

Ю. В. Андрусенко, М. И. Жадан

(ГГУ им. Ф. Скорины, Гомель)

РАЗРАБОТКА КЛИЕНТ – СЕРВЕРНОГО ПРИЛОЖЕНИЯ НА JAVA

Для того чтобы приступить к дальнейшему рассмотрению этой темы, сначала необходимо дать определение таким понятиям, как сервер и клиент. Сервер – это специальное приложение, обычно запущенное на отдельном компьютере, выполняющее некий круг задач. Также сервер можно называть хостом (от англ. Host – хозяин). Кли-

ент – это приложение, запрашивающее хост выполнить то или иное действие и вернуть полученные данные клиенту. На хосте для работы сервера обычно выделяется порт, к которому должен будет обращаться клиент. Клиент для связи с портом хоста, соединенный в свою очередь с нужным сервером, создает сокет.

Алгоритм работы системы клиент-сервер выглядит следующим образом:

- сервер подключается к порту на хосте и ждет соединения с клиентом;
- клиент создает сокет и пытается соединить его с портом на хосте;
- если создание сокета прошло успешно, то сервер переходит в режим ожидания команд от клиента;
- клиент формирует команду и передает ее серверу, переходит в режим ожидания ответа;
- сервер принимает команду, выполняет ее и пересылает ответ клиенту.

В работе реализованы два клиент-серверные приложения на языке Java, в результате чего решены следующие задачи.

Задача 1. Создать сервер необходимый для создания сокетов с возможностью их управления при помощи специальных классов `java.net.Socket` и `java.net.ServerSocket` для клиента и хоста соответственно, и двух классов из пакета `java.io.*`: `BufferedReader` и `PrintWriter` для чтения и записи в сокет. Создать возможность подключения к порту сервера с использованием конструктора класса `ServerSocket`. При возникновении ошибки, конструктор должен выбрасывать исключение типа `IOException`. Необходимо использовать блок `try-catch`. Подключения клиента выполнить методом `servers.accept()`. Сервером должны создаваться потоки ввода и вывода для связи.

Задача 2. Создать клиента с использованием класса `Socket` и двух классов для ввода/вывода. Сокет должен принимать два параметра: имя компьютера, на котором запущен сервер, и номер порта. В случае возникновения ошибки, конструктор должен выбрасывать исключение типа `IOException`. Создать потоки ввода/вывода аналогичные потокам сервера.