**Теория**

1. Что такое сериализация / десериализация?
2. Какая последовательность работы с потоком?
3. Какие виды потоков существуют?
4. Чем отличается IO от NIO?

1) Сериализация — это **процесс преобразования объекта в поток байтов для сохранения или передачи в память, базу данных или файл**. Эта операция предназначена для того, чтобы сохранить состояния объекта для последующего воссоздания при необходимости. Обратный процесс называется десериализацией.

2) **Последовательность работы с потоками в Java:**

1. Создаётся экземпляр необходимого потока.
2. Поток открывается для дальнейшего считывания или записи новой информации (при необходимости).
3. Пользователь проводит задуманные действия (чтение и запись информации).
4. Поток закрывается

3) В Java существует два типа потоков — пользовательские (те, которые мы создаем) **потоки и потоки-демоны**. Когда запускается Java-программа, сразу же начинается выполняться один поток — основной поток. Основной поток запускает метод main(). Мы можем создавать новые потоки из основного потока.

4) Основное отличие между двумя подходами к организации ввода/вывода в том, что Java IO является потокоориентированным, а Java NIO – буфер-ориентированным.

Потокоориентированный ввод/вывод подразумевает чтение/запись из потока/в поток одного или нескольких байт в единицу времени поочередно. Данная информация нигде не кэшируются. Таким образом, невозможно произвольно двигаться по потоку данных вперед или назад. Чтобы произвести подобные манипуляции, придется сначала кэшировать данные в буфере.

Подход, на котором основан Java NIO немного отличается. Данные считываются в буфер для последующей обработки. Можно двигаться по буферу вперед и назад. Это дает немного больше гибкости при обработке данных. В то же время, необходимо проверять содержит ли буфер необходимый для корректной обработки объем данных. Также необходимо следить, чтобы при чтении данных в буфер вы не уничтожили ещё не обработанные данные, находящиеся в буфере.