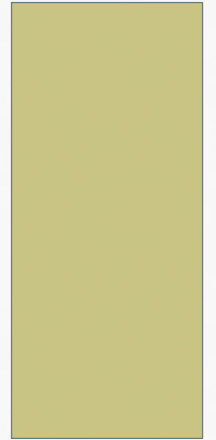
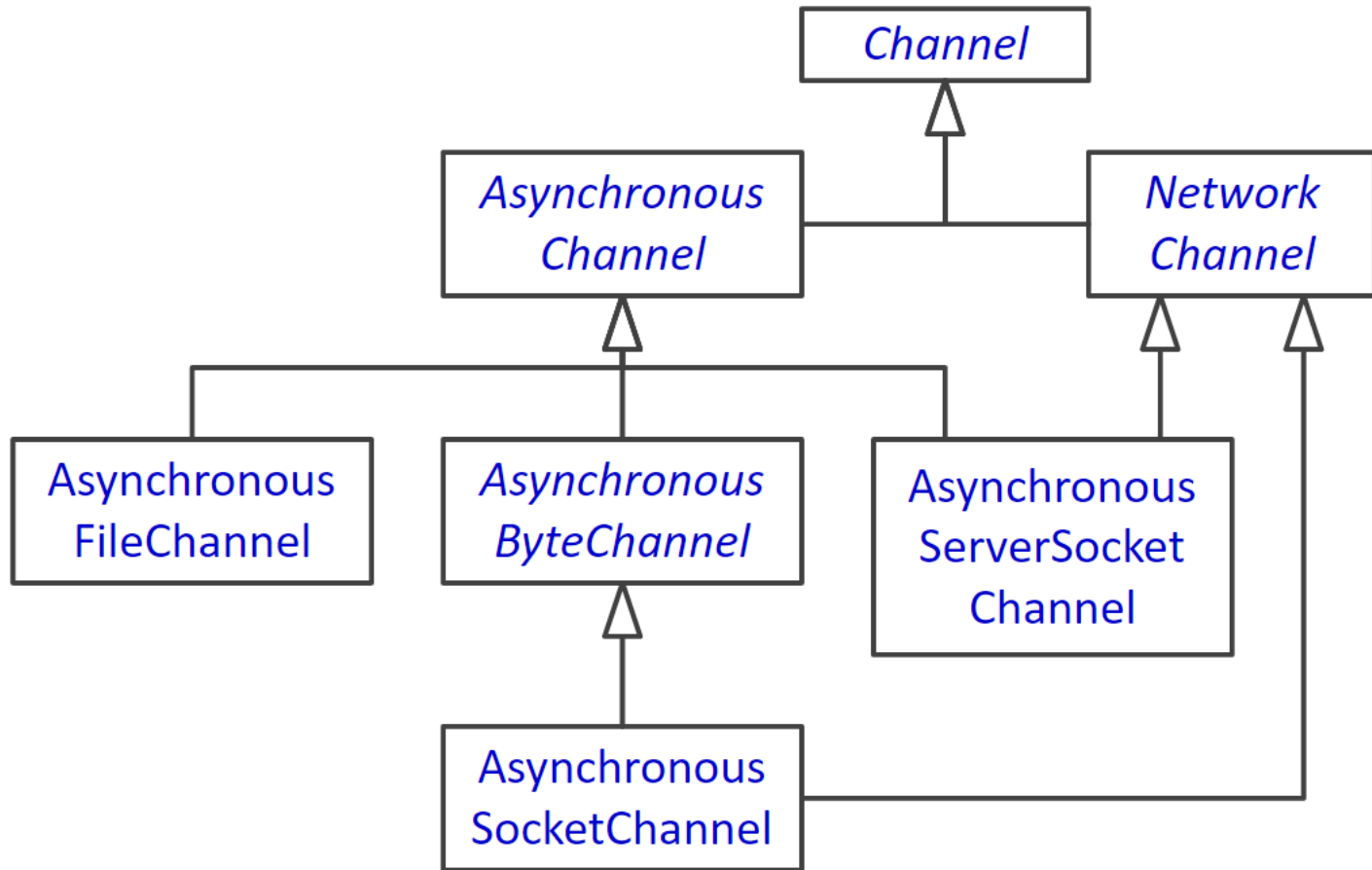


# Java Network Programming



# Асинхронные каналы



# Асинхронный ввод/вывод

- Интерфейс `AsynchronousChannel`
- Операция
  - Результат типа `V`
  - `Future<V> op(...)`
  - `op(..., A attachment, CompletionHandler<V, A>)`
- Отмена операции
  - `future.cancel()`
  - Операция выполнена частично  $\Rightarrow$  все сломалось, повторять нельзя

# Обработка результата

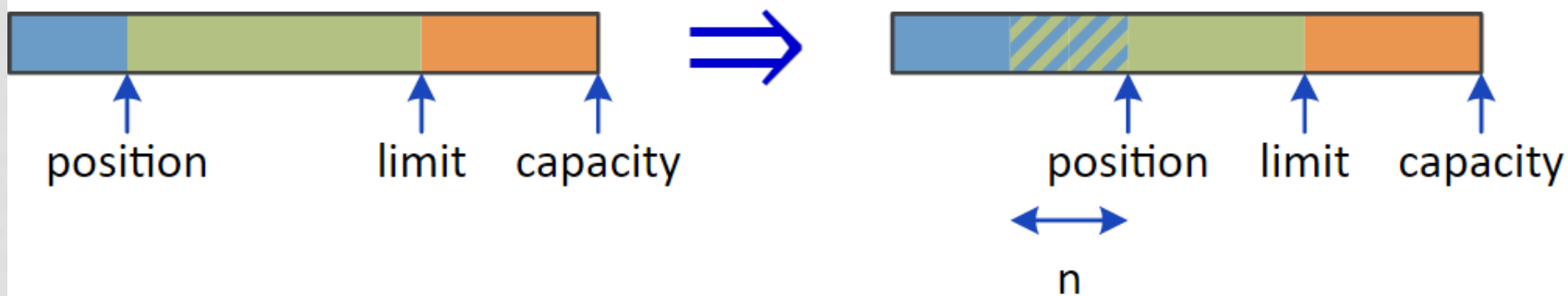
- Интерфейс `CompletionHandler<V, A>`
- Возможные результаты
  - Успех `completed(v, a)`
  - Ошибка `failed(exception, A)`

# Пример

```
socketChannel.read(  
    buffer,  
    new CompletionHandler<Integer, Context>() {  
        public void completed(Integer result, Context context) {  
            if (context.process(result)) {  
                socketChannel.read(buffer, this);  
            }  
        }  
        public void failed(Throwable e, Context context) {  
            context.error();  
            e.printStackTrace();  
        }  
    }  
);
```

# Асинхронный байтовый канал

- Интерфейс `AsynchronousByteChannel`
- Операции
  - Чтение `read(buffer)`, `read(buffer, A, handler)`
  - Запись `write(buffer)`, `write(buffer, A, handler)`
  - Результат `Integer n` — количество байт
  - Параллельная запись может не поддерживаться



# Асинхронный сокетный канал

- Класс `AsynchronousSocketChannel`
- Создание
  - Не привязанный `ASC.open()`
- Действия
  - Привязать `bind(socketAddress)`
  - Прекратить ввод `shutdownInput()`,
  - Прекратить вывод `shutdownOutput()`
- Установить соединение
  - `<Void> connect(socketAddress, ...)`

# Асинхронный серверный канал

- Класс `AsynchronousServerSocketChannel`
- Создание
  - `ASSC.open()`
- Действия
  - Привязать `bind(socketAddress)`, `bind(socketAddress, backlog)`
- Установить соединение
  - `<AsynchronousSocketChannel> accept(...)`
  - `AcceptPendingException`



# Группы асинхронных каналов

- Класс `AsynchronousChannelGroup`
- Создание
  - `ACG.withThreadPool(executor)`
  - `ACG.withCachedThreadPool(executor, min)`
  - `ACG.withFixedThreadPool(threads, threadFactory)`
- Использование
  - `ASSC.open(this)`, `ASC.open(this)`
- Завершение
  - `shutdown()`, `shutdownNow()`, `isShutdown()`
  - `isTerminated()`, `awaitTermination(time, unit)`
  - ```
final AsynchronousServerSocketChannel serverChannel =  
AsynchronousServerSocketChannel.open(group);
```