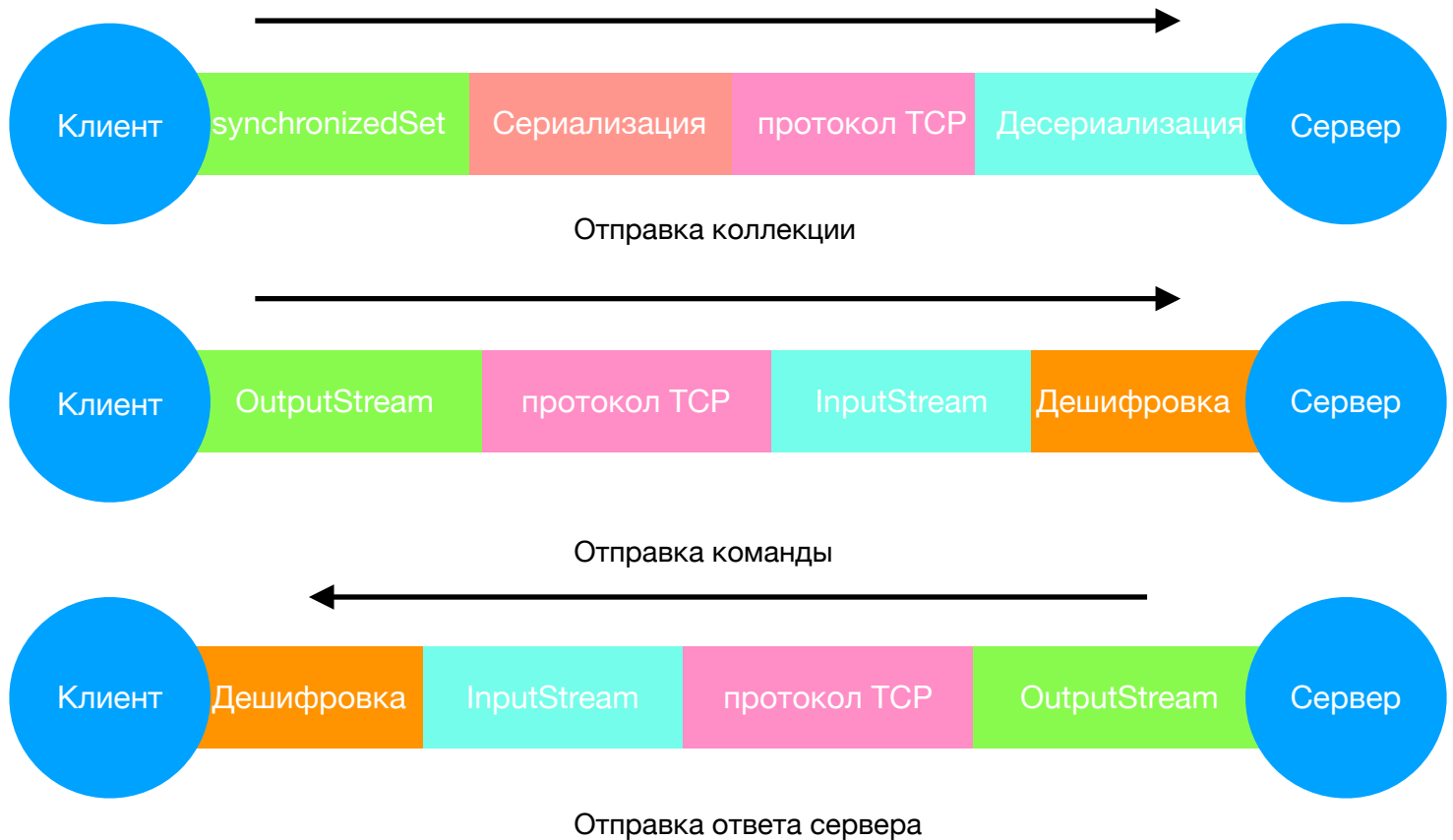


Как это работает

Клиент	Сервер
Выполнение кода из лабораторной №3	Прием команд
Отправка содержимого коллекции или самой коллекции	Прием объектов коллекции
Прием команд из консоли	Прием коллекции
Отправка команд на сервер	Выполнение команд
	<p>Коллекция должна содержать ноды-специальные объекты содержащие объект исходной коллекции и следующие поля:</p> <ul style="list-style-type: none"> • имя, название или аналогичный текстовый идентификатор (результат вызова toString); • размер или аналогичный числовой параметр; • характеристику, определяющую местоположение объекта на плоскости/в пространстве(номер в массиве); • время/дату рождения/создания объекта.



Как распределяется труд

Гриша	Олег
Сделать работу с потоками	Переделать команды в формат StreamAPI с лямбда выражениями
Заставить сервер общаться с клиентом и наоборот	

Материалы:

Сериализация и десериализация- <https://javarush.ru/groups/posts/2022-serializacija-i-deserializacija-v-java>

Потокобезопасные коллекции (synchronizedSet)- <http://java-online.ru/concurrent-collections.xhtml>

StreamAPI и лямбды- <https://habr.com/ru/company/luxoft/blog/270383/>

Принцип работы TCP- <https://se.ifmo.ru/documents/10180/639377/3+лекция+2+семестр/f5839fa7-3fdd-4b0d-9dbf-0901cece0ee3> слайд 57

Вопросы к защите лабораторной работы:

1. Сетевое взаимодействие - клиент-серверная архитектура, основные протоколы, их сходства и отличия. <http://www.4stud.info/networking/lecture5.html>
2. Протокол TCP. Классы **Socket** и **ServerSocket**. <http://pro-java.ru/rabota-s-setyu-java/klientskie-sokety-po-protokolu-tcpip-v-java/>
3. Протокол UDP. Классы **DatagramSocket** и **DatagramPacket**. <http://crypto.pp.ua/2010/06/datagrammy-i-protokol-udp-v-java/>
4. Передача данных по сети. <http://streletzcoder.ru/obmen-dannyimi-po-seti-na-osnove-soketov-v-java/> Сериализация объектов. <https://javarush.ru/groups/posts/2022-serializacija-i-deserializacija-v-java>
5. Интерфейс **Serializable**. Объектный граф, сериализация и десериализация полей и методов. <https://habr.com/ru/post/60317/>
6. Многопоточные программы. Концепции.
7. Класс **Thread** и интерфейс **Runnable**. <http://developer.alexanderklimov.ru/android/java/thread.php>
8. Состояние потока. <https://javadevblog.com/zhiznennyj-tsikl-java-potoka-sostoyaniya-potoka-v-java.html> Синхронизация потока. <https://javarush.ru/groups/posts/1994-sinkhronizacija-potokov-operator-synchronized>
9. Пакет **java.util.concurrent**. Интерфейс **Lock** и его реализации. <https://habr.com/ru/company/luxoft/blog/157273/>
10. Атомарные операции. <https://javarush.ru/groups/posts/2060-threadom-java-ne-isportishjh--chastijh-iii---vzaimodeystvie>
11. Java Stream API. Создание конвейеров. Промежуточные и терминальные операции.