

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 12.04.2023 10:47:35
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d37e511c11eabb175e945d14a48511da56d089

МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования «Юго-Западный
государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
« 13 » 03 2023 г.



ЗАЩИТА СЕТЕЙ С ПРИМЕНЕНИЕМ МЕЖСЕТЕВЫХ ЭКРАНОВ

Методические указания по выполнению практических работ
для студентов укрупненной группы специальностей и направлений
подготовки 10.00.00

Курск 2023

УДК 004.725

Составитель: А.В. Митрофанов

Рецензент

Кандидат технических наук, доцент кафедры «Информационная безопасность» А.Л. Марухленко

Защита сетей с применением межсетевых экранов: методические указания к выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем» / Юго-Зап. гос. ун-т; сост.: А. В. Митрофанов. Курск, 2023. 14 с. Библиогр.: с. 14.

Указываются необходимые теоретические сведения, порядок выполнения практической работы, содержание отчета.

Методические указания по выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем», предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00

Текст печатается в авторской редакции

Подписано в печать _____ . Формат 60×84 1/16.
Усл.печ.л. . Уч.-изд.л. . Тираж 50 экз. Заказ 183 . Бесплатно
Юго–Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ	4
2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	4
3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.....	5
Библиографический список	14

1. ЦЕЛЬ РАБОТЫ

Овладеть навыками работы с сетевой программой ATGuard

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Межсетевой экран (firewall или брандмауэр) является программно-аппаратным средством осуществления сетевой политики безопасности в выделенном сегменте IP-сети.

В сфере компьютерных сетей межсетевой экран представляет собой барьер, защищающий от вторжения злоумышленников во внутреннюю локальную сеть для того, чтобы скопировать, изменить или стереть информацию либо воспользоваться памятью или вычислительной мощностью работающих в этой сети компьютеров. Межсетевой экран призван обеспечить безопасный доступ к внешней сети и ограничить доступ внешних пользователей к внутренней сети.

Название «брандмауэр», может относиться к одному устройству или одной программе. Термин «межсетевой экран» был принят для обозначения совокупности компонентов, которые находятся между вашей сетью и внешним миром и образуют защитный барьер.

Брандмауэр не может защитить от:

- вирусов. Хотя некоторые брандмауэры и способны распознавать вирусы в проходящем через них трафике, существует множество способов спрятать вирусы в программе. Если даже в описании вашего брандмауэра заявлена функция антивирусной проверки, не выключайте проверку вирусов на отдельных компьютерах в сети;

- «тройанских коней». Как и в случае с вирусами, блокировать проникновение в сеть «тройанских коней» (Trojan horses) достаточно сложно. Пользователь нередко поддается искушению загрузить программу из Internet или открыть прикрепленный к сообщению электронной почты файл, проложив тем самым путь в систему вредоносной программе;

– «социальной инженерии». Термин «social engineering» возник недавно и применяется для описания методов получения хакерами информации от доверчивых пользователей. Часто люди готовы сообщить свой пароль любому, кто позвонил по телефону и отрекомендовался представителем службы безопасности, что-нибудь «проверяющим». Межсетевой экран не в состоянии остановить невоздержного на язык сотрудника

– некомпетентности. Плохо подготовленные сотрудники или небрежное руководство приводят к ошибкам в настройках локальной сети и межсетевого экрана. Если сотрудники не понимают, как работает брандмауэр и как правильно его настраивать, не исключено, что это будет способствовать возникновению проблем;

– атаки изнутри. Межсетевой экран не может предотвратить злонамеренные действия внутри вашей сети. Это одна из причин, по которой безопасность компьютеров в сети остается важной проблемой и после установки брандмауэра.

3. ПОРЯДОК ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ

1) Установка и описание программы AtGuard

После инсталляции программы и перезагрузки компьютера Вы обнаружите в системном трее (system tray) иконку запущенного AtGuard'a , а вверху экрана его же панель (dashboard). Это означает, что инсталляция и первый запуск прошли успешно. Двойной щелчок на иконке открывает окно настроек.

Установка флажка Enable web filters включает блокирование, опции секретности и активные установки фильтров, определенные в диалоговом окне Web (HTTP) Filters. Уберите этот флажок, если Вы хотите выключить все web-фильтры.

Enable web filters действует как главный переключатель, который позволяет вам отменять индивидуальные установки фильтра в диалоговом окне Web (HTTP) Filters и отключать всю фильтрацию веб-трафика. Когда Вы отключаете web-фильтры, программное обеспечение прекратит фильтровать любые HTTP данные, входящие или исходящие от рабочей станции. Если Вы устанавливаете или снимаете флажок Enable web filters, то эти изменения начинают действовать сразу.

Опции в этом диалоговом окне позволяют Вам включать или выключать индивидуальные web-фильтры, cookie и Java/ActiveX мониторы. Вы можете также модифицировать список портов, которые AtGuard контролирует для HTTP связи, когда web-фильтры включены. Установите флажки Ad Blocking, Privacy, Active Content и Cookie Assistant в этом диалоговом окне - этого будет достаточно. Флажок Java/ActiveX Assistant можете не устанавливать, иначе AtGuard будет каждый раз задавать ненужные вопросы.

HTTP Port List - сетевые сервисы (типа HTTP или FTP) используют специфические порты на вашем компьютере. Например, HTTP-связь обычно проводится через порт 80. Web-фильтры AtGuard'a контролируют весь HTTP-трафик, посланный и полученный через порты, которые указаны в Port List, применяя блокирование, секретность и другие опции, которые Вы определили. Ваша рабочая станция может соединяться с Интернетом посредством прокси-сервера, при этом весь HTTP-трафик проходит через порт, используемый этим прокси-сервером. Или Вы можете использовать приложение, которое инициирует HTTP-связь через нестандартный порт. Если HTTP-трафик идет через нестандартный порт, вы должны добавить номер этого порта в Port List. Изменения в настройках фильтров вступают в действие сразу после нажатия кнопки "Применить".

Add Site - нажмите эту кнопку, чтобы открыть диалоговое окно New Site/Domain, которое используется для добавления нового сайта или домена к иерархическому списку сайтов в левом окне. Напечатайте имя web-сайта или

имя домена и нажмите ОК. После добавления сайта Вы можете выбрать его в списке. Используйте установки Ad Blocking, Privacy, Active Content, чтобы определить правила и набор блокировок, которые AtGuard использует только когда Вы посещаете конкретный web-сайт.

Ad Blocking - эти установки позволяют поддерживать блокирующий список по умолчанию и специфические для конкретных сайтов блокирующие списки, которые используются, чтобы указать, чего не нужно отображать на веб-страницах. Когда блокирующий фильтр включен, все HTML-страницы просматриваются на предмет наличия HTML-строк, специфичных для конкретного сайта, указанного в списке для блокирования, плюс значения по умолчанию, определенные для всех сайтов. Любой HTML-код, который содержит разрешенную к блокированию строку, будет удален из веб-страницы AtGuard'ом прежде, чем эта страница будет интерпретирована и показана браузером.

Privacy - установки секретности позволяют определять правила, управляющие тем, как Ваш браузер обрабатывает запросы о различных типах информации, сделанных сайтами, которые Вы посещаете.

Cookies - это информация, которую web-серверы сохраняют на вашем компьютере для более позднего использования. Web-серверы могут читать cookies, чтобы следить, сколько раз вы их посетили, когда и какую информацию вы просматривали. Они могут даже использовать cookies, чтобы передать эту информацию другим web-серверам, типа серверов рекламы. Положительная сторона cookies в том, что они могут использоваться, чтобы сохранить вашу собственную конфигурацию web-сайта, запоминать, что вы поместили в вашу "покупательскую корзину" в интерактивном магазине или сохранять имя пользователя и пароль для сайтов подписки. Чтобы обеспечить максимальную секретность, разрешите использование cookies только проверенным сайтам, которым вы доверяете.

Referer - позволяет вам определить, узнают ли третьи сайты о том, из какого места поступил запрос данных с этих серверов.

Refer field - эти поля используются, чтобы обеспечить "третьи" сайты информацией относительно сайта, с которого поступил запрос данных из их сервера. Refer поля позволяют веб-серверам знать, где вы только что были. Вполне возможно, что вы не захотите, чтобы эта информация становилась известной. Иногда это опасно. Например, некоторые онлайн-почтовые службы подставляют пароль просто в сетевой путь, который отображается в браузере. Если вам пришло письмо, содержащее ссылку на какой-нибудь сайт, и вы последовали по ссылке прямо из web-mail, то в статистике сайта на, который вы пришли, будет зафиксирован адрес страницы, содержавшей ссылку на него - refferer. А этот самый refferer может содержать ваш логин и пароль к вашему почтовому ящику. Некоторые сайты не позволяют заходить на них с включенным Block refer fields.

Browser (User-agent) - позволяет определить, обеспечиваются ли сайты информацией, какой браузер вы используете.

E-mail (From) - позволяет определить, получают ли сайты адрес электронной почты, который использует ваш браузер, чтобы идентифицировать вас как отправителя почты.

Active Content - эти установки позволяют Вам предотвращать выполнение следующих типов программ: JavaScript, Java applets, ActiveX controls. Кроме того, можно установить, чтобы анимационные изображения проигрывались только один раз. Когда блокирование активного содержимого включено, все HTML страницы просматриваются, и любой HTML-код, который активизирует нежелательное содержание, будет удален из страницы AtGuard'ом прежде, чем страница интерпретируется и отобразится веб-браузером.

Установки файрвола определяют, должен ли AtGuard запретить или разрешить приложениям на вашем компьютере посылать или получать информацию по TCP/IP. Для этого имеется список правил, которые описывают, какие типы сетевой активности разрешаются, и через какие

сервисы приложения могут связываться. Вы можете добавлять, изменять, или удалять правила.

Включите опции Enable firewall, Enable RuleAssistant (interactive learning mode)

Для временного отключения какого-либо правила уберите флажок напротив соответствующей строки в списке правил.

Если правило "Блокировать" (Block) или "Разрешить" (Permit) указано, все оставшиеся правила игнорируются. Другими словами если вы, например, закрыли порт номер N, а ниже прописано правило, разрешающее использование этого порта, то оно будет проигнорирована и соединение по этому порту будет закрыто.

Если правило "Игнорировать" (Ignore) указано, тип связи, которая была предпринята, регистрируется в лог-файле firewall'a и затем обработка продолжается, пока не произойдет какого-либо другого соответствия. Если не найдется никакого правила, связь или блокируется (по умолчанию) или вызывается RuleAssistant. Чтобы перемещать правило по списку, выделите соответствующую строку и затем используйте кнопки "стрелка вверх" или "стрелка вниз" для помещения правила в соответствующую позицию.

Любое TCP/IP соединение, для которого нет firewall правила, блокируется по умолчанию. Если Вы хотите выборочно блокировать или разрешать соединение, для которого нет правила, установите флажок Enable RuleAssistant (интерактивный режим изучения).

Если флажок RuleAssistant включен, вам будет автоматически задан вопрос запретить (Block) или разрешить (Permit) соединение всякий раз, когда приложение на вашей рабочей станции или какое-то приложение извне делает попытку установить связь, для которой не описано никаких правил в firewall. В результате вашего решения AtGuard разрешает или блокирует сетевую связь и может создавать правило firewall, которое применяется в дальнейшем для данного типа сетевого соединения.

Direction. Inbound связь включает пакеты, посланные вашему компьютеру. Outbound связь включает пакеты, посланные вашим компьютером. Either - связь в любом направлении.

Protocol. Определяет, к какому протоколу связи применяется правило: TCP, UDP, или TCP и UDP, ICMP...

TCP - стандартный протокол Интернета транспортного уровня, обеспечивает надежную полнодуплексную связь. Программное обеспечение, реализующее протокол TCP, обычно постоянно находится в операционной системе и использует IP протокол, чтобы передать информацию. Примеры TCP приложений и сервисов - FTP, web-браузер, email и IRC.

UDP - транспортный уровень в TCP/IP сетях. UDP - низкоуровневый протокол, который использует IP, чтобы доставить пакеты. Примеры сервисов и приложений, которые используют UDP - DNS, NetBIOS.

ICMP - протокол межсетевых управляющих сообщений.

Application. Эта опция позволяет определять, применяется ли правило к конкретному приложению или к любому приложению, которое делает попытку сетевой связи, определенной правилом.

Service. Позволяет определять, применяется ли правило к локальным или удаленным сервисам и применяется ли это к одиночному определенному сервису или к любому сервису, который делает попытку сетевой связи, определенной правилом.

Time Active. Используйте эти установки, чтобы определить время когда, правило будет действовать.

Logging. Определяет, что событие регистрируется в лог-файле, когда устанавливается описанное правилом соединение.

Show taskbar icon - при запущенном AtGuard показывать его иконку в панели задач.

Show dashboard window - при запущенном AtGuard показывать dashboard.

Enable password protection - если выбрано, то как только вы попытаетесь открыть диалоговое окно AtGuard Settings, окно Dashboard Properties, Event Log, или окно статистики, вы будете должны ввести пароль.

StartUp Options / Run at network startup. Когда эта опция выбрана, AtGuard запускается автоматически, если вы открываете сетевое соединение, и останавливается также автоматически, когда вы закрываете ваше сетевое соединение.

Сразу после инсталляции, зайдите в настройки Firewall. Снимите флажки Default Inbound ICMP, Default Inbound DNS, Default Inbound Bootp, Default Inbound NetBIOS

Для дальнейшей настройки Inbound DNS необходимо узнать DNS адрес вашего провайдера. Затем Settings -> Firewall -> Add. В поле "Name" впишите DNS, поля "Action" и "Directon" изменять не нужно. Поле "Protocol" установите "TCP or UDP 3". Здесь же нажмите закладку "Service" и установите "Remote service" в "Single service", в появившееся поле впишите 53 (номер порта домена). Теперь выберите закладку "Address" и поставьте "Remote address" = "Host address" и в появившееся поле впишите адрес DNS вашего провайдера, нажмите ОК. Теперь повторите эту же операцию только измените "Directon" на "Inbound". Повторите то же самое и для остальных адресов DNS, если они есть. Все, настройка DNS закончена.

Как можно защититься от рекламных баннеров. Правой кнопкой мыши нажимаем по баннеру. В всплывающем меню выбираем "Копировать ярлык". Вызываем AtGuard Settings -> Web. Выбираем в списке (Defaults), Ad Blocking -> Add. В появившемся окошке нажимаем правой кнопкой мыши, выбираем «вставить». Например, для linkexchange будет такая строка <http://www.linkexchange.ru/users/091164/goto.map> Ее нужно отредактировать, чтобы получилось [linkexchange.ru/users/](http://www.linkexchange.ru/users/) ибо удаленная часть может изменяться от сайта к сайту. Всё. Еще пример. <http://www.reklama.ru/cgi-bin/href/myclub?3353573> После правки: [reklama.ru/cgi-bin/](http://www.reklama.ru/cgi-bin/) .

Есть еще один более простой способ. Если у вас запущен Dashboard, то в правом углу будет Trashcan ("Мусорная корзина") AtGuard'a. Чтобы переместить рекламу в мусорку при использовании MSIE 4.0, выберите рисунок и мышкой перетащите его в Trashcan. При использовании Netscape или MSIE 3.0, щелкните правой кнопкой мыши на баннере. Чтобы заблокировать все подобные ссылки, выберите пункт Copy link location (если картинка грузится с того же сервера, что и страница). Если баннер грузится с сервера рекламодателя (например, это могут быть баннеры сетей reklama.ru, linkexchange), то выберите пункт Copy image location. Затем щелкните правой кнопкой мыши на иконке Trashcan и выберите пункт Paste (Вставка) из всплывающем меню.

Как говорилось выше, нашу задачу сильно облегчает то, что вся реклама объединяется или уже объединена в рекламные (баннерные) сети. Поэтому закрыв для себя AtGuard'ом один рекламный сайт, вы избавитесь от сотен и сотен рекламных баннеров. Это сохранит вам деньги, нервы и высокую скорость соединения.

Обязательно установите все флажки в окне AtGuard Settings -> Web -> Active Content.

В окне AtGuard Settings -> Firewall включите Enable Rule-Assistant для интерактивного обучения вашего стража. Если в процессе вам встретится непонятное на первый взгляд сообщение о каком-либо соединении, запретите его, потом всегда можно посмотреть в лог-файлах.

2) *Защита от атак WinNuke*

Чтобы защититься от атаки WinNuke, нужно поставить соответствующий фильтр. Атака WinNuke заключается в послышке ООВ-данных на 139 порт. Таким образом, достаточно будет заблокировать TCP-соединения с 139 портом. Однако 139 порт используется для NetBIOS и потому при работе в локальной сети его перекрывать не следует. Но если вы заходите в Сеть с домашнего компьютера, то блокируйте смело.

В настройке Firewall добавляем новое правило – “Add”. Назовем “WinNuke”, действие – “Block”, направление только входящие – “Inbound”, протокол “TCP”. Далее на закладках: Any Application. Service: remote - "Any", local – single service 139 порт. Остальные настройки можно оставить по умолчанию. Включите протоколирование, чтобы можно было видеть, что вы подверглись атаке. По аналогии можно настроить и другие фильтры.

3) *Задание на практическую работу*

- Изучить функции программы, пользуясь описанием программы.
- Установить 2 виртуальные машины. Настроить локальную сеть между двумя виртуальными машинами, если требуется.
- Установить обе виртуальные машины AtGuard.
- Осуществить обмен пакетами запрещенного типа при включенном и при выключенном AtGuard с другим компьютером сети.
- Создать правило запрещающее получение доступа к компьютеру с удаленного компьютера (с конкретного IP-адреса или с определенного имени компьютера), попытаться обратиться с запрещенного компьютера и отследить реакцию AtGuard.
- Составить отчет о проделанной работе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Шувалов В.П., Величко В.В., Субботин Е.А., Ярославцев А.Ф. Телекоммуникационные системы и сети. Том 3. Мультисервисные сети (2005)
- 2) Петраков А.В. Основы практической защиты информации. 2-е изд. Учебн. пособие. – М.: Радио и связь. 2000. – 368 с.
- 3) Цифровые и аналоговые системы передачи: Учебник для вузов/ В.И.Иванов, В.Н.Гордиенко, Г.Н.Попов и др.; Под ред. В.И.Иванова. – 2-е изд. – М.: Горячая линия – Телеком, 2003. – 232 с.
- 4) Гольдштейн Б.С., Соколов Н.А., Яновский Г.Г. Сети связи: Учебник для ВУЗов. - СПб.: БХВ-Петербург, 2010. - 400 с.
- 5) Башарин Г.П. Лекции по математической теории телетрафика: Учеб. пособие. Изд. 3-е, испр. и доп. - М.: РУДН, 2009. - 342 с.
- 6) Буч Г. Объектно-ориентированный анализ и проектирование. – М.: Вильямс, 2008.
- 7) Леоненков А.В. Самоучитель языка UML. – СПб.: БХВ-Петербург, 2004.
- 8) Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов. – М.: ДМК Пресс, 2002.
- 9) А.В. Росляков. Виртуальные частные сети. Основы построения и применения. - М.: Эко-Трендз, 2006. - 242 с.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования «Юго-Западный
государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

« 23 » 03

2023 г.



КРИТЕРИИ ОЦЕНКИ И ВЫБОРА CASE- СРЕДСТВ

Методические указания по выполнению практических работ
для студентов укрупненной группы специальностей и направлений
подготовки 10.00.00

Курск 2023

УДК 004.725

Составитель: А.В. Митрофанов

Рецензент

Кандидат технических наук, доцент кафедры «Информационная безопасность» А.Л. Марухленко

Критерии оценки и выбора case-средств: методические указания к выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем» / Юго-Зап. гос. ун-т; сост.: А. В. Митрофанов. Курск, 2023. 10 с. Библиогр.: с. 10.

Указывается порядок выполнения практической работы, правила оформления, содержание отчета.

Методические указания по выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем», предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00

Текст печатается в авторской редакции

Подписано в печать _____ . Формат 60×84 1/16.
Усл.печ.л. . Уч.-изд.л. . Тираж 50 экз. Заказ 132. Бесплатно
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ	4
2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	4
3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.....	8
4. СОДЕРЖАНИЕ ОТЧЕТА	9
Библиографический список	10

1. ЦЕЛЬ РАБОТЫ

Описать и проанализировать информационную систему.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Критерии формируют базис для процессов оценки и выбора и могут принимать различные формы, включая:

- числовые меры в широком диапазоне значений, например, объем требуемой памяти;
- числовые меры в ограниченном диапазоне значений, например, простота освоения, выраженная в баллах от 1 до 5;
- двоичные меры (истина/ложь, да/нет), например, способность генерации документации в формате Postscript;
- меры, которые могут принимать одно или более из конечных множеств значений, например, платформы, для которых поддерживается CASE-средство.

Типичный процесс оценки и/или выбора может использовать набор критериев различных типов.

Структура набора критериев приведена на рисунке 1. Каждый критерий должен быть выбран и адаптирован экспертом с учетом особенностей конкретного процесса. В большинстве случаев только некоторые из множества описанных ниже критериев оказываются приемлемыми для использования, при этом также добавляются дополнительные критерии. Выбор и уточнение набора используемых критериев является критическим шагом в процессе оценки и/или выбора.

Функциональные характеристики

Критерии первого класса предназначены для определения функциональных характеристик CASE-средства. Они в свою очередь подразделяются на ряд групп и подгрупп.

1. Среда функционирования:
 - а. Проектная среда:

- *поддержка процессов жизненного цикла.* Определяет набор процессов ЖЦ, которые поддерживает CASE-средство. Примерами таких процессов являются анализ требований, проектирование, реализация, тестирование и оценка, сопровождение, обеспечение качества, управление конфигурацией и управление проектом, причем они зависят от принятой пользователем модели ЖЦ.

- *область применения.* Примерами являются системы обработки транзакций, системы реального времени, информационные системы и т.д.

- *размер поддерживаемых приложений.* Определяет ограничения на такие величины, как количество строк кода, уровней вложенности, размер базы данных, количество элементов данных, количество объектов конфигурационного управления.

в. ПО/технические средства:

- *требуемые технические средства.* Оборудование, необходимое для функционирования CASE-средства, включая тип процессора, объем оперативной и дисковой памяти.

- *поддерживаемые технические средства.* Элементы оборудования, которые могут использоваться CASE-средством, например, устройства ввода/вывода.

- *требуемое ПО.* ПО, необходимое для функционирования CASE-средства, включая операционные системы и графические оболочки.

- *поддерживаемое ПО.* Программные продукты, которые могут использоваться CASE-средством.

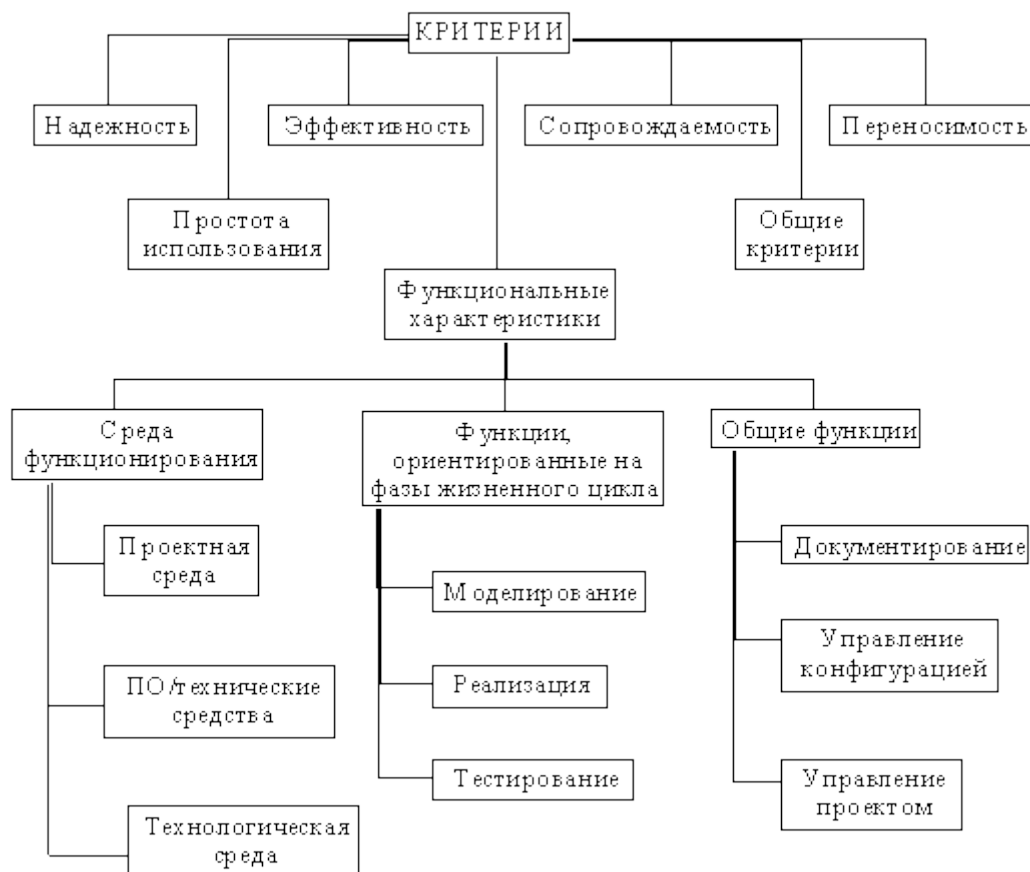


Рис. 1 - Структура набора критериев

Разработка требований — это процесс, включающий мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований. Для новых программных систем процесс разработки требований должен начинаться с анализа осуществимости. Началом такого анализа является общее описание системы и ее назначения, а результатом анализа — отчет, в котором должна быть четкая рекомендация, продолжать или нет процесс разработки требований проектируемой системы. Другими словами, анализ осуществимости должен осветить следующие вопросы.

1. Отвечает ли система общим и бизнес-целям организации-заказчика и организации-разработчика?
2. Можно ли реализовать систему, используя существующие на данный момент технологии и не выходя за пределы заданной стоимости?

3. Можно ли объединить систему с другими системами, которые уже эксплуатируются?

Критическим является вопрос, будет ли система соответствовать целям организации. Если система не соответствует этим целям, она не представляет никакой ценности для организации. В то же время многие организации разрабатывают системы, не соответствующие их целям, либо не совсем ясно понимая эти цели, либо под влиянием политических или общественных факторов.

Выполнение анализа осуществимости включает сбор и анализ информации о будущей системе и написание соответствующего отчета. Сначала следует определить, какая именно информация необходима, чтобы ответить на поставленные выше вопросы. Например, эту информацию можно получить, ответив на следующее:

1. Что произойдет с организацией, если система не будет введена в эксплуатацию?
2. Какие текущие проблемы существуют в организации и как новая система поможет их решить?
3. Каким образом система будет способствовать целям бизнеса?
4. Требуется ли разработка системы технологии, которая до этого не использовалась в организации?

Далее необходимо определить источники информации. Это могут быть менеджеры отделов, где система будет использоваться, разработчики программного обеспечения, знакомые с типом будущей системы, технологи, конечные пользователи и т.д.

После обработки собранной информации готовится отчет по анализу осуществимости создания системы. В нем должны быть даны рекомендации относительно продолжения разработки системы. Могут быть предложены изменения бюджета и графика работ по созданию системы или предъявлены более высокие требования к системе.

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить предлагаемый теоретический материал.
2. Составить подробное описание информационной системы.
3. На основании описания системы провести анализ осуществимости. В ходе анализа ответить на вопросы:

- Что произойдет с организацией, если система не будет введена в эксплуатацию?
- Какие текущие проблемы существуют в организации и как новая система поможет их решить?
- Каким образом система будет способствовать целям бизнеса?
- Требуется ли разработка системы технологии, которая до этого не использовалась в организации?

Результатом анализа должно явиться заключение о возможности реализации проекта.

4. Распределить роли в группе (руководитель проекта-разработчик, системный аналитик-разработчик, тестер-разработчик).
5. Заполнить разделы плана:
 - Введение
 - Организация выполнения проекта
 - Анализ рисков

Разделы должны содержать рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению.

6. Составить отчет о проделанной работе.

4. СОДЕРЖАНИЕ ОТЧЕТА

В отчете следует указать:

1. Цель работы
2. Введение. Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.), которые важны для управления проектом
3. Описание информационной системы (ПО) - наличие заключения о возможности реализации проекта, содержащего рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению
4. Анализ осуществимости (согласно требованиям к результатам выполнения лабораторного практикума п.2), указать возможные проблемы и пути их решения.
5. Роли участников группы разработки ПО.
6. Программно-аппаратные средства, используемые при выполнении работы.
7. Заключение (выводы)

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Шувалов В.П., Величко В.В., Субботин Е.А., Ярославцев А.Ф. Телекоммуникационные системы и сети. Том 3. Мультисервисные сети (2005)
- 2) Петраков А.В. Основы практической защиты информации. 2-е изд. Учебн. пособие. – М.: Радио и связь. 2000. – 368 с.
- 3) Цифровые и аналоговые системы передачи: Учебник для вузов/ В.И.Иванов, В.Н.Гордиенко, Г.Н.Попов и др.; Под ред. В.И.Иванова. – 2-е изд. – М.: Горячая линия – Телеком, 2003. – 232 с.
- 4) Гольдштейн Б.С., Соколов Н.А., Яновский Г.Г. Сети связи: Учебник для ВУЗов. - СПб.: БХВ-Петербург, 2010. - 400 с.
- 5) Башарин Г.П. Лекции по математической теории телетрафика: Учеб. пособие. Изд. 3-е, испр. и доп. - М.: РУДН, 2009. - 342 с.
- 6) Буч Г. Объектно-ориентированный анализ и проектирование. – М.: Вильямс, 2008.
- 7) Леоненков А.В. Самоучитель языка UML. – СПб.: БХВ-Петербург, 2004.
- 8) Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов. – М.: ДМК Пресс, 2002.
- 9) А.В. Росляков. Виртуальные частные сети. Основы построения и применения. - М.: Эко-Трендз, 2006. - 242 с.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования «Юго-Западный
государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
« 23 » 03 2023 г.



ОТПРАВКА И ПРИЕМ СООБЩЕНИЙ С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛОВ UDP И TCP/IP

Методические указания по выполнению практических работ
для студентов укрупненной группы специальностей и направлений
подготовки 10.00.00

Курс 2023

УДК 004.725

Составитель: А.В. Митрофанов

Рецензент

Кандидат технических наук, доцент кафедры
«Информационная безопасность» А.Л. Марухленко

Отправка и прием сообщений с использованием протоколов udp и tcp/ip: методические указания к выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем» / Юго-Зап. гос. ун-т; сост.: А. В. Митрофанов. Курск, 2023. 14 с. Библиогр.: с. 14.

Содержат сведения по вопросам установки и настройки среды разработки Eclipse. Указывается порядок выполнения практической работы, правила оформления, содержание отчета.

Методические указания по выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем», предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00

Текст печатается в авторской редакции

Подписано в печать _____ . Формат 60×84 1/16.
Усл.печ.л. . Уч.-изд.л. . Тираж 50 экз. Заказ 170. Бесплатно
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

1.		
ЦЕЛЬ РАБОТЫ		4
2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.....		4
2.1 Подготовительный этап		4
2.2 Создание нового проекта		4
2.3 Создание класса Server.....		4
2.4 Создание класса Client.....		6
2.5 Запуск и тестирование.....		7
3. СОЗДАНИЕ ПРИЛОЖЕНИЯ TSP/IP		8
3.1 Постановка задачи		8
3.2 Подготовительный этап		9
3.3 Создание класса DateMessage.....		9
3.4 Создание класса ServerTCP.....		9
3.5 Создание класса ClientTCP		11
3.6 Запуск и тестирование.....		11
4 Варианты заданий		12
Библиографический список		14

1. ЦЕЛЬ РАБОТЫ

Разработать клиент/серверное приложение, в котором сервер может распространять сообщения всем клиентам, зарегистрированным в группе 233.0.0.1, порт 1502. Пользователь сервера должен иметь возможность ввода и отправки текстовых сообщений, а пользователь-клиент просматривает полученные сообщения.

2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Для решения поставленной задачи необходимо выполнить следующие шаги:

1. Создать новый проект.
2. Реализовать класс сервера для ввода и отправки сообщений.
3. Реализовать класс клиента для получения и просмотра сообщений.
4. Протестировать приложение – запустить сервер и клиент, и отправить сообщение.

2.1 Подготовительный этап

Для реализации проекта необходимо установить и настроить среду разработки Eclipse (см. п. «Установка и настройка программного обеспечения»).

2.2 Создание нового проекта

- 1) Выберите пункт меню File/New/Project, в окне выбора типа проекта укажите other/Java Project и нажмите Next.
- 2) Укажите имя проекта Lab2 и нажмите Finish.

2.3 Создание класса Server

Класс `Server` предназначен для отсылки сообщений всем клиентам, зарегистрированным в группе 233.0.0.1. Создание класса `Server` включает в себя следующие основные задачи:

1. Создание сокета с помощью класса `DatagramSocket`. Сокет сервера выполняет задачу отправки сообщения.
2. Создание объекта `InetAddress`, представляющего адрес сервера. Адреса для групповой (multicast) передачи сообщений выбираются из диапазона 224.0.0.0 - 239.255.255.255. В нашем приложении будет указан адрес 233.0.0.1.
3. Организация ввода строки сообщения с клавиатуры и создание объекта `packet` класса `DatagramPacket`, который хранит введенные данные и использует метод `send()` объекта класса `DatagramSocket`, для отсылки пакета всем клиентам группы.

1) Для создания класса сервера щелкните правой кнопкой мыши на каталог src в окне Package Explorer и выберите New/Class

2) В появившемся окне в качестве имени пакета (Package) укажите ru.tpu.javaEELabs.lab2, а в качестве имени класса (Name) задайте Server. Нажмите Finish.

Код класса Server приведен ниже:

```
package ru.tpu.javaEELabs.lab2;

import java.io.*;
import java.net.*;

public class Server {

    private BufferedReader in = null;
    private String str = null;
    private byte[] buffer;
    private DatagramPacket packet;
    private InetAddress address;
    private DatagramSocket socket;

    public Server() throws IOException {
        System.out.println("Sending messages");

        // Создается объект DatagramSocket, чтобы
        // принимать запросы клиента
        socket = new DatagramSocket();

        // Вызов метода transmit(), чтобы передавать сообщение всем
        // клиентам, зарегистрированным в группе
        transmit();
    }

    public void transmit() {
        try {
            // создается входной поток, чтобы принимать
            // данные с консоли
            in = new BufferedReader(new InputStreamReader(System.in));
            while (true) {
                System.out.println(
                    "Введите строку для передачи клиентам: ");
                str = in.readLine();
                buffer = str.getBytes();
                address = InetAddress.getByName("233.0.0.1");
                // Посылка пакета датаграмм на порт номер 1502
                packet = new DatagramPacket(
                    buffer,
                    buffer.length,
                    address,
                    1502);

                //Посылка сообщений всем клиентам в группе
                socket.send(packet);
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {

```

```

        try {
            // Закрытие потока и сокета
            in.close();
            socket.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

public static void main(String arg[]) throws Exception {
    // Запуск сервера
    new Server();
}
}

```

2.4 Создание класса Client

Класс `Client` позволяет клиенту присоединиться к группе 233.0.0.1 для получения сообщений от сервера. Создание класса `Client` включает в себя следующие основные задачи:

1. Создание сокета для просмотра групповых сообщений с помощью класса `MulticastSocket`. Сокет клиента выполняет задачу приема сообщения.
2. Создание объекта `InetAddress`, представляющего адрес сервера и присоединение к группе этого сервера с помощью метода сокета `joinGroup`.
3. Организация чтения пакетов датаграмм (`DatagramPacket`) из сокета и отображение полученных данных на экране.

1) Для создания класса клиента щелкните правой кнопкой мыши на пакет `ru.tpu.javaEELabs.lab2` в каталоге `src` окна `Package Explorer` и выберите `New/Class`.

2) В появившемся окне в качестве имени класса (`Name`) задайте `Client`. Нажмите `Finish`.

Код класса `Client` приведен ниже:

```

package ru.tpu.javaEELabs.lab2;

import java.net.*;

public class Client {
    private static InetAddress address;
    private static byte[] buffer;
    private static DatagramPacket packet;
    private static String str;
    private static MulticastSocket socket;

    public static void main(String arg[]) throws Exception {
        System.out.println("Ожидание сообщения от сервера");
        try {

            // Создание объекта MulticastSocket, чтобы получать
            // данные от группы, используя номер порта 1502
            socket = new MulticastSocket(1502);

```



```

Server [Java Application] C:\jdk1.5.0_12\bin\javaw.exe (16.05.2008 20:09:14)
Sending messages
Введите строку для передачи клиентам:
hello!!!

```

4) Просмотрите консоль клиента и убедитесь, что клиент успешно принял сообщение.

```

Client [Java Application] C:\jdk1.5.0_12\bin\javaw.exe (16.05.2008 20:09:11)
Ожидание сообщения от сервера
Получено сообщение: hello!!!

```

5) Остановка приложения осуществляется с помощью кнопки Terminate представления Console.

3. СОЗДАНИЕ ПРИЛОЖЕНИЯ ТСР/Р

3.1 Постановка задачи

Необходимо разработать клиент/серверное приложение, в котором сервер слушает запросы клиентов на порт 1500 и отправляет объект-сообщение содержащий текущую дату/время сервера и строку сообщения. Пользователь-клиент должен иметь возможность просмотра полученного сообщения.

Для решения поставленной задачи необходимо выполнить следующие шаги:

1. Создать класс `DateMessage` с двумя полями: дата и строка – для хранения и передачи сообщения клиенту.
2. Реализовать класс сервера для прослушивания соединений на порту 1500 и отправки сообщений. Задача класса сервера должна выполняться в отдельном потоке.
3. Реализовать класс клиента для получения и просмотра сообщений
4. Протестировать приложение – запустить сервер и клиент, и проверить передачу и получение сообщения.

3.2 Подготовительный этап

Для реализации проекта необходимо установить и настроить среду разработки Eclipse (см. п. «Установка и настройка программного обеспечения»).

3.3 Создание класса DateMessage

- 1) Создайте новый Java-класс DateMessage в пакете ru.tpu.javaEELabs.lab2.
- 2) Скопируйте следующее содержимое класса:

```
package ru.tpu.javaEELabs.lab2;

import java.io.Serializable;
import java.util.Date;

public class DateMessage implements Serializable {

    private Date date;
    private String message;

    public DateMessage(Date date, String message) {
        this.date = date;
        this.message = message;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}}
```

3.4 Создание класса ServerTCP

Создание класса ServerTCP включает в себя следующие основные задачи:

1. Создание серверного сокета с помощью класса ServerSocket.
2. Ожидание запроса от клиента с помощью метода accept() серверного сокета.
3. Формирование объекта-сообщения и отправка его с помощью выходного потока клиентского сокета.

- 1) Создайте новый Java-класс ServerTCP в пакете ru.tpu.javaEELabs.lab2.

Код класса Server приведен ниже:

```

package ru.tpu.javaEELabs.lab2;

import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Calendar;

/**
 * Класс сервера (выполняется в отдельном процессе)
 */
public class ServerTCP extends Thread {

    // Объявляется ссылка
    // на объект - сокет сервера
    ServerSocket serverSocket = null;

    /**
     * Конструктор по умолчанию
     */
    public ServerTCP() {
        try {
            // Создается объект ServerSocket, который получает
            // запросы клиента на порт 1500
            serverSocket = new ServerSocket(1500);
            System.out.println("Starting the server ");
            // Запускаем процесс
            start();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * Запуск процесса
     */
    public void run() {
        try {
            while (true) {
                // Ожидание запросов соединения от клиентов
                Socket clientSocket = serverSocket.accept();

                System.out.println("Connection accepted from " +
                    clientSocket.getInetAddress().getHostAddress());

                // Получение выходного потока,
                // связанного с объектом Socket
                ObjectOutputStream out =
                    new ObjectOutputStream(
                        clientSocket.getOutputStream());

                // Создание объекта для передачи клиентам
                DateMessage dateMessage = new DateMessage(
                    Calendar.getInstance().getTime(),
                    "Текущая дата/время на сервере");
                // Запись объекта в выходной поток
                out.writeObject(dateMessage);
                out.close();
            }
        } catch (Exception e) {

```

```

        e.printStackTrace();
    }
}

public static void main(String args[]) {
    // Запуск сервера
    new ServerTCP();
}
}

```

3.5 Создание класса ClientTCP

Класс ClientTCP позволяет клиенту присоединиться к серверу, используя его IP-адрес (в нашем случае localhost) и получить от него сообщение. Создание класса ClientTCP включает в себя следующие основные задачи:

1. Создание сокета для доступа к серверу localhost на порт 1500.
2. Получение входного потока сокета.
3. Чтение объекта-сообщения из потока и отображение полученных данных на экране.

1) Создайте новый Java-класс ClientTCP в пакете ru.tpu.javaEELabs.lab2.

Код класса Client приведен ниже:

```

package ru.tpu.javaEELabs.lab2;

import java.io.ObjectInputStream;
import java.net.Socket;

public class ClientTCP {

    public static void main(String args[]) {
        try {
            // Создается объект Socket
            // для соединения с сервером
            Socket clientSocket = new Socket("localhost", 1500);

            // Получаем ссылку на поток, связанный с сокетом
            ObjectInputStream in =
                new ObjectInputStream(clientSocket.getInputStream());

            // Извлекаем объект из входного потока
            DateMessage dateMessage =
                (DateMessage) in.readObject();

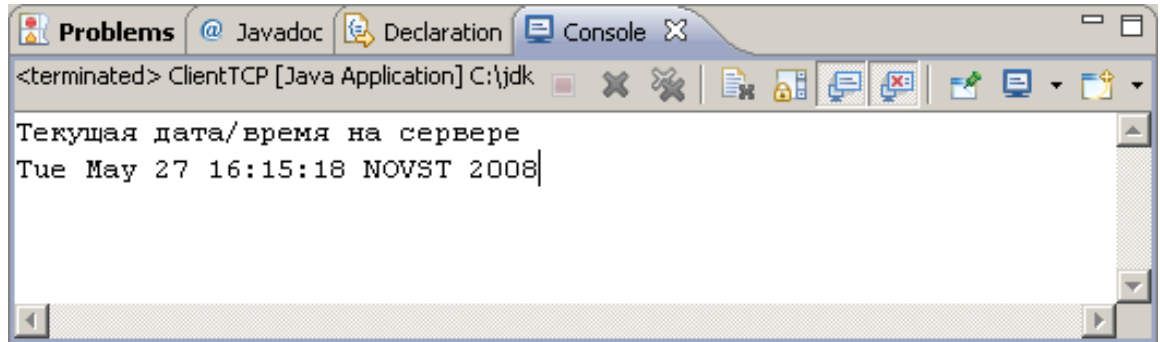
            // Выводим полученные данные в консоль
            System.out.println(dateMessage.getMessage());
            System.out.println(dateMessage.getDate());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

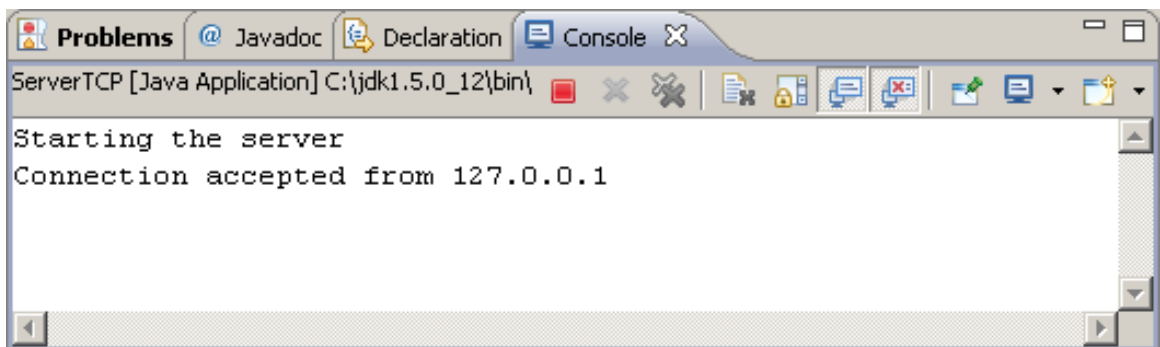
3.6 Запуск и тестирование

Щелкните правой кнопкой мыши на класс ServerTCP в окне Package Explorer и выберите команду Run As/Java Application. В консоли отображается сообщение Starting the server.

Проделайте то же самое с классом ClientTCP. При запуске клиент пытается соединиться с сервером и обрабатывает полученное сообщение. В результате в консоли клиента выводится следующее:



Выберите консоль сервера и просмотрите сообщение о приеме соединения от клиента:



4. ВАРИАНТЫ ЗАДАНИЙ

1. Необходимо разработать клиент/серверное приложение, в котором сервер каждые 10 секунд распространяет некоторое текстовое сообщение, например, о погоде, всем *промежуточным* клиентам, зарегистрированным в группе 233.0.0.1, порт 1502 с помощью UDP. Текст сообщения хранится в текстовом файле на сервере. Промежуточный клиент фильтрует полученные сообщения и в случае изменения содержимого отображает его в консоли. Для *конечного* клиента промежуточный клиент выступает сервером. Конечный клиент присоединяется к промежуточному и получает тексты последних пяти отфильтрованных сообщений (с помощью протокола TCP/IP). Необходимо снабдить приложение конечного клиента графическим интерфейсом.

2. Разработать приложение для широковещательного общения пользователей (чат). Клиент отправляет сообщение серверу (с помощью

протокола TCP/IP). Сервер накапливает порции сообщений и каждые 10 секунд распространяет очередную порцию сообщений всем клиентам, зарегистрированным в группе 233.0.0.1, порт 1502 с помощью UDP. Если за указанный период не поступило ни одного нового сообщения, то рассылка не производится. Клиент принимает сообщения и отображает их на экране. Клиентское приложение должно иметь удобный графический интерфейс для ввода новых и просмотра полученных сообщений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Шувалов В.П., Величко В.В., Субботин Е.А., Ярославцев А.Ф. Телекоммуникационные системы и сети. Том 3. Мультисервисные сети (2005)
- 2) Петраков А.В. Основы практической защиты информации. 2-е изд. Учебн. пособие. – М.: Радио и связь. 2000. – 368 с.
- 3) Цифровые и аналоговые системы передачи: Учебник для вузов/ В.И.Иванов, В.Н.Гордиенко, Г.Н.Попов и др.; Под ред. В.И.Иванова. – 2-е изд. – М.: Горячая линия – Телеком, 2003. – 232 с.
- 4) Гольдштейн Б.С., Соколов Н.А., Яновский Г.Г. Сети связи: Учебник для ВУЗов. - СПб.: БХВ-Петербург, 2010. - 400 с.
- 5) Башарин Г.П. Лекции по математической теории телетрафика: Учеб. пособие. Изд. 3-е, испр. и доп. - М.: РУДН, 2009. - 342 с.
- 6) Буч Г. Объектно-ориентированный анализ и проектирование. – М.: Вильямс, 2008.
- 7) Леоненков А.В. Самоучитель языка UML. – СПб.: БХВ-Петербург, 2004.
- 8) Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов. – М.: ДМК Пресс, 2002.
- 9) А.В. Росляков. Виртуальные частные сети. Основы построения и применения. - М.: Эко-Трендз, 2006. - 242 с.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
«23» 03 2023 г.



СОЗДАНИЕ ПРИЛОЖЕНИЯ ДЛЯ ДОСТУПА К БАЗЕ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛО- ГИИ JDBC

Методические указания по выполнению практических работ
для студентов укрупненной группы специальностей и направлений
подготовки 10.00.00

Курск 2023

УДК 004.725

Составитель: А.В. Митрофанов

Рецензент

Кандидат технических наук, доцент кафедры «Информационная безопасность» А.Л. Марухленко

Создание приложения для доступа к базе данных с использованием технологии jdbc: методические указания к выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем» / Юго-Зап. гос. ун-т; сост.: А. В. Митрофанов. Курск, 2023. 19 с. Библиогр.: с. 19.

Указывается порядок выполнения практической работы, правила оформления, содержание отчета.

Методические указания по выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем», предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00

Текст печатается в авторской редакции

Подписано в печать _____ . Формат 60×84 1/16.
Усл.печ.л. . Уч.-изд.л. . Тираж 50 экз. Заказ 181 . Бесплатно
Юго–Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ	4
2. ВЫПОЛНЕНИЕ РАБОТЫ.....	4
2.1 Подготовительный этап	4
2.2 Создание нового проекта	4
2.3 Создание таблицы products	4
2.4 Создание класса Product.....	8
2.5 Создание класса обработки данных Product DAO.....	9
2.6 Разработка комплексного приложения.....	12
2.7 Запуск и тестирование приложения.....	16
3 Варианты заданий	17
Библиографический список	19

1. ЦЕЛЬ РАБОТЫ

Разработать приложение для автоматизации учета товаров на складе. Приложение должно иметь диалоговый пользовательский интерфейс и позволять пользователю просматривать, добавлять, изменять и удалять информацию о товарах из таблицы products. Таблица products должна содержать следующие поля: идентификатор товара (id), описание (description), цена (rate), количество (quantity).

2. ВЫПОЛНЕНИЕ РАБОТЫ

Для решения поставленной задачи необходимо выполнить следующие шаги:

1. Создать новый проект.
2. Создать в БД таблицу products для хранения данных о товарах.
3. Реализовать Java-класс Product для представления и обработки данных о товаре.
4. Реализовать Java-класс обработки данных ProductDAO, в котором обеспечить получение соединения БД и реализовать методы выборки, создания, обновления и удаления товара.
5. Разработать графический интерфейс приложения (Swing) и обеспечить вызов методов обработки данных.
6. Запустить и проверить работу приложения.

2.1 Подготовительный этап

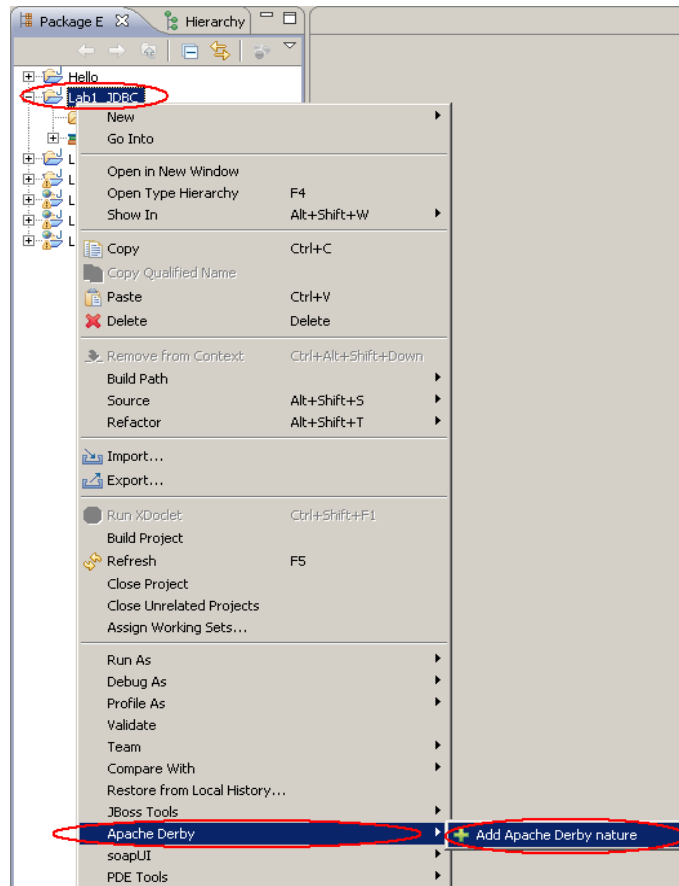
Для реализации проекта необходимо установить и настроить среду разработки Eclipse, Apache Derby и Derby Plugins (см. п. «Установка и настройка программного обеспечения»).

2.2 Создание нового проекта

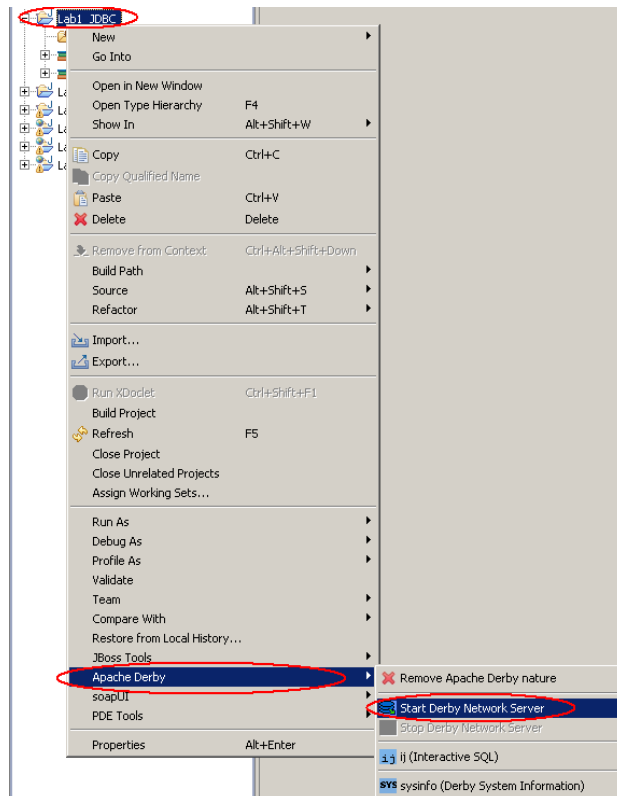
- 1) Выберите пункт меню File/New/Project, в окне выбора типа проекта укажите other/Java Project и нажмите Next.
- 2) Укажите имя проекта Lab1_JDBC, проверьте параметры подраздела JRE – должна быть указана JRE 1.6. (или 1.5, если используется предыдущая версия, но лучше 1.6) Нажмите Finish.

2.3 Создание таблицы products

- 1) Подключите к проекту окружение БД Derby и запустите сервер БД. Для этого, щелкните правой кнопкой мыши на созданный проект в представлении Package Explorer и выберите пункт меню Apache Derby/Add Apache Derby Nature.



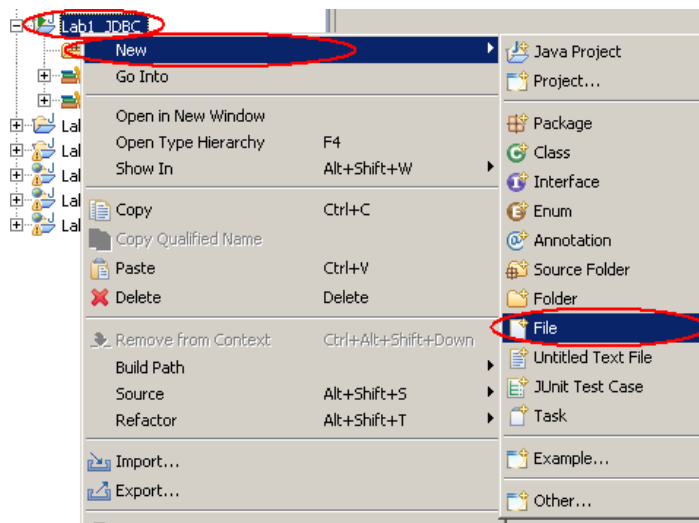
Затем повторно щелкните правой кнопкой мыши на проект и выберите Apache Derby/Start Derby Network Server



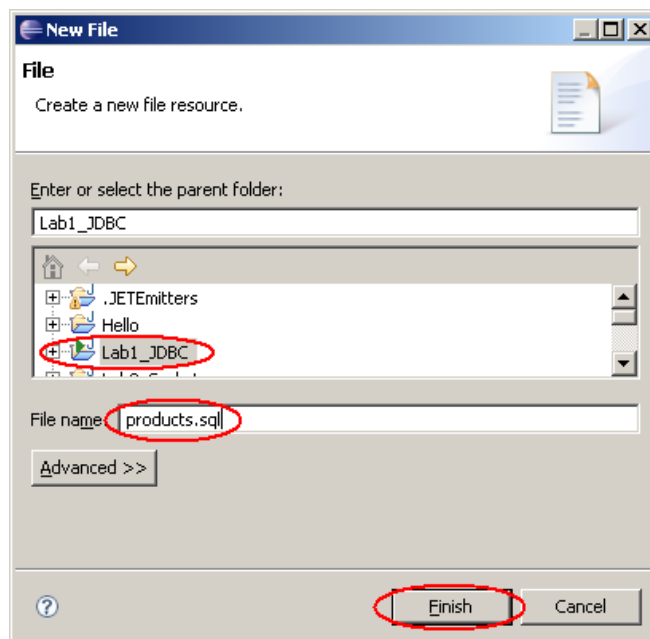
В случае успешного запуска сервера БД в консоли (представление Console) появляется следующее сообщение:

```
DRDA_SecurityInstalled.I
Сетевой сервер Apache Derby Network Server - 10.3.2.1 - (599110) запущен
и готов принимать соединения на порту 1527
```

2) Для хранения SQL-скриптов создадим новый файл products.sql. В окне Package Explorer щелкните правой кнопкой мыши на значок проекта и выберите New/File.



3) В появившемся окне укажите имя файла products.sql и нажмите Finish.



4) SQL-скрипт должен содержать команду создания таблицы products. Скопируйте в файл следующие команды:

```

-- подключение
connect 'jdbc:derby://localhost:1527/myDB;
       create=true;user=me;password=mine';

-- раскомментируйте следующую строку, если требуется пересоздать таблицу
-- drop table products;

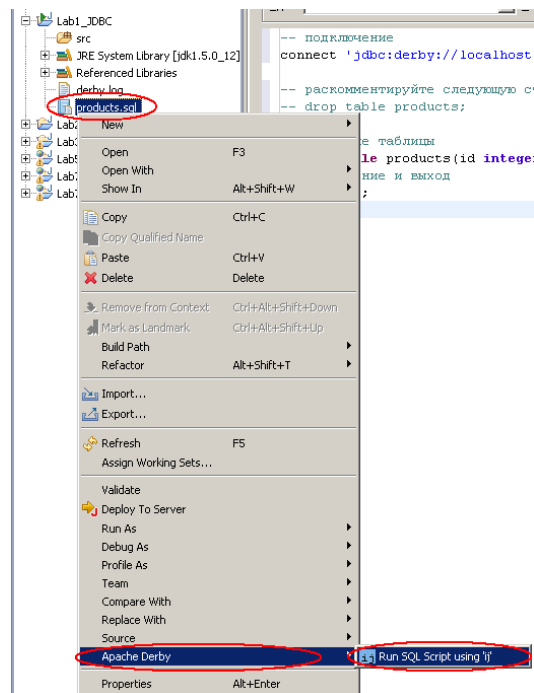
-- создание таблицы
create table products(id integer, description varchar(100), rate float, quantity integer);

-- отключение и выход
disconnect;
exit;

```

5) Сохраните файл нажатием на Ctrl-S.

6) Щелкните правой кнопкой мыши на файл products.sql в окне Package Explorer и выберите Apache Derby/Run SQL Script using 'ij'.



7) В случае успешного выполнения скрипта в консоли выводится следующее:

```

ij version 10.3
ij> -- подключение
connect
'jdbc:derby://localhost:1527/myDB;create=true;user=me;password=mine';
ij> -- раскомментируйте следующую строку, если требуется пересоздать
таблицу
-- drop table products;

-- создание таблицы
create table products(id integer, description varchar(100), rate float,
quantity integer);
0 rows inserted/updated/deleted

```

```

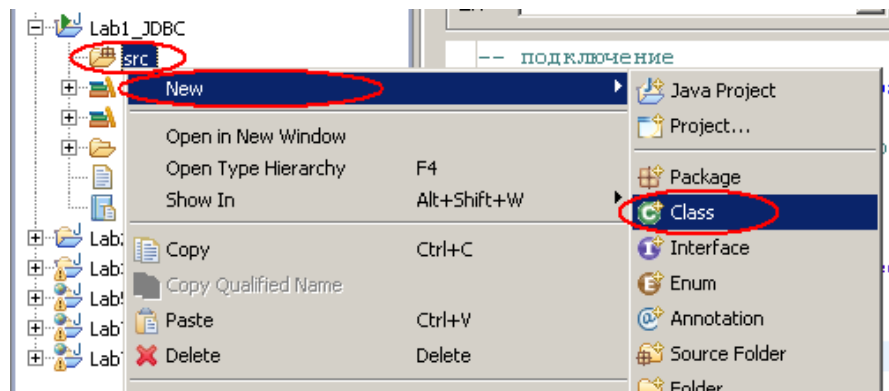
ij> -- отключение и выход
disconnect;
ij> exit;

```

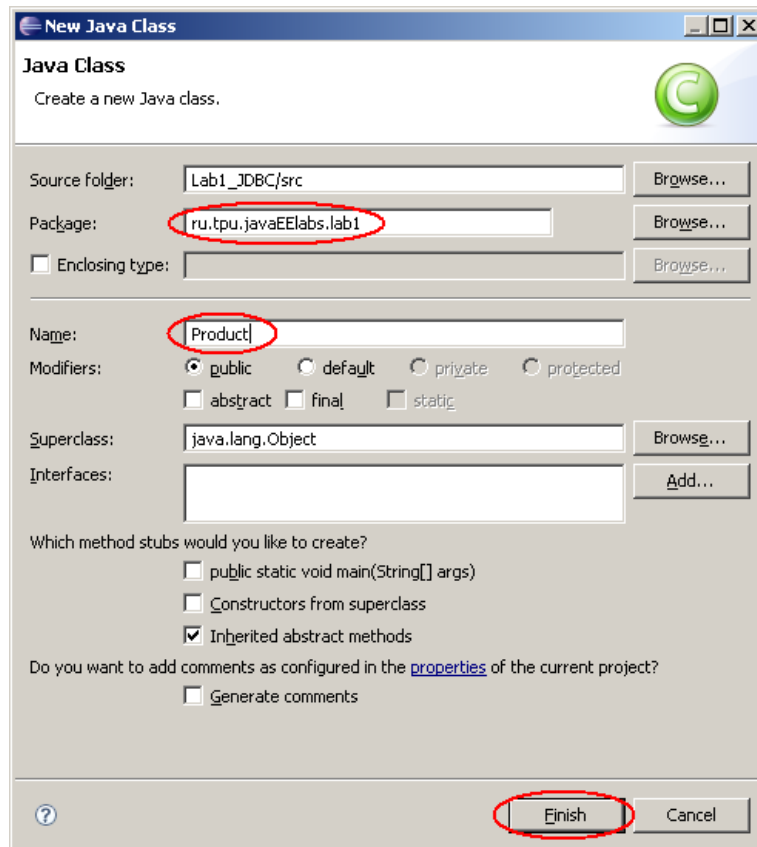
2.4 Создание класса Product

Класс Product является обычным Java-классом и будет использоваться для передачи и обработки данных о товаре. Класс содержит четыре поля – id, description, rate, quantity, соответствующие полям таблицы products.

1) В окне Package Explorer щелкните правой кнопкой мыши на каталог src проекта Lab1_JDBC и выберите пункт меню New/Class.



2) Назовите класс Product и задайте имя пакета (Package) ru.tpu.javaEElabs.lab1. Нажмите Finish.



3) В созданном классе необходимо реализовать четыре поля `id`, `description`, `rate`, `quantity`, создать конструктор и методы `get/set` для доступа к этим полям. Полный код класса `Product` приведен ниже:

```
package ru.tpu.javaEEElabs.lab1;

public class Product {

    private int id;
    private String description;
    private float rate;
    private int quantity;

    public Product(int id, String description, float rate, int quantity) {
        this.id = id;
        this.description = description;
        this.rate = rate;
        this.quantity = quantity;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public float getRate() {
        return rate;
    }

    public void setRate(float rate) {
        this.rate = rate;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
}
```

2.5 Создание класса обработки данных `ProductDAO`

На следующем этапе создадим класс `ProductDAO` (DAO – Data Access Object – объект доступа к данным), который скрывает детали реализации логики работы с БД – получение соединения, выполнение операций выборки, создания, обновления и удаления.

1) В окне Package Explorer щелкните правой кнопкой мыши на пакет ru.tpu.javaEElabs.lab1 проекта Lab1_JDBC и выбрав пункт меню New/Class.

2) Назовите класс ProductDAO и убедитесь, что имя пакета (Package) задано в ru.tpu.javaEElabs.lab1. Нажмите Finish.

Код класса ProductDAO приведен ниже:

```
package ru.tpu.javaEElabs.lab1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

/**
 */
public class ProductDAO {
    // Вспомогательный метод получения соединения
    private Connection getConnection() throws Exception {
        // Подгрузка драйвера БД Derby
        Class.forName("org.apache.derby.jdbc.ClientDriver").newInstance();
        // Получение соединения с БД
        return DriverManager.getConnection(
            "jdbc:derby://localhost:1527/myDB;create=true;user=me;password=mime");
    }

    /**
     * Возвращает список идентификаторов товаров
     *
     * @return
     */
    public List<Integer> getProductIds() throws Exception {
        List<Integer> productIds = new ArrayList<Integer>();
        // Получение соединения с БД
        Connection con = getConnection();

        // Выполнение SQL-запроса
        ResultSet rs = con.createStatement().executeQuery(
            "Select id From products");
        // Перечисляем результаты выборки
        while (rs.next()) {
            // Из каждой строки выборки выбираем
            // результат и помещаем в коллекцию
            productIds.add(rs.getInt(1));
        }
        // Закрываем выборку и соединение с БД
        rs.close();
        con.close();
        return productIds;
    }

    /**
     * Возвращает товар по идентификатору
     *
     * @return
     */
    public List<Product> getProductById(int id) throws Exception {
        List<Product> products = new ArrayList<Product>();
    }
}
```



```

// Получение соединения с БД
Connection con = getConnection();

// Подготовка SQL-запроса
PreparedStatement st = con.prepareStatement(
    "Select description, rate, quantity " +
    "From products " +
    "Where id = ?");
// Указание значений параметров запроса
st.setInt(1, id);

// Выполнение запроса
ResultSet rs = st.executeQuery();

Product product = null;
// Перечисляем результаты выборки
while (rs.next()) {
    // Из каждой строки выборки выбираем результаты,
    // формируем новый объект Product
    // и помещаем его в коллекцию
    product = new Product(
        id,
        rs.getString(1),
        rs.getFloat(2),
        rs.getInt(3));
    products.add(product);
}
// Закрываем выборку и соединение с БД
rs.close();
con.close();
return products;
}

/**
 * Добавляет новый товар
 * @param product
 * @throws Exception
 */
public void addProduct(Product product) throws Exception {
    // Получение соединения с БД
    Connection con = getConnection();

    // Подготовка SQL-запроса
    PreparedStatement st = con.prepareStatement(
        "Insert into products " +
        "(id, description, rate, quantity) " +
        "values (?, ?, ?, ?)");
    // Указание значений параметров запроса
    st.setInt(1, product.getId());
    st.setString(2, product.getDescription());
    st.setFloat(3, product.getRate());
    st.setInt(4, product.getQuantity());

    // Выполнение запроса
    st.executeUpdate();

    con.close();
}

/**
 * Обновляет данные о товаре
 * @param product
 * @throws Exception
 */

```

```

*/
public void setProduct(Product product) throws Exception {
    // Получение соединения с БД
    Connection con = getConnection();

    // Подготовка SQL-запроса
    PreparedStatement st = con.prepareStatement(
        "Update products " +
        "Set description=?, rate=?, quantity=? " +
        "Where id=?");
    // Указание значений параметров запроса
    st.setString(1, product.getDescription());
    st.setFloat(2, product.getRate());
    st.setInt(3, product.getQuantity());
    st.setInt(4, product.getId());

    // Выполнение запроса
    st.executeUpdate();
    con.close();
}

public void removeProduct(int id) throws Exception {
    // Получение соединения с БД
    Connection con = getConnection();

    // Подготовка SQL-запроса
    PreparedStatement st = con.prepareStatement(
        "Delete from products " +
        "Where id = ?");
    // Указание значений параметров запроса
    st.setInt(1, id);

    // Выполнение запроса
    st.executeUpdate();
    con.close();
}
}
}

```

2.6 Разработка клиентского приложения

Клиентское приложения для просмотра/редактирования товаров будет представлять собой диалоговое приложение Swing, содержащее список для выбора товара по идентификатору и полей, отображающих информацию по выбранному товару. Эти же поля будут использоваться для ввода/редактирования информации по добавляемому/изменяемому товару. Ниже будут располагаться кнопки «Добавить», «Обновить» и «Удалить». Обработчики нажатия на эти кнопки обращаются к полям формы и методам класса ProductDAO для выполнения соответствующих действий.

1) Создайте новый Java-класс с именем ProductInfoClient в пакете ru.tpu.javaEElabs.lab1. Класс наследует JDialog.

Код класса ProductInfoClient приведен ниже:

```

package ru.tpu.javaEElabs.lab1;

import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

import java.util.List;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;

public class ProductInfoClient extends JDialog {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    // Создаем DAO-объект
    ProductDAO productDAO = new ProductDAO();

    // Объявление элементов управления
    private JLabel lbSelectId = new JLabel("Выбор товара по Id");
    private JLabel lbId = new JLabel("Id");
    private JLabel lbDescription = new JLabel("Описание");
    private JLabel lbRate = new JLabel("Цена");
    private JLabel lbQuantity = new JLabel("Остаток");

    private JComboBox comboId = new JComboBox();
    private JTextField txtId = new JTextField();
    private JTextField txtDescription = new JTextField();
    private JTextField txtRate = new JTextField();
    private JTextField txtQuantity = new JTextField();

    private JButton btnAdd = new JButton("Добавить");
    private JButton btnUpdate = new JButton("Обновить");
    private JButton btnRemove = new JButton("Удалить");
    private JButton btnClear = new JButton("Очистить");

    /**
     * Создает экземпляр диалога
     * @param args
     */
    public static void main(String[] args) {
        new ProductInfoClient();
    }

    /**
     * Конструктор диалога
     */
    public ProductInfoClient() {
        this.setTitle("Информация о товарах");
        this.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
        this.setLayout(new GridLayout(8, 2));
        this.setBounds(100, 50, 400, 200);

        // Добавление элементов управления в диалог
        this.add(lbSelectId);
        this.add(comboId);
        this.add(lbId);
        this.add(txtId);
        this.add(lbDescription);
        this.add(txtDescription);
        this.add(lbRate);
        this.add(txtRate);
        this.add(lbQuantity);
    }
}

```

```

this.add(txtQuantity);
this.add(btnAdd);
this.add(btnUpdate);
this.add(btnRemove);
this.add(btnClear);

// Описание обработчиков событий
comboId.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        showProductData();
    }
});

btnAdd.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        addProduct();
    }
});

btnUpdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        updateProduct();
    }
});

btnRemove.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        removeProduct();
    }
});

btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        clearProductInfo();
    }
});

// Обновляем список идентификаторов товаров
refreshIdList();

// Отображаем диалог на экране
this.setVisible(true);
}

/**
 * Считывает список идентификаторов товаров
 * и заполняет список
 */
private void refreshIdList() {
    try {
        // получаем список идентификаторов
        List<Integer> productIds = productDAO.getProductIds();
        // очищаем список
        comboId.removeAllItems();
        // заполняем список полученными данными
        for (Integer productId: productIds) {
            comboId.addItem(productId);
        }
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}
}

```

```

/**
 * Отображает данные о товаре по
 * выбранному в списке идентификатору
 */
protected void showProductData () {
    try {
        // забираем значение, выбранное в списке
// идентификаторов товаров
        Integer productId = (Integer) comboId.getSelectedItem();

        if (productId != null) {
            // получаем товар по идентификатору

            List<Product> product = product-
DAO.getProductById(productId);
            // заполняем текстовые поля значениями
// параметров товара
            for(int i=0; i<product.size(); i++){
                txtId.setText (String.valueOf(product.get(i).getId()));
                txtDescrip-
tion.setText (product.get(i).getDescription());

                txtRate.setText (String.valueOf(product.get(i).getRate()););
                txtQuanti-
ty.setText (String.valueOf(product.get(i).getQuantity()));
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

/**
 * Добавляет новый товар на основе
 * данных текстовых полей
 */
protected void addProduct () {
    try {
        // создаем новый объект-товар
        // на основе данных диалога
        Product product = new Product(
            Integer.parseInt(txtId.getText()),
            txtDescription.getText(),
            Float.parseFloat(txtRate.getText()),
            Integer.parseInt(txtQuantity.getText()));
        // сохраняем товар в БД
        productDAO.addProduct(product);
        // обновляем список идентификаторов
        refreshIdList();
        // устанавливаем текущим добавленный товар
        comboId.setSelectedItem(
Integer.parseInt(txtId.getText()));
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

/**
 * Обновляет информацию о товаре на основе
 * данных текстовых полей

```

```

*/
protected void updateProduct () {
    try {
        // формируем объект-товар
        // на основе данных диалога
        Product product = new Product(
            Integer.parseInt(txtId.getText()),
            txtDescription.getText(),
            Float.parseFloat(txtRate.getText()),
            Integer.parseInt(txtQuantity.getText()));
        // обновляем данные о товаре в БД
        productDAO.setProduct(product);
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

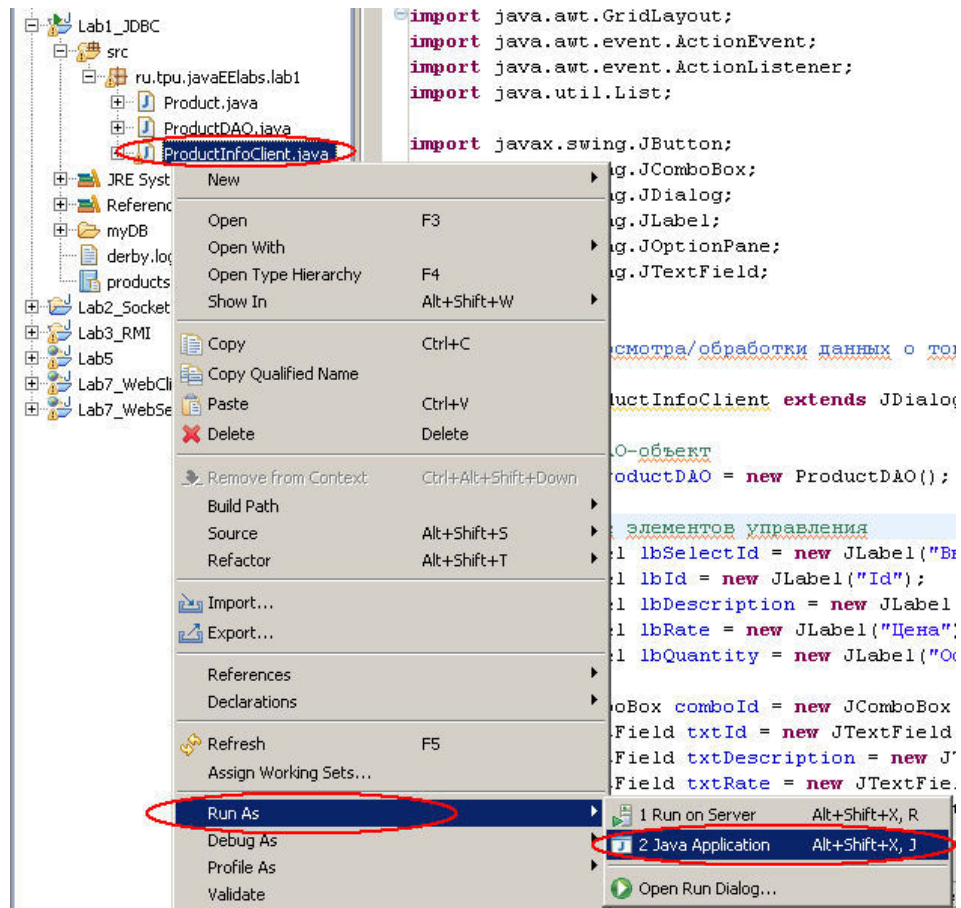
/**
 * Удаляет выбранный товар
 */
protected void removeProduct () {
    try {
        // удаляем товар из БД
        productDAO.removeProduct(
Integer.parseInt(txtId.getText()));
        // обновляем список идентификаторов товаров
        refreshIdList();
        // отображаем данные по первому товару в списке
        showProductData();
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

/**
 * Очищает данные в текстовых полях
 */
protected void clearProductInfo () {
    try {
        txtId.setText("");
        txtDescription.setText("");
        txtRate.setText("");
        txtQuantity.setText("");
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}
}

```

2.7 Запуск и тестирование приложения

1) Щелкните правой кнопкой мыши на класс ProductInfoClient в окне Package Explorer и выберите команду Run As/Java Application.



2) Проверьте работу приложения – добавьте несколько товаров, проверьте функции обновления и удаления.

Информация о товарах	
Выбор товара по Id	2
Id	2
Описание	апельсины
Цена	60.3
Остаток	700
Добавить	Обновить
Удалить	Очистить

3. ВАРИАНТЫ ЗАДАНИЙ

1. Необходимо разработать приложение для учета компакт-дисков в музыкальном салоне. Приложение должно позволять пользователю просматривать, добавлять, изменять и удалять информацию о дисках. Описание диска должно содержать следующие поля: идентификатор (id), жанр, исполнитель, название, год. Необходимо обеспечить отображение

списка дисков в виде таблицы и возможность просмотра дисков по жанру, выбранному пользователем из выпадающего списка.

2. Необходимо разработать приложение для учета книг в книжном магазине. Приложение должно позволять пользователю просматривать, добавлять, изменять и удалять информацию о книгах. Описание книги должно содержать следующие поля: идентификатор (id), жанр, ФИО автора, название, год. Необходимо обеспечить отображение списка книг в виде таблицы и возможность фильтрации книг по ФИО автора, введенной пользователем в текстовом поле.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Шувалов В.П., Величко В.В., Субботин Е.А., Ярославцев А.Ф. Телекоммуникационные системы и сети. Том 3. Мультисервисные сети (2005)
- 2) Петраков А.В. Основы практической защиты информации. 2-е изд. Учебн. пособие. – М.: Радио и связь. 2000. – 368 с.
- 3) Цифровые и аналоговые системы передачи: Учебник для вузов/ В.И.Иванов, В.Н.Гордиенко, Г.Н.Попов и др.; Под ред. В.И.Иванова. – 2-е изд. – М.: Горячая линия – Телеком, 2003. – 232 с.
- 4) Гольдштейн Б.С., Соколов Н.А., Яновский Г.Г. Сети связи: Учебник для ВУЗов. - СПб.: БХВ-Петербург, 2010. - 400 с.
- 5) Башарин Г.П. Лекции по математической теории телетрафика: Учеб. пособие. Изд. 3-е, испр. и доп. - М.: РУДН, 2009. - 342 с.
- 6) Буч Г. Объектно-ориентированный анализ и проектирование. – М.: Вильямс, 2008.
- 7) Леоненков А.В. Самоучитель языка UML. – СПб.: БХВ-Петербург, 2004.
- 8) Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов. – М.: ДМК Пресс, 2002.
- 9) А.В. Росляков. Виртуальные частные сети. Основы построения и применения. - М.: Эко-Трендз, 2006. - 242 с.