

Chapter 2

RQs + PEs:
Introduction to NIO

Advanced Topics in Java

Khalid Azim Mughal
khalid@ii.uib.no
<http://www.ii.uib.no/~khalid/atij/>

Version date: 2004-08-31

REVIEW QUESTIONS - NIO

1. Explain what a *buffer* is and what its attributes are ?
2. How are the buffer attributes affected when a buffer is filled, when it is flipped and when it is drained?
3. What is the difference between direct and nondirect buffers?
4. What are the advantages of using NIO?
5. What is a *channel*?
6. What is a *file channel* and how would you create one for a file?
7. What is a *socket channel*? Name the socket channel classes and their corresponding peer socket classes.
8. Why is *multiplexing* useful for implementing client-server applications?
9. Explain what is meant by *readiness selection*? Which classes make this mechanism possible?
10. Which method is invoked on the selector to monitor the registered channels?
11. A *selection key* encapsulates relationship between which two objects? What is the purpose of a selection key?
12. What is the *selection key set* and how is it generated?

13. Which of these statements are true. Explain your reasoning.

- All primitive types have a corresponding buffer class.
- All buffer classes in the `java.nio` package implement a constructor which can be called to create a buffer.
- The `java.nio` package has a class named `BooleanBuffer`.
- File channels can operate in *nonblocking* mode.
- A channel can be *bidirectional*.
- The method call `channel.read(buffer)` reads from the buffer to the channel.
- The method call `channel.write(buffer)` reads from the buffer to the channel.
- Socket channels can only operate in nonblocking mode.
- A server socket channel cannot be used for reading or writing between buffers and socket connections.
- File channels are selectable.
- A selectable channel is registered with a selection key.
- The call `channel.register(selector, SelectionKey.OP_READ)` will register the channel with the selector to monitor reading activity on the channel.
- The call `selector.register(channel, SelectionKey.OP_READ)` will register the channel with the selector to monitor reading activity on the channel.
- The option `SelectionKey.OP_READ` indicates that the registered channel is interested in knowing when a connection wants to send data to the channel.

PROGRAMMING EXERCISES - NIO

1. Write a program that reads a file and write its contents on the console. The program assumes that the file contents are encoded in US-ASCII charset. It repeatedly fills a byte buffer with bytes from the file, which are decoded into a char buffer. The contents of the char buffer are then printed on the console.
2. Modify the multiplexing server (`MultiplexingServer.java`) so that I/O activity for each client identified by a selected key is handled in a separate thread.

Tip: Implement a nested class which encapsulates the functionality of the `replyClient()` method executed in a thread. The byte buffer used in the `MultiplexingServer` class should also be declared in the nested class so that each thread has its own buffer.