Факультет комп’ютерних наук

Залікова робота

«*Крос-платформне програмування*»

**Варіант №2**

П.І.Б. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Група \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Особливості SAX та DOM парсерів. Основи створення та використання SAX, DOM парсерів. (*10 балів*)
2. *JavaBeans* компоненти. Підтримка простих властивостей та властивостей з обмеженнями (simple, constrained properties). (*10 балів*)
3. Основи роботи з TCP сокетами. Основні етапи створення мережевого додатку за допомогою TCP сокетів. Приклад простого TCPсервера та клієнта: клієнт відправляє рядок на сервер, сервер повертає рядок великими літерами. (*10 балів*)

Задание выполнено по аналогии с Лабораторной работой №6

Необходимо разработать минимум три класса: Сервер, Клиент, Хендлер. Приложение делится на две чести: клиентскую(Client) и серверную(Server, Handler). Клиент и сервер хранят сокеты ServerSocket и Socket соответственно.

Клиентская часть.

В классе Client создается Socket, в конструктор передаются InetSocketAddress, который состоит из адреса сервера и номера порта, и время задержки.

Также Client хранит OI и OO потоки (ObjectInputStream, ObjectOutputStream), полученные из Socket. Это необходимо для прямого взаимодействия с сервером.

Важно: все объекты которые передаются через потоки должны быть сериализуемы.

Далее в отдельном методе формируется запрос в который помещается исходная строка. Метод отправляет через oos строку на сервер. Далее метод ожидает появления объекта в ois и выводит ее в консоль.

Серверная часть.

В классе Server содержится ServerSocket, в конструктор которого передается номер порта (номера портов клиента и сервера должны совпадать). В отдельном потоке содержится бесконечный цикл, который постоянно пытается принять запрос на соединение от клиента. Как только это происходит, сокет клиента сохраняется и передается хендлеру.

В классе Handler выполняется обработка запроса. Так же как и в Client там хранятся oos и ois. В потоке хендлера из ois принимается исходная строка, изменяется и помещается в oos. После этого Handler закрывается.

1. Основи *RMI*. Наведіть приклад простого розподіленого додатку, виконаного за технологією *RMI*, в якому клієнт надає серверу рядок, а сервер повертає клієнту цей рядок подвоєним. (*10 балів*)

Взаимодействие происходит через интерфейсы compute.jar. Это общие файлы для сервера и клиента. Далее в пакете клиента формируется запрос в классе ComputeString. Класс DoubledString содержит задание, которое будет выполнено на сервере.

Сервер принимает объект класса, наследника интерфейса Task (в нашем случае это DoubledString), применяет метод execute и отправляет ответ.

Клиент печатает ответ.