ссылка на шаблон личного отчета студента;

- **EclipseEE** — значек запуска среды разработки Eclipse EE (см. далее более подробное пояснение).



Рисунок 1.1 — Изображение рабочего стола пользователя urpk

Для проверки полной работоспособности среды разработки **Eclipse EE** необходимо запустить виртуальный терминал рабочего стола с файловым менеджером **Midnight Commander**, перейти в каталог `~/bin`, как показано на рисунке 1.2, и запустить сценарий монтирования **mountEE**. В результате, дистрибутив среды разработки будет подключен (подмонтирован) к каталогу `/opt/eclipseEE` и станет готов к запуску.

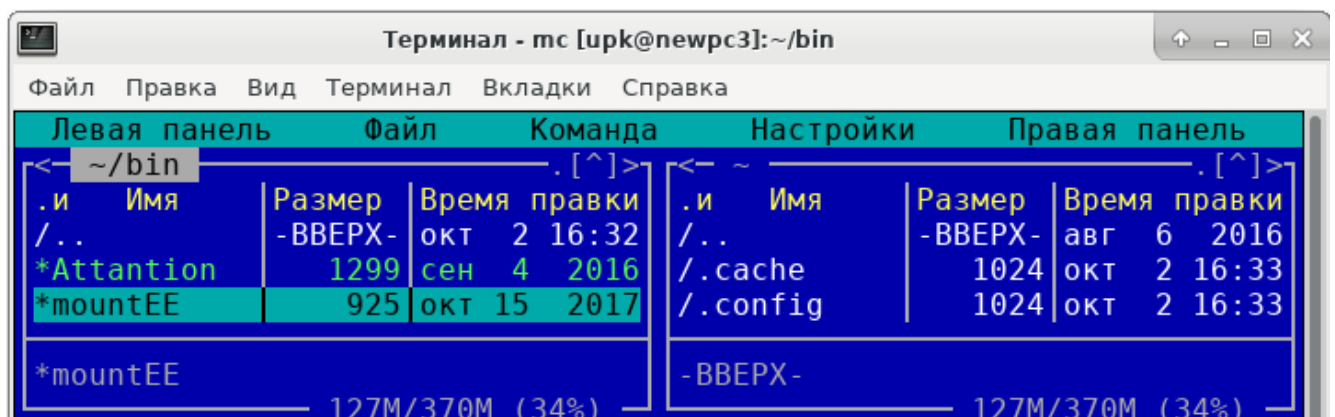


Рисунок 1.2 — Подключение среды разработки Eclipse EE

При первом запуске среды разработки, система загрузится с заставкой, показанной на рисунке 1.3.

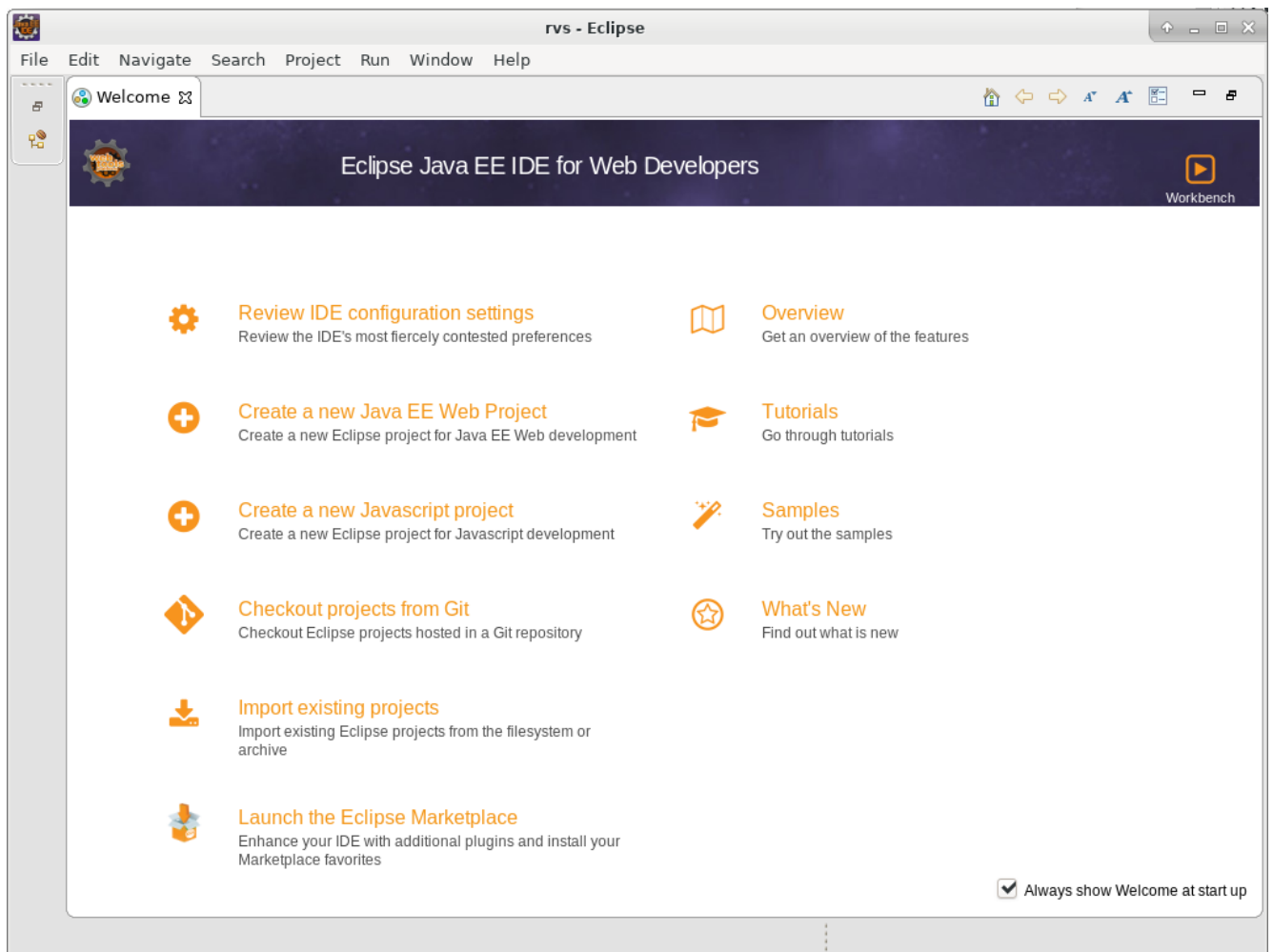


Рисунок 1.3 — Первый запуск среды разработки Eclipse EE

Правильно запущенная среда разработки **Eclipse EE** должна создать в рабочей директории **\$HOME** пользователя **upk** каталог **~/rvs** с необходимыми первоначальными настройками. Студент должен непосредственно убедиться в правильности создания рабочего каталога поскольку все учебные примеры пособий [1 и 2] приведены с учетом указанных настроек.

Дополнительно, студент должен убедиться в наличии в рабочей директории пользователя **upk** следующих каталогов с программным обеспечением:

- **~/derby** — каталог с дистрибутивом СУБД **Apache Derby**;
- **~/tomcat** — каталог с дистрибутивом **Apache Tomcat**.
- **~/src/rvs** — рабочая директория последующих проектов;
- **~/lib** — рабочая директория последующих проектов.

Наличие в домашнем каталоге пользователя **upk**, перечисленных выше каталогов и программного обеспечения, говорит о правильной настройке учебной среды и считается конечным результатом работ по данному пункту.

1.2 Состав учебного материала

Рабочая область пользователя *urk* содержит, кроме инструментальных средств, набор файлов учебного материала, который полностью сконцентрирован в каталоге *\$HOME/Документы*, показанном на рисунке 1.4.

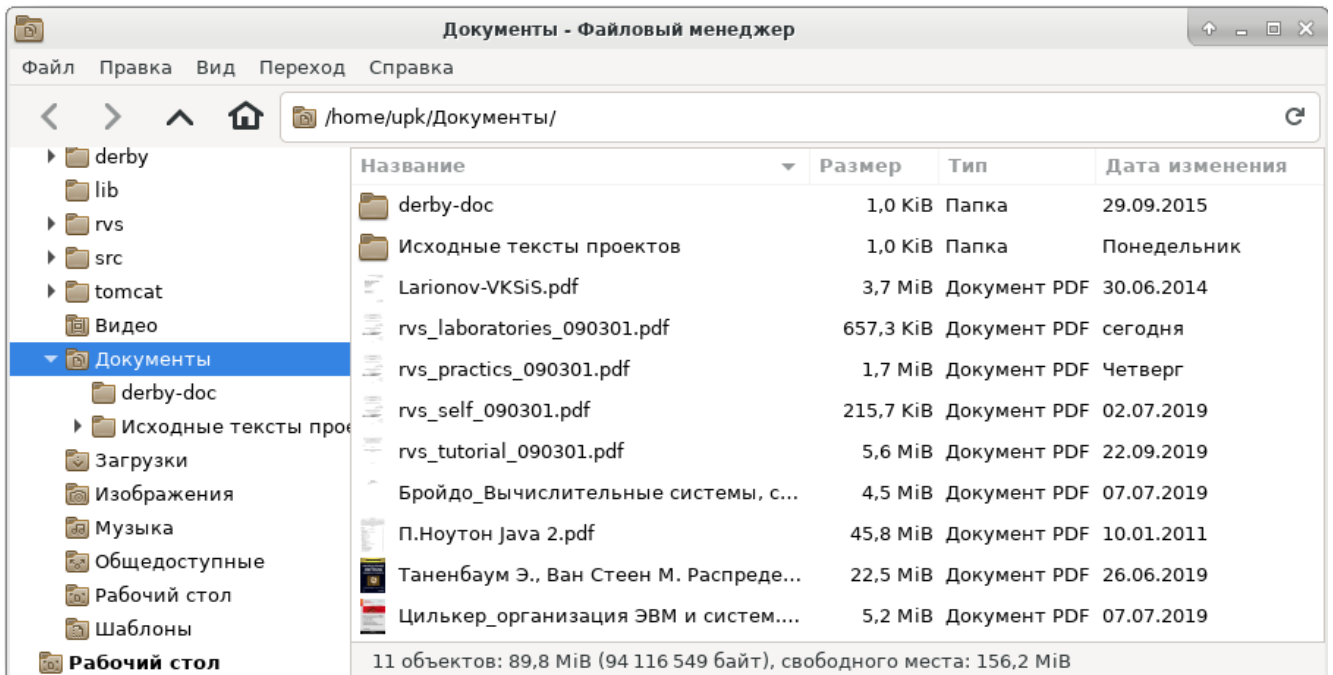


Рисунок 1.4 — Список учебных материалов по изучаемой дисциплине

В плане изучения, представленный учебный материал разделен на три категории:

- *обязательный* к изучению — файлы, начинающиеся с префикса *rvs* и составляющие официальную часть учебно-методического обеспечения (УМО) изучаемой дисциплины; их назначение описано в пункте 1.2.1;
- *вспомогательный* для практического использования — представлен файлами каталогов, которые кратко описаны в пункте 1.2.2;
- *расширенный* для развития — файлы монографий и учебников, на основе которых был подготовлен учебный материал УМО.

Студент должен осознавать, что учебный материал данной дисциплины подготовлен на основе ряда фундаментальных работ и учебных пособий, которые в сумме составляют не только большое количество печатных страниц, но и знания, которые выходят за рамки программы обучения бакалавра. Тем не менее, указанные работы авторитетных авторов содержат информацию, необходимую студенту для специализации или более углубленного изучения предмета. С целью аннотации этих работ, перечислю их краткое назначение:

- ***Larionov-VKSiS.pdf*** — довольно устаревший учебник (1987 год) Ларионова А.М., Майорова С.А. и Новикова Г.И., озаглавленный «Вычислительные

комплексы, системы и сети», в котором предложена и описана классификация систем обработки данных (СОД).

- **Бройдо_Вычислительные системы, сети и телекоммуникации.pdf** — достаточно современный учебник для вузов (2011 год), в котором достаточно полно раскрыты объявленные в названии вопросы.
- **П.Ноутон Java 2.pdf** — полное руководство по языку Java.
- **Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы (2003).pdf** — фундаментальный труд Э. Таненбаума, раскрывающий всю проблематику распределенных систем.
- **Цилькер_Организация ЭВМ и систем.pdf** — базовый учебник для вузов (2011 год) ориентированный в основном на описание аппаратной части вычислительных систем.

1.2.1 Назначение обязательных учебно-методических пособий

К обязательным учебно-методическим пособиям относятся следующие файловые источники, составляющие УМО данной дисциплины и представленные на рисунке 1.3:

- **rvs_laboratories_090301.pdf** — учебно-методическое пособие по выполнению лабораторных работ, которое сейчас читает студент;
- **rvs_practics_090301.pdf** — учебно-методическое пособие по выполнению практических занятий, дополняющий пособие по проведению лабораторных работ; представлен в данном пособии как источник [2];
- **rvs_self_090301.pdf** — учебно-методическое пособие по выполнению самостоятельной и индивидуальной работе студента; содержит перечень теоретического и практического материала по всем пяти главам изучаемой дисциплины; для каждой изучаемой главы приведен список вопросов, которые входят в состав экзаменационного или иного оценочного контроля знаний студентов;
- **rvs_tutorial_090301.pdf** — основное учебное пособие по изучаемой дисциплине, обозначаемый в данном пособии как источник [1]; содержит пять глав, в конце каждой из которых представлены вопросы для самостоятельной проверки знаний; демонстрационные примеры 2-й, 3-й и 4-й глав составляют основное содержание восьми лабораторных работ, описанных в данном пособии.

1.2.2 Исходные тексты учебных проектов

Каталог *~/Документы* содержит ряд вспомогательных материалов, которые призваны обеспечить успешное выполнение студентом всех лабораторных заданий. К ним относятся:

- файлы каталога **derby-doc**, где расположена официальная документация по СУБД Apache Derby, а также три русскоязычных файла по этой тематике;

- файлы каталога **Исходные тексты проектов**, где расположены исходные тексты всех примеров, представленных в пособиях [1] и [2]. Все исходные тексты примеров сгруппированы в каталогах, относящихся к 15-ти проектам, показанных на рисунке 1.5.

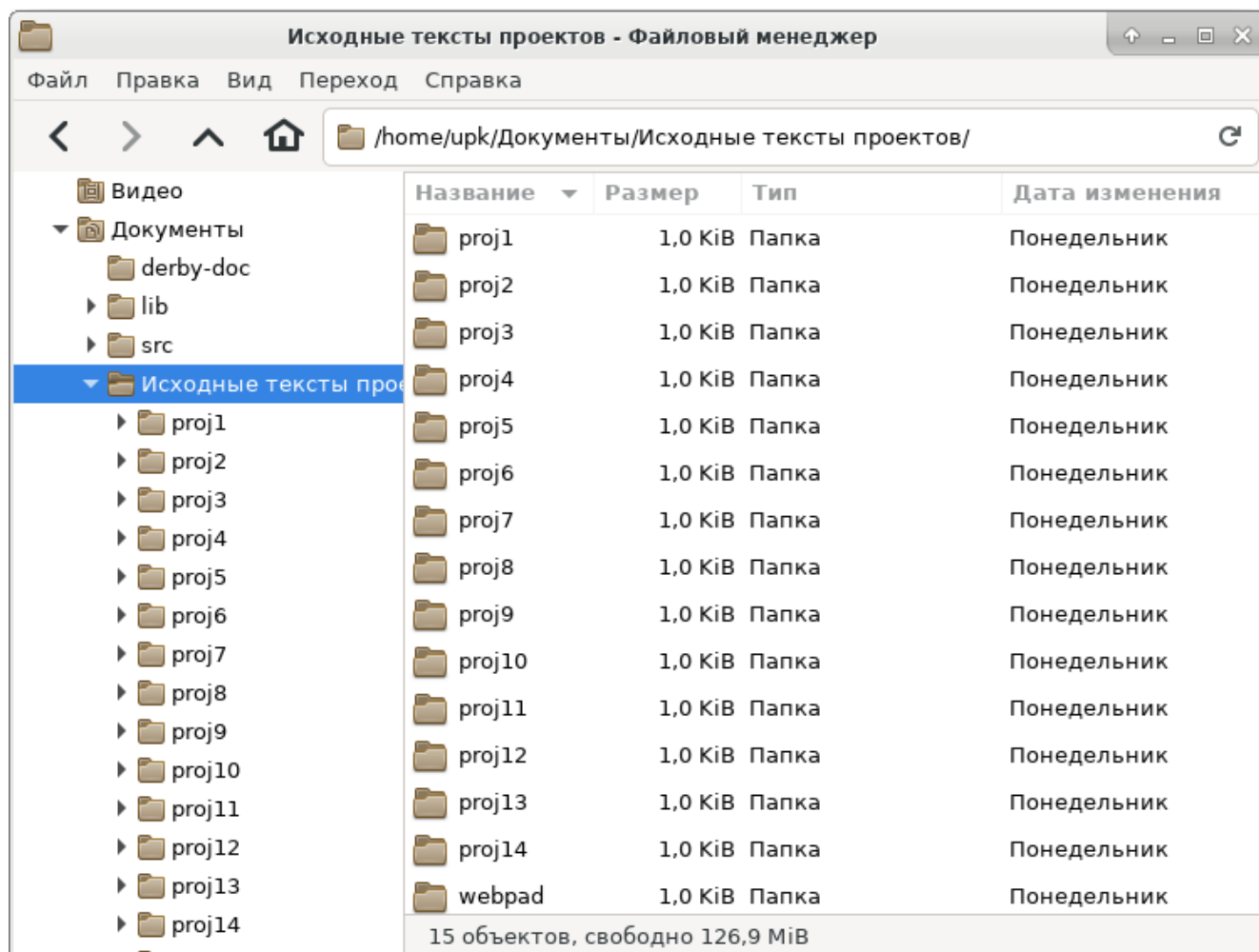


Рисунок 1.5 — Каталоги проектов с исходными текстами примеров, описанных в источниках [1] и [2]

Следует обратить внимание, что сами каталоги не являются проектами среды разработки Eclipse EE, а исходные тексты примеров представлены с целью избежания ошибок, которые возникают если их копировать из учебников [1] и [2]. Более детальное их применение описывается в методиках соответствующих работ.

1.2.3 Отчетность по результатам выполненных работ

По всем выполненным лабораторным работам студент заполняет единый отчет, представленный на рабочем столе пользователя *upk* в виде значка файла **Отчет.doc**. Содержание этого отчета, дополненное вопросами преподавателя оценивается в период контрольных точек обучения студента.

2 Инструментальные средства языка Java

Данная глава охватывает учебно-методическое описание четырех лабораторных работ основанных на материалах учебного пособия [1, глава 2] и методического пособия [2, глава 1]. Они составляют своеобразный экспресс-курс по изучению языка Java в объеме, соответствующем стандартной платформы *J2SE*.

Общая целевая направленность этой серии работ — практическое закрепление теоретических сведений о языке Java, достаточных для дальнейшего применения их в процессе изучения специальных разделов, посвященных тематике распределенных систем.

В процессе выполнения данной части лабораторных работ студент заполняет соответствующие подразделы второй главы индивидуального отчета, представленного на рабочем столе пользователя *upk* файлом *Отчет.doc*. Название каждого подраздела данной главы соответствует названию отдельной лабораторной работы.

2.1 Дистрибутив языка Java и среда разработки Eclipse EE

Вторая лабораторная работа, названная «*Дистрибутив языка Java и среда разработки Eclipse EE*», полностью основана на материале второй главы учебного пособия [1, подраздел 2.1 «Общее описание инструментальных средств языка»]. Ее выполнение условно разделено на две части:

- инструментальные средства командной строки;
- инструментальные средства Eclipse.

Приступив к выполнению лабораторной работы, следует познакомиться с расположением используемого дистрибутива языка Java в файловой системе используемой ОС. В начале главы 2 и подраздела 2.1 даются соответствующие пояснения, касающиеся различных платформ языка, поставок дистрибутива и их базовое расположение в ОС УПК АСУ и MS Windows.

После закрепления общего материала, следует приступить к выполнению заданий по выделенным пунктам учебного материала.

2.1.1 Инструментальные средства командной строки

Выполнение этой части работ начинается с изучения и практической апробации примеров, описанных в пункте 2.1.1 учебного пособия [1]. Студент должен освоить:

- назначение и применение необходимых параметров, входящих в состав переменных среды ОС;
- написание простейших программ на языке Java, используя средства тексто-

- вых редакторов ОС;
- технику компиляции Java-программ средствами утилиты **javac**;
- запуск простейших Java-программ средствами виртуальной машины **java**.

Пакетная организация приложений языка Java описана в источнике [1, пункт 2.1.2]. Студент должен освоить:

- назначение базовых пакетов языка;
- оператор пакетной организации приложений Java;
- особенности компиляции и запуска примеров с пакетной организацией приложений;
- организацию и создание JAR-архивов приложений с помощью утилиты **jar**.

2.1.2 Инструментальные средства Eclipse

Выполнение этой части работ предполагает, что студент освоил практически учебный материал первой лабораторной работы и способен:

- подключить ПО дистрибутива Eclipse EE средствами специального сценария **~/bin/mounEE**;
- запустить среду разработки Eclipse EE, используя специальный значек, расположенный на рабочем столе пользователя **upk**.

Учебный материал данной части лабораторной работы изложен в учебном пособии [1, пункт 2.1.3]. Студент должен освоить:

- правильную настройку инструментария Eclipse EE, предполагающую, что местоположение всех проектов должно находиться в каталоге **~/rvs**;
- правила создания проектов и классов языка Java с учетом их пакетной организации;
- создание текста простейших приложений, их запуск и контроль исполнения в среде Eclipse EE;
- местоположение созданных исходных текстов и классов проектов в файловой системе ОС, с учетом пакетной организации приложений;
- технологию создания JAR-архивов приложений и их расположение вне файлового пространства проектов среды разработки.

В процессе выполнения лабораторной работы, студент должен учесть, что исходные тексты всех используемых примеров учебных пособий [1] и [2] размещены в каталоге **~/Документы/Исходные тексты проектов**. Они сгруппированы по именам проектов и находятся в каталогах с учетом их пакетной организации. Правильное использование этих текстов предполагает, что:

- сначала, студент создает в проекте необходимый класс средствами среды разработки Eclipse EE;
- затем, находит файл с исходным тестом нужного класса, открывает его редактором **mousepad** и копирует его по месту уже созданного класса в среду разработки Eclipse EE.

2.2 Базовые средства и ввод-вывод языка Java

Третья лабораторная работа, имеющая название «*Базовые средства и ввод-вывод языка Java*», содержит достаточно объемный материал, касающийся синтаксиса, семантики и различных аспектов применения изучаемого языка, которым можно посвятить отдельную самостоятельную дисциплину. Тем не менее, методика настоящей дисциплины опирается на достаточно высокую профессиональную подготовку студентов в плане уже изученных языков *C* и *C++*, что вызывает уверенность в успешном освоении базовых основ языка Java, а также в успешном выполнении данной лабораторной работы.

Учебный материал данного подраздела сформирован в виде двух пунктов, в которых изложены задания разного уровня сложности:

- синтаксис и семантика базовых средств языка Java — ориентирована на уровень начального освоения языка;
- дополнительные аспекты использования языка Java — предполагают успешное освоение материала предыдущего уровня.

Указанная выше учебная нагрузка на качество освоения изучаемого материала варьируется преподавателем — в зависимости от объема выделенного учебного времени, что определяется соответствующими учебными планами и рабочей программой.

2.2.1 Синтаксис и семантика базовых средств языка Java

Весь учебный материал данного пункта лабораторной работы изложен в подразделах 2.2, 2.3 и 2.4 учебного пособия [1]. Студент последовательно читает указанный материал и выполняет примеры, описанные в нем.

Особое внимание следует уделить учебным примерам подраздела 2.4, посвященному потокам ввода-вывода языка Java. Студент должен усвоить, что:

- ввод и вывод языка Java осуществляется специальными отдельными классами, предназначенными для этого;
- классы стандартного и общего ввода-вывода принадлежат разным пакетам дистрибутивов ПО языка;
- общий ввод-вывод обеспечен достаточно большим набором специализированных классов различного прикладного назначения.

В результате выполненных заданий по данному пункту лабораторных работ, студент должен в совершенстве владеть методами следующих основных классов:

- ***InputStream*** — основной класс объектов потокового ввода информации;
- ***OutputStream*** — основной класс объектов потокового вывода информации;
- ***File*** — основной класс информационного обеспечения обработки файлов.

2.2.2 Дополнительные аспекты использования языка Java

Весь учебный материал данного пункта лабораторной работы изложен в подразделах 1.1 и 1.2 учебно-методического пособия [2]. Состав и объем выполняемых по этой части работ определяется преподавателем, в зависимости от планируемого объема почасовой нагрузки, выделенной учебным планом и рабочей программой обучения.

Методика выполнения этой части лабораторной работы — аналогична предыдущей, описанной в пункте 2.2.1:

- студент последовательно читает указанный материал учебно-методического пособия [2] и выполняет примеры, описанные в нем;
- выполненные примеры проектов фиксирует в соответствующем подразделе личного отчета.

2.3 Сокеты и сетевое ПО языка Java

Четвертая лабораторная работа, выполняемая под официальным названием «Сокеты и сетевое ПО языка Java», призвана расширить навыки студента в плане программирования сетевых приложений, инструментально обеспечиваемые программными средствами пакета **java.net**. Тем не менее, эта часть обучения уже напрямую связана с основной тематикой изучаемой дисциплины, поскольку распределенные системы, в первую очередь, ориентированы на сетевые приложения.

Как и в предыдущих двух работах, учебные задания выполняемой лабораторной работы разделены на две части:

- базовые средства пакета **java.net**;
- сети и многопоточное программирование.

2.3.1 Базовые средства пакета java.net

Базовым средствам пакета **java.net** языка Java посвящен подраздел 2.5 учебного пособия [1]. Сам подраздел разделен на пять пунктов, описывающих:

- методы классов **InetAddress**, **URL** и **URLConnection**, связанные с объектами адресации сети для стека протоколов TCP/IP;
- классы сокетов **Socket** и **ServerSocket** — предназначенные для синхронного взаимодействия по протоколу **TCP**;
- классы сокетов **DatagramSocket** и **DatagramPacket** — предназначенные для асинхронного взаимодействия по протоколу **UDP**;
- простейшую задачу, реализующую ПО клиента и сервера на базе протокола **TCP**.

В плане методического обеспечения, студенту рекомендуется последовательно читать и выполнять примеры пунктов подраздела 2.5 учебного пособия [1],

фиксируя результаты достижений в соответствующем подразделе личного отчета.

Особое внимание студенту следует уделить на правила использования классов *InputStream* и *OutputStream*, а также проблемы синхронизации приложений клиента и сервера.

2.3.2 Сети и многопоточное программирование

Учебный материал этой части лабораторной работы изложен в подразделах 1.3 и 1.4 учебно-методического пособия [2]. Здесь изучаются программные средства языка Java для реализации многопоточных приложений.

Сами средства многопоточного программирования реализованы в программном пакете *java.lang*, поскольку относятся к системным средствам операционных систем. Они предоставляют две методики программирования приложений:

- методика, основанная на использовании интерфейса *Runnable*;
- методика, основанная на использовании класса *java.lang.Thread*.

Во многом, используемые методики являются эквивалентными по своим функциональным возможностям, но использование интерфейса *Runnable* является более простым. В любом случае, обе методики связаны с переопределением метода *run()*, создаваемых многопоточных объектов, с технологией которой следует внимательно разобраться студенту.

Другой аспект данной части лабораторной работы связан с применением многопоточных приложений к задачам технологии «клиент-сервер». На примерах уже рассмотренных в предыдущем пункте классов *TCPServer* и *TCPClient*, приводится модификация приложения-клиента с использованием методики интерфейса *Runnable*.

Общая методика выполнения этой части лабораторной работы продолжает оставаться неизменной и связана с последовательным изучением материала подразделов 1.3 и 1.4 учебно-методического пособия [2] и реализацией приведенных там примеров. Результаты программирования должны быть аккуратно зафиксированы в личном отчете клиента.

Учитывая все возрастающий объем и сложность исходных тестов изучаемых студентом примеров, настоятельно рекомендуется использовать файлы исходных текстов программ, размещенные в каталоге *~/Документы* рабочей области пользователя *upk*. Время, которое будет экономить студент, следует направить на изучение и исследование алгоритмов реализуемых программ.

2.4 Технология работы с базами данных

Пятая лабораторная работа выполняется под официальным названием «Технология работы с базами данных». Она начинает серию примеров, которые напрямую связаны с распределенными приложениями, поскольку демонстрируют взаимодействие с внешними, по отношению к самому приложению, системами. Таковыми внешними системами являются СУБД, реализуемые как самостоятельные инструментальные приложения, работа с которыми поддерживается программными средствами пакета *java.sql*.

Этой теме, в учебнике [1], полностью посвящен подраздел 2.6, где в качестве конкретного инструмента выбрана СУБД Apache Derby, полностью реализованная на языке Java. Интерес к ней также вызван возможностью ее использования как в *сетевом варианте*, так и — *во встроенном*, что расширяет потенциал целевых приложений.

Общая методика выполнения данной лабораторной работы — аналогична предыдущим: необходимо последовательно выполнять пункты 2.6.1-2.6.3, указанного выше подраздела, осваивая:

- инструментальные средства СУБД Apache Derby;
- SQL-запросы и драйверы баз данных;
- типовой пример выборки данных.

Все примеры, приведенные в перечисленных выше пунктах, - крайне полезны в плане получения навыков программирования на языке Java и завершают краткий экспресс-курс по его изучению. Эти навыки используются в последующих двух темах посвященных объектным распределенным системам и web-технологиям распределенных систем. В связи с этим, студенту следует обратить внимание на следующие вопросы, изложенные в следующих двух пунктах:

- типовой пример выборки данных;
- функции и типы данных СУБД Apache Derby.

2.4.1 Типовой пример выборки данных

Описание типового примера приведено в учебном пособии [1, пункт 2.6.3]. В нем рассматриваются:

- база данных с именем *exampleDB*, демонстрирующая технологию использования инструментальных средств СУБД Apache Derby;
- таблица *notepad*, содержащая два поля: *notekey* — уникальный ключ и *text* — поле записей текстовых сообщений.

В таком контексте, как сама база данных, так и созданная в ней таблица, являются примером простейшего хранилища информации, на основе которого демонстрируются инструментальные средства языка Java, сосредоточенные в

программном пакете *java.sql*.

С другой стороны, этот же пример можно рассматривать как элемент распределенного приложения, являющийся частью некоторой информационной системы, взаимодействующей с другими сетевыми приложениями или системами.

Студенту следует внимательно изучить всю технологию работы с этим примером, поскольку он далее рассматривается как основа типового примера, используемого в технологиях последующих двух глав данной дисциплины.

2.4.2 Функции и типы данных СУБД Apache Derby

В указанных выше трех пунктах подраздела 2.6 учебного пособия [1], кратко рассмотрены:

- установка и структура ПО СУБД Apache Derby;
- настройка параметров среды ОС для правильного запуска СУБД;
- первичная процедура тестирования правильности настроек, связанная со стартом и завершением работы СУБД;
- использование административной утилиты *ij*;
- настройки параметров безопасности языка Java;
- установка драйверов баз данных и параметры настройки соединения с базами данных;
- классы и методы для осуществления SQL-запросов к таблицам баз данных.

Пречисленной информации во многом недостаточно для полноценной работы с объектами СУБД, поэтому в источнике [2, подраздел 1.5] приведено краткое описание часто используемых типов данных и функций СУБД Derby.

Дополнительно, в рабочей области пользователя *upk*, от имени которого осуществляется лабораторная работа, представлена официальная документация на СУБД и ряд русскоязычных файлов, поясняющих ряд аспектов ее администрирования. Указанные источники размещены в каталоге *~/Документы/derby-doc* и могут быть использованы для дальнейшего изучения используемых инструментальных средств или как справочная литература, во время выполнения лабораторных работ.

В качестве рекомендации, можно предложить к изучению следующие файлы:

- *derbytools.pdf* — описание работы с утилитой *ij*;
- *derbyadmin.pdf* — руководство по администрированию сервера баз данных;
- *refderby.pdf* — ссылочное руководство по всем аспектам использования языка SQL, применительно к СУБД Derby.

3 Реализация распределенной системы средствами RMI

Шестая лабораторная работа непосредственно посвящена объектным распределенным системам. Она выполняется под официальным названием «*Реализация распределенной системы средствами RMI*».

С настоящего момента считается, что студент освоил теоретические и практические вопросы использования языка Java и способен реализовывать большинство приложений поддерживаемых платформой J2SE.

Тематике объектных распределенных систем полностью посвящена глава 3 учебного пособия [1], где описаны теоретические положения брокерной архитектуры, а также варианты реализации таких систем средствами технологий CORBA и RMI.

Данная лабораторная работа охватывает только теоретические и практические решения, предоставляемые технологией RMI, которая рассматривается как частный вариант реализации объектных распределенных систем, использующих язык программирования Java. Тем не менее, студент должен иметь полное представление о технологии CORBA, поскольку она является стандартом теоретических построений распределенных систем.

В методическом плане, лабораторная работа разделена на две части, последовательно выполняемые студентом:

1. Первая часть — подготовительная. Она посвящена предварительному преобразованию проектируемой системы, предполагающему: явное выделение серверной и клиентской частей распределенного приложения, а также явное описание интерфейса взаимодействия между этими частями.
2. Вторая часть — непосредственная реализация серверной и клиентской частей распределенной системы.

Методические рекомендации по выполнению каждой из выделенных частей лабораторной работы описаны в двух отдельных подразделах данной главы. Предполагается, что студент уже освоил теоретическую часть учебного пособия [1] в пределах подразделов 3.1 и 3.2 главы 3. Этих сведений вполне достаточно для понимания общей схемы выполняемой работы.

В качестве целевого приложения, которое рассматривается как объект разработки, выбран типовой пример выборки данных, изученный студентом на предыдущей лабораторной работе. Таким образом сокращается подготовительная часть текущей работы, поскольку студенту известна содержательная часть приложения и вариант его реализации классическими средствами языка Java, основанными на инструментарии пакета *java.sql*. С другой стороны, парадигма объектных распределенных систем отличается от классических представлений парадигмы «клиент-сервер», поэтому студент должен уделить особое внимание первой части выполняемой работы, описанной в подраздела 3.1 данного руководства.

3.1 Технология КЛИЕНТ-СЕРВЕР

Приступая к выполнению этой части лабораторной работы, студент должен явно осознать различия парадигм *классического* и *распределенного* подходов построения вычислительных систем.

Классический подход, изученный в предыдущей лабораторной работе на примере типовой выборки информации из базы данных, предполагает наличие некоторого хранилища данных и некоторого потенциала доступа к этим данным, предоставляемого инструментарием СУБД. Соответственно, классическое приложение проектируется и реализуется исходя из общей совокупности инструментальных средств, предоставляемых как СУБД, так и средствами языка реализации. При этом, создаваемые объекты, методы или функции приложения содержат инструментарий обоих типов.

Например, приложение типового примера (см. [1, пункт 2.6.3, листинг 2.15]) базируется на объекте класса *Example11*, который, кроме методов доступа к таблице *notepad* базы данных *exampleDB*, содержит методы *getKey()* и *getString()* для чтения данных с клавиатуры компьютера.

Распределенный подход, изучаемый в данной лабораторной работе, требует:

- явное описание интерфейса, который будет представлять удаленный объект;
- реализацию серверной части приложения, которая должна обеспечивать реализацию объявленного интерфейса;
- реализацию клиентской части приложения, которая использует только функционал удаленного объекта, в рамках объявленного интерфейса, и должен самостоятельно реализовать инструментарий своей локальной части.

Таким образом, распределенный подход, как правило, требует явного разделения функционала серверной и клиентской частей приложения, что и демонстрируется примерами, описанными в [1, пункты 3.2.2 и 3.2.3]. Методические указания по реализации этих примеров приведены в следующих двух пунктах данного методического пособия.

Следует заметить, что само разделение приложений на клиентскую и серверную части не требует обязательной идентичности общему интерфейсу, который будет описывать удаленный объект. Главное, чтобы серверная часть приложения была потенциально способной реализовать такой интерфейс. Это связано с тем, что в большом проекте достаточно трудно дать полное и точное описание такого интерфейса, поэтому явное применение технологии «*клиент-сервер*» является предварительным решением процесса проектирования распределенного приложения.

3.1.1 Серверная часть приложения *NotePad*

Проектирование и реализация распределенного приложения должна начинаться с выделения серверной его части. Это обосновано тем, что серверная часть должна будет реализовывать конкретные интерфейсы удаленных объектов.

Указанная парадигма применена и в рассматриваемом примере создания класса *NotePad*, полностью описанном в [1, пункт 3.2.2].

Студенту следует выполнить данный пункт задания лабораторной работы, полностью опираясь на описание указанного выше источника.

Следует обратить внимание, что данная часть работы выполняется не только в пределах создания отдельного класса, но и оформление его в виде библиотеки, которая, в соответствии с технологией языка Java, представляет собой JAR-архив.

3.1.2 Клиентская часть приложения *Example12*

В рамках общей технологии проектирования и реализации объектной распределенной системы, клиентская часть приложения создается после реализации серверной части.

С методической точки зрения, текущая реализация клиентской части приложения должна быть нацелена не столько на создание функционала приложения, сколько на тестирование функционала серверной части. Такой подход вполне обоснован, поскольку полный интерфейс удаленного объекта еще не описан. Более того, не выбрана даже технология реализации распределенного приложения.

Таким образом, создаваемое клиентское приложение следует рассматривать как обобщенный шаблон будущих реализаций распределенных приложений, который будет уточняться по мере выбора технологии реализации всей распределенной системы.

Что касается выполнения этой части лабораторной работы, то ее описание соответствует учебному материалу источника [1, пункт 3.2.3], где:

- само формальное описание клиентской части приложения представлено в виде класса *Example12*;
- в качестве локальных для этого класса используются методы *getKey()* и *getString()*, подробно описанные в предыдущей лабораторной работе;
- в качестве исследуемой модели удаленного объекта используется экземпляр класса *NotePad*, реализованный в предыдущем пункте текущей работы.

Реализация, исследование и описание класса *Example12*, заканчивает первую часть задания всей лабораторной работы.

3.2 Технология RMI

Вторая часть лабораторной работы посвящена реализации демонстрационного приложения на основе технологии RMI, которая в теоретическом плане рассматривается как частный случай технологии CORBA. В таком аспекте изложенного теоретического материала, студент должен в совершенстве изучить учебный материал источника [1, подраздел 3.2] и выполнить описание интерфейса удаленного объекта средствами языка IDL, применительно к примеру, описанному в первой части текущей лабораторной работы.

Выполнив подготовку материала, описанную выше, студент приступает к реализации распределенной системы средствами технологии RMI, методические указания по которой изложены в следующих двух пунктах данного пособия.

3.2.1 Реализация распределенной системы *RmiPad*

Полное описание реализации учебного примера средствами технологии RMI приведено в отдельных пунктах источника [1, подраздел 3.3].

В методическом плане, последовательность выполнения этой части работ осуществляется в виде следующей последовательности шагов:

1. Описание интерфейса удаленного объекта.
2. Изучение брокера регистрации приложений — *rmiregistry*.
3. Создание серверной части приложения, реализованного с помощью класса *RmiPadServer*.
4. Создание клиентской части приложения, реализованной с помощью класса *RmiPadClient*.
5. Отладка и завершение реализации приложения посредством создания двух библиотек в виде JAR-архивов.

Все указанные стадии проектирования и реализации приложения подробно описаны в указанном выше источнике и не требуют дополнительных пояснений.

3.2.2 Использование утилиты *rmic*

В общем случае, современная реализация объектных распределенных систем с помощью технологии RMI предполагает использование протокола *JRMP* и является самодостаточной. В тех случаях, когда технология RMI используется как средство реализации только части приложения, создаваемого с помощью технологии CORBA, необходимо применять утилиту *rmic*, которая формирует дополнительные классы, необходимые для согласования используемых интерфейсов.

Краткое описание возможностей утилиты *rmic* приведено в главе 2 учебно-методического руководства [2].

4 Web-технологии распределенных систем

Последние три лабораторные работы изучаемой дисциплины посвящены тематике web-технологий, которые, в настоящее время, широко применяются в построении распределенных вычислительных систем. Считается что они более современны и перспективны, чем изученные технологии объектных распределенных систем.

Весь используемый в данной группе работ учебный материал связан общей тематикой, которая изложена:

- в учебном пособии [1, глава 4, стр. 152-197];
- в учебно-методическом пособии [2, глава 3, стр. 60-94].

Описание каждой лабораторной работы приведено в отдельном подразделе данной главы. Кроме СУБД Apache Derby, каждая лабораторная работа использует инструментальные средства Apache Tomcat, который может выполнять функции простейшего web-сервера, но главная его задача — служить контейнером сервлетов и поддерживать технологию JSP-страниц.

Перед началом выполнения этой серии лабораторных работ, студент должен изучить теоретический материал, касающийся «Общего описания технологии web» и «Модели «Клиент-сервер»». Этот учебный материал изложен в [1, подразделы 4.1 и 4.2]. Сами лабораторные работы основаны на материале подраздела 4.3 (см. также источник [1]).

4.1 Технология сервлетов на базе сервера Apache Tomcat

Седьмая лабораторная работа по изучаемой дисциплине имеет официальное название «Технология сервлетов на базе сервера Apache Tomcat». Студент должен понимать, что сами web-технологии появились и развивались как некоторый самостоятельный подход реализации информационных технологий. Тем не менее, они всегда претендовали на нечто большее и всегда демонстрировали многообещающие реализации отдельных технологических решений.

Ситуация кардинально изменилась, когда корпорация Sun Microsystems выпустила новую платформу языка Java, обозначаемую как **J2EE**, в которой была представлена технология сервлетов, обеспечивающих эффективное формирование HTML-страниц на стороне сервера. С этого времени web-технологии стали неотъемлемым элементом создания прикладных систем уровня предприятия и прочно закрепились как перспективный вариант реализации распределенных вычислительных систем.

Специально для целей поддержки технологии сервлетов был разработан сервер Apache Tomcat, предоставляющий инструментарий для реализации приложений средствами web-технологий. Изучению этого инструментария и посвящена текущая лабораторная работа, которая в методическом плане разделена на две

части:

- Apache Tomcat — как контейнер сервлетов, содержащая изучение основ технологии сервлетов;
- дополнительные классы технологии сервлетов, расширяющие представления о сервлетах и выполняемые в лабораторной работе в зависимости от выделенного для этого учебного времени.

4.1.1 Apache Tomcat — как контейнер сервлетов

Этой части лабораторной работы посвящены пункты 4.3.1-4.3.2 учебного пособия [1], где последовательно выполняются следующие задания:

- создание и демонстрация работы с простейшей HTML-страницей [1, стр. 162-163];
- описание базовых классов и методов сервлетов [1, стр. 165-167];
- структура ПО сервера Apache Tomcat v8.5 [1, стр. 167-168];
- создание проекта типа «*Dynamic Web Project*» [1, стр. 169-176];
- работа с диспетчером запросов *RequestDispatcher* [1, стр. 176-179].

Итогом выполнения этой части работы должны быть:

- практические навыки и теоретическое понимание возможностей технологии сервлетов;
- знания и навыки по работе с дистрибутивом Apache Tomcat;
- знания и навыки по созданию проектов в среде Eclipse EE, направленные на работу с технологией сервлетов;
- выполненные в среде Eclipse EE и описанные в личном отчете примеры, приведенные в пунктах 4.3.1-4.3.3 учебного пособия [1].

4.1.2 Дополнительные классы технологии сервлетов

Эта часть лабораторных работ выполняется в объеме выделенного для этого учебного времени, что указывается непосредственно преподавателем, ведущим данный вид занятий.

Учебный материал этой части лабораторных работ изложен в учебно-методическом пособии [2] и соответствует пунктам подраздела 3.1 в нумерации этого источника.

В этой части работ осваивается учебный материал, который с теоретической точки зрения является — второстепенным, но очень важным — с практической точки зрения, поскольку связан с кодированием, передачей и отображением информации в процессе создания и исследования изучаемой технологии.

Результаты выполненных по этой части работ также должны быть отражены в личном отчете студента.

4.2 Технология JSP для формирования динамических HTML-страниц

Восьмая лабораторная работа, выполняемая по тематике инструментальных средств web-технологий, имеет официальное название «*Технология JSP для формирования динамических HTML-страниц*». Ее выполнение предполагает, что студент с успехом закончил все задания предыдущей лабораторной работы.

Текущая работа, как и предыдущая, выполняются в инструментальной среде Eclipse EE и в рамках проекта типа «*Dynamic Web Project*». Здесь — важно, чтобы студент хорошо представлял себе файловую структуру таких проектов, поскольку сервер Apache Tomcat стремится защитить программируемые ресурсы приложения от несанкционированного доступа.

Другая особенность выполняемой работы состоит в том, что студент работает с текстом языка HTML, синтаксис и семантика которого не описана в учебном пособии [1]. С учетом этого обстоятельства, все предлагаемые для выполнения задания подготовлены с условиями:

- HTML-текст JSP-страниц имеет минимальное для его использования содержание;
- в тексте JSP- страниц не используются средства популярного в настоящее время языка JavaScript.

4.2.1 Основные возможности JSP-страниц

Содержание этой части лабораторной работы полностью основано на материале учебного пособия [1, пункт 4.3.4, стр. 179-188]. Предполагается, что на его базе студент получит представление о всех пяти группах синтаксических конструкций JSP: *директивах, объявлениях, выражениях, скрипетах и действиях*. Необходимо также освоить работу с уже доступными в JSP-страницах объектами языка Java: *request, response, session* и *out*.

Все описанные в этой части демонстрационные примеры явно указывают на их возможное дальнейшее использование при реализации приложения, рассмотренного в примерах изучения объектных технологий.

4.2.2 Параметры и атрибуты JSP-страниц

Учебные задания этой части лабораторной работы посвящены передаваемым на сервер параметрам и дополнительному «механизму» взаимодействия сервлета и JSP-страниц, представленного передачей атрибутов посредством методов объекта *request*. Здесь, необходимый материал для изучения и выполнения работы изложен в учебно-методическом пособии [2, подраздел 3.2, стр. 71-80].

4.3 Шаблон проектирования MVC

Девятая (завершающая) лабораторная работа имеет официальное название «Шаблон проектирования MVC». В теоретическом плане, этот шаблон является частью общей методологии проектирования сложных систем, когда система разделяется на три взаимодействующих части: *модель*, *представление* и *управление*.

В плане изучаемой дисциплины, методология шаблона MVC проецируется на технологию сервлетов и JSP-страниц, что делает ее важной методикой разработки распределенных систем средствами web-технологий.

В плане выполняемой лабораторной работы, шаблон MVC применен к задаче реализации приложения простейшей информационной системы, которая уже создавалась с помощью средств технологии CORBA и RMI в разделе, посвященном объектным распределенным системам.

Сам процесс выполнения лабораторной работы разделен на две части:

1. Базовая архитектура шаблона MVC, где в пределах класса **Example14** проекта **proj14** подключается модель системы, уже реализованная в виде класса **NotePad**.
2. Реализация проекта **webpad**, где, на базе предыдущей части, создается полноценный функционал работы с классом **NotePad**.

4.3.1 Базовая архитектура шаблона MVC

В этой части лабораторной работы демонстрируется простейший способ подключения функционала класса **NotePad** к классу сервлета **Example14**. Методика реализации такого подключения основана на прямом расширении функционала самого сервлета и описана в учебном пособии [1, пункт 4.3.5, стр. 189-196].

Студенту необходимо проделать все предложенные примеры, описать их в личном отчете и оценить эффективность использованного подхода.

4.3.2 Реализация проекта webpad

Эта часть лабораторной работы посвящена реализации информационной системы **webpad** в виде отдельного проекта с полным функционалом, аналогичным решению средствами использованной ранее технологии **RMI**. В ней использованы все наработки, описанные в предыдущей части, но в методическом плане — шаблон MVC используется в полной мере (см. [2, подраздел 3.3, стр. 81-94]):

- проведен предварительный анализ задачи и функциональная часть, соответствующая модели, реализована в виде полноценного класса **ModelPad**;
- сервлет **WebPad** выполняет только функции контроллера (управления), обеспечивая коммуникации между моделью, представлением и браузером.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Резник, В. Г. Распределенные вычислительные сети: Учебное пособие [Электронный ресурс] / В. Г. Резник. — Томск: ТУСУР, 2019. — 211 с. — Режим доступа: <https://edu.tusur.ru/publications/9072>
2. Резник, В. Г. Распределенные вычислительные системы: Практические занятия по направлению подготовки бакалавриата 09.03.01 [Электронный ресурс] / В. Г. Резник. — Томск: ТУСУР, 2019. — 96 с. — Режим доступа: <https://edu.tusur.ru/publications/9108>
3. Ноутон П., Шилдт Г. JAVA 2. Наиболее полное руководство в подлиннике. СПб.: БХВ-Петербург, 2008. - 1072 с. - ISBN 978-5-94157-012-6
4. Резник, В. Г. Учебный программный комплекс кафедры АСУ на базе ОС ArchLinux: Учебно-методическое пособие для студентов направления 09.03.01, Направление подготовки "Программное обеспечение средств вычислительной техники и автоматизированных систем" [Электронный ресурс] / В. Г. Резник. — Томск: ТУСУР, 2016. — 33 с. — Режим доступа: <https://edu.tusur.ru/publications/6238>
5. Java DB Reference Manual (Документация по СУБД Derby) Дополнительная документация по дисциплине в файле: [refderby.pdf](#)

Учебное издание

Резник Виталий Григорьевич

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Учебно-методическое пособие предназначено для проведения лабораторных работ по дисциплинам «Вычислительные системы и сети» и «Распределенные вычислительные системы» для студентов направления подготовки бакалавра: 09.03.01 «Информатика и вычислительная техника» направленности (профиля) программы - «Программное обеспечение средств вычислительной техники и автоматизированных систем».

Учебно-методическое пособие

Усл. печ. л. . Тираж . Заказ .

Томский государственный университет
систем управления и радиоэлектроники

634050, г. Томск, пр. Ленина, 40