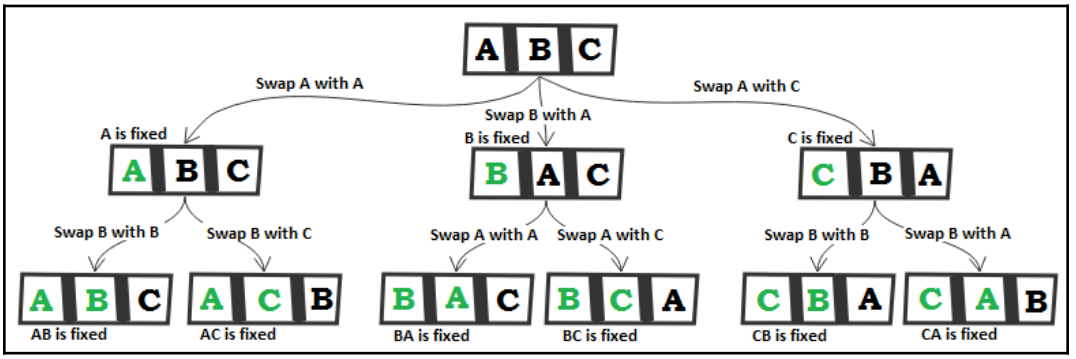
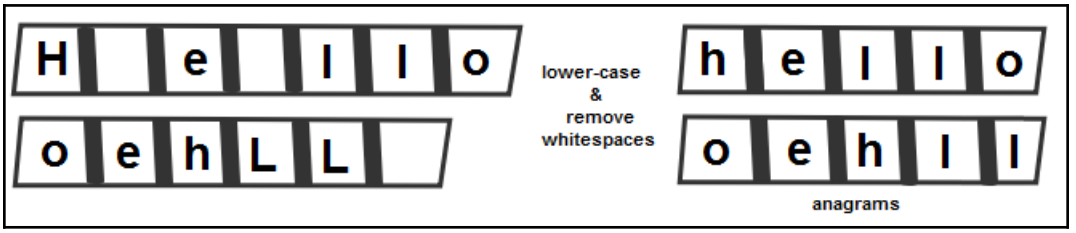
Chapter 1: Strings, Numbers, and Math






```
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

```
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

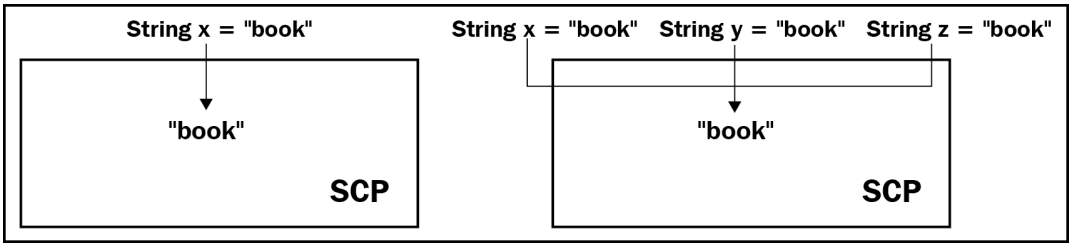
```
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

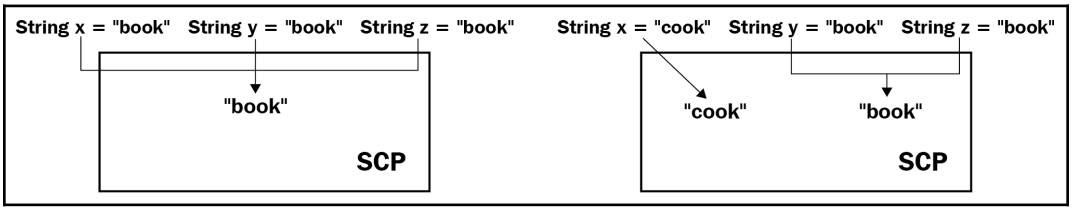
```
<html>
    <body>
        <h2>
            Hello world!
        </h2>
    </body>
</html>
```

X	Y	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

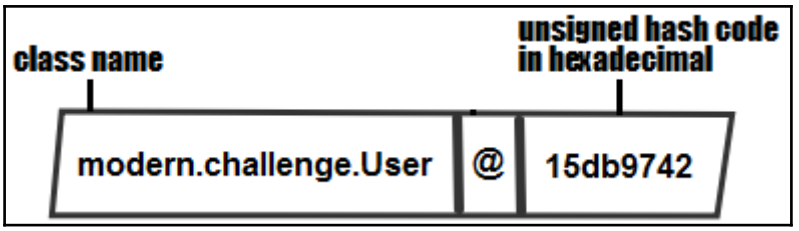
Chapter 2: Objects, Immutability, and Switch Expressions

```
requireNonNullElseThrow(T obj, X exception)                                  T
requireNonNullElseThrowIAE(T obj, String message)                            T
requireNonNullElseThrowIAE(T obj, Supplier<String> messageSupplier)          T
requireNotNullElseThrow(T obj, Supplier<? extends X> exceptionSupplier)      T
```



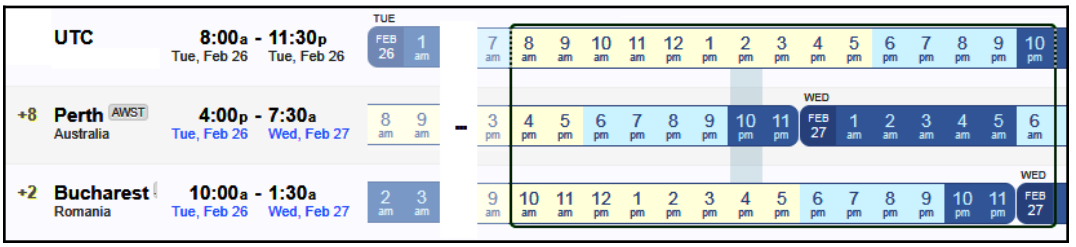



```
deepClone(T o)                                                               T
deepCloneDontCloneInstances(T o, Object... dontCloneThese)                   T
fastCloneOrNewInstance(Class<T> c)                                           T
shallowClone(T o)                                                            T
copyPropertiesOfInheritedClass(T src, E dest)                             void
dontClone(Class<?>... c)                                                  void
dontCloneInstanceOf(Class<?>... c)                                        void
equals(Object obj)                                                     boolean
getClass()                                                             Class<?>
getDumpCloned()                                                    IDumpCloned
...
```

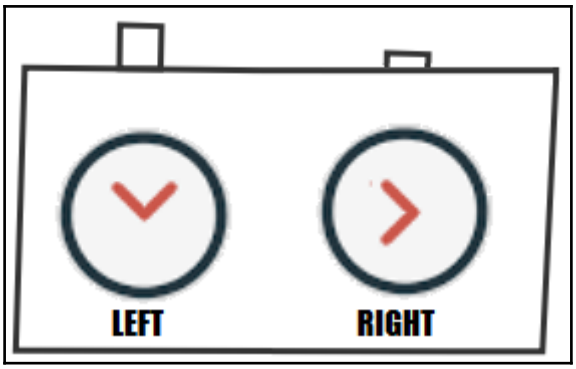



```
57
58      the switch expression does not cover all possible input values
        ----
        (Alt-Enter shows hints)                    tePlayer(PlayerTypes playerType) {
            return switch (playerType) {
60              case TENNIS->
61                  new TennisPlayer();
```

Chapter 3: Working with Date and Time




```
DEFAULT_TIME_ZONE                                                    ZoneId
dateToInstant(Date date)                                             Instant
dateToLocalDate(Date date)                                           LocalDate
dateToLocalDateTime(Date date)                                       LocalDateTime
dateToLocalTime(Date date)                                           LocalTime
dateToOffsetDateTime(Date date)                                      OffsetDateTime
dateToOffsetTime(Date date)                                          OffsetTime
dateToZonedDateTime(Date date)                                       ZonedDateTime
instantToDate(Instant instant)                                       Date
localDateTimeToDate(LocalDateTime localDateTime)                     Date
localDateToDate(LocalDate localDate)                                 Date
localTimeToDate(LocalTime localTime)                                 Date
offsetDateTimeToDate(OffsetDateTime offsetDateTime)                  Date
offsetTimeToDate(OffsetTime offsetTime)                              Date
zonedDateTimeToDate(ZonedDateTime zonedDateTime)                     Date
class
```


LEFT
RIGHT

Chapter 4: Type Inference

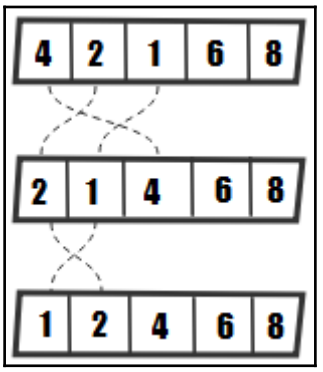
Decompilation of the class containing this method:

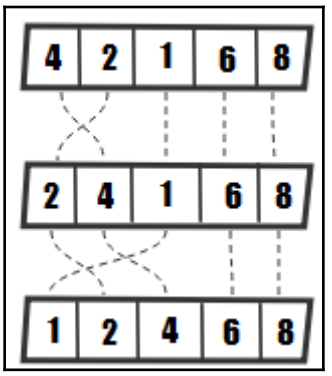
```
public Object fetchTransferableData(String data)
     throws UnsupportedFlavorException, IOException {

        StringSelection ss = new StringSelection(data);
        DataFlavor[] df = ss.getTransferDataFlavors();
        Object obj = ss.getTransferData(df[0]);

        return obj;
}
```

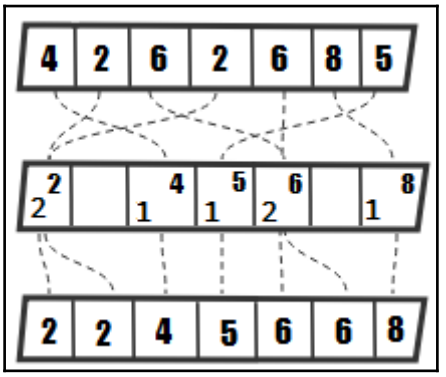
Decompilation of the class containing these declarations:

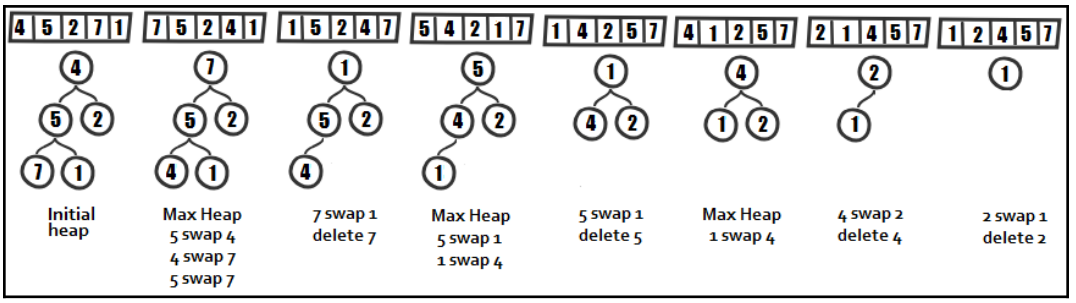
```
int intNumber = 10;
int longNumber = 10;
int floatNumber = 10;
int doubleNumber = 10;
```

Chapter 5: Arrays, Collections, and Data Structures



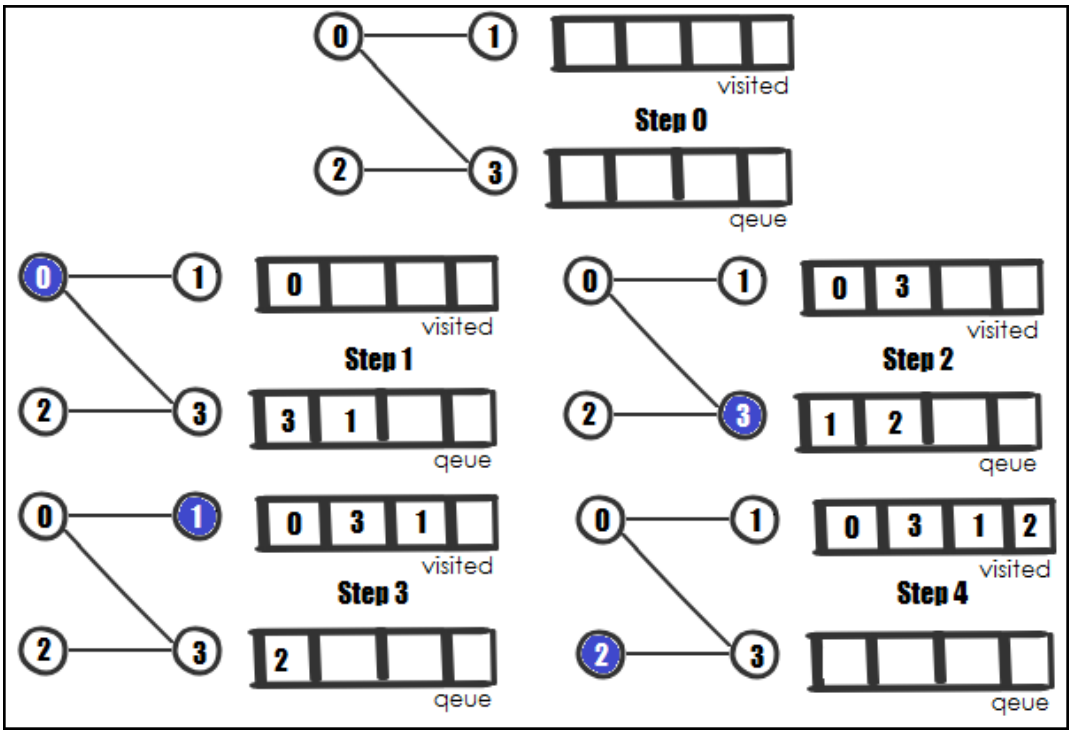


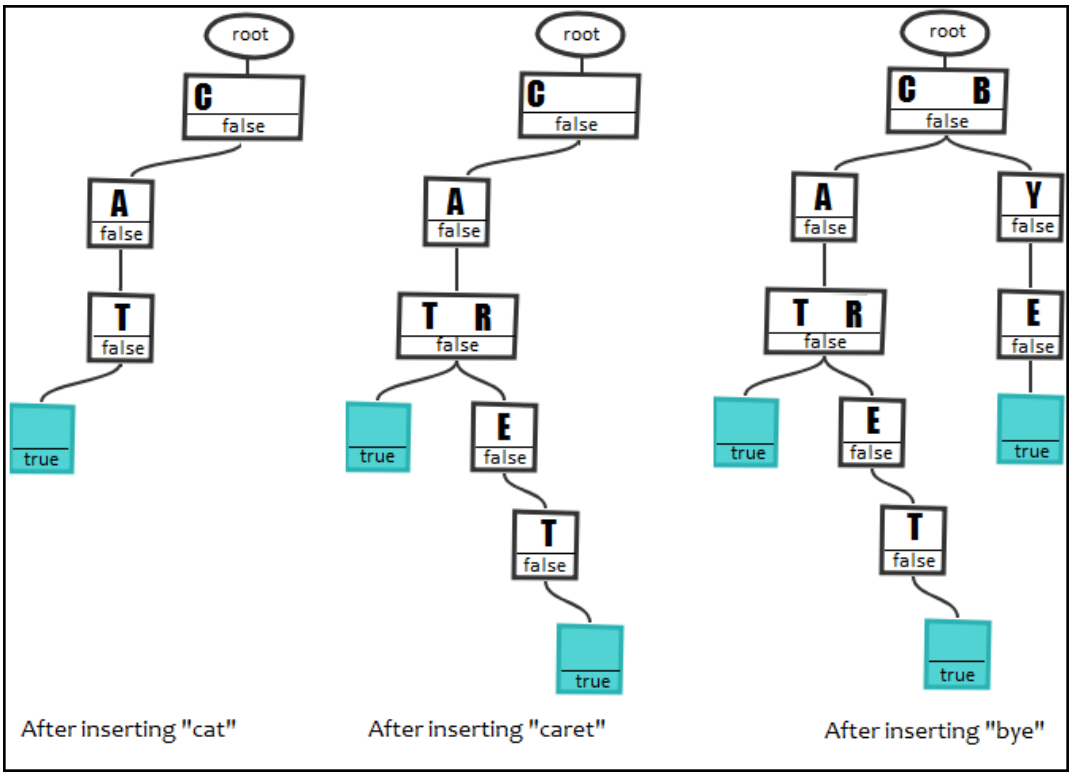


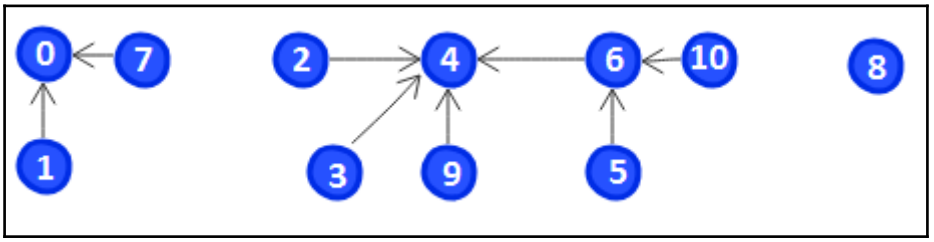


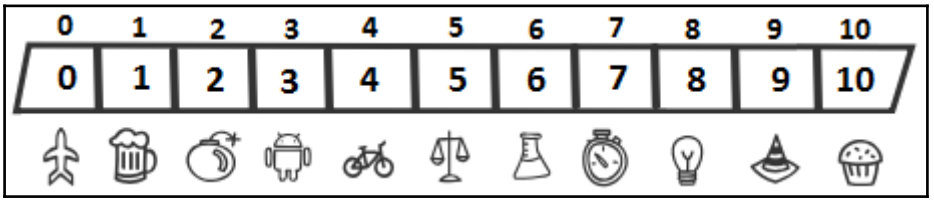

```
containsElementObjectV1(T[] arr, T toContain)                                  boolean
containsElementObjectV2(T[] arr, T toContain, Comparator<? super T> c)         boolean
containsElementObjectV3(T[] arr, T toContain, Comparator<? super T> c)         boolean
containsElementV1(int[] arr, int toContain)                                    boolean
containsElementV2(int[] arr, int toContain)                                    boolean
containsElementV3(int[] arr, int toContain)                                    boolean
findIndexOfElementObjectV1(T[] arr, T toFind)                                      int
findIndexOfElementObjectV2(T[] arr, T toFind, Comparator<? super T> c)             int
findIndexOfElementObjectV3(T[] arr, T toFind, Comparator<? super T> c)             int
findIndexOfElementV1(int[] arr, int toFind)                                        int
findIndexOfElementV2(int[] arr, int toFind)                                        int
```

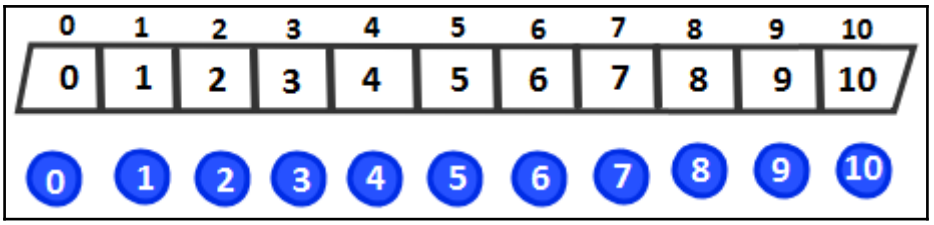
```
sortByKeyList(Map<K, V> map)                                          List<K>
sortByKeyStream(Map<K, V> map, Comparator<? super K> c)            Map<K, V>
sortByKeyTreeMap(Map<K, V> map)                                 TreeMap<K, V>
sortByValueList(Map<K, V> map)                                        List<V>
sortByValueStream(Map<K, V> map, Comparator<? super V> c)          Map<K, V>
```

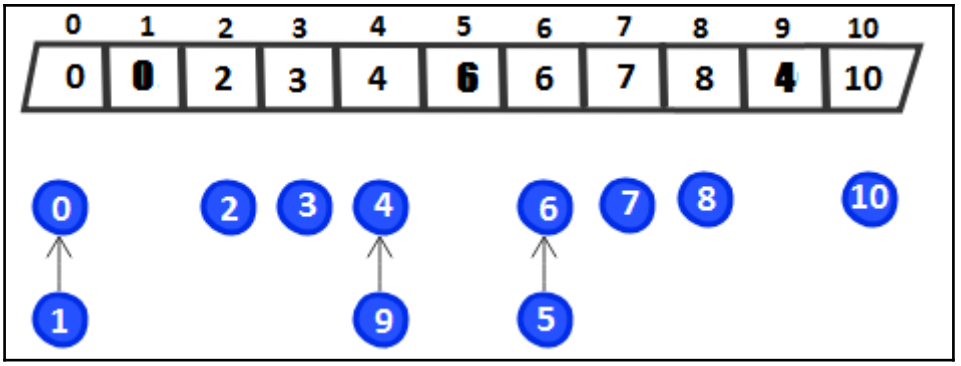

0
1
2
3
visited
Step 0
qeue
0
visited
Step 1
3
1
qeue
0
3
visited
Step 2
1
2
qeue
0
3
1
visited
Step 3
2
qeue
0
3
1
2
visited
Step 4
qeue

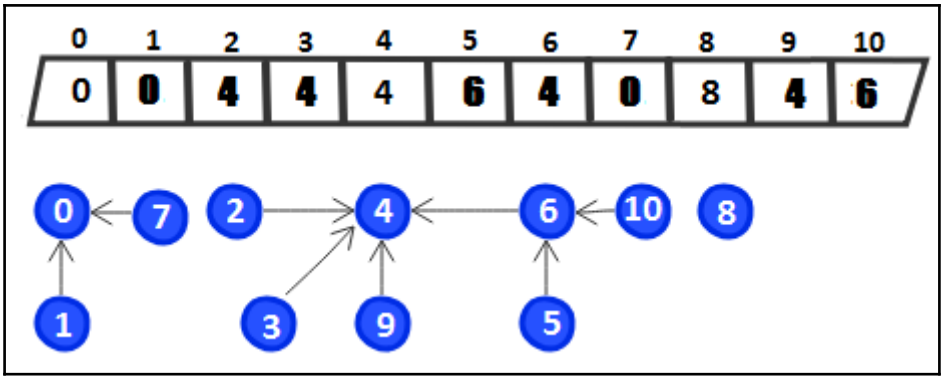

root
C
false
A
false
T
false
true
root
C
false
A
false
T R
false
true
E
false
T
false
true
root
C B
false
A
false
Y
false
T R
false
E
false
true
E
false
T
false
true
After inserting "cat"
After inserting "caret"
After inserting "bye"

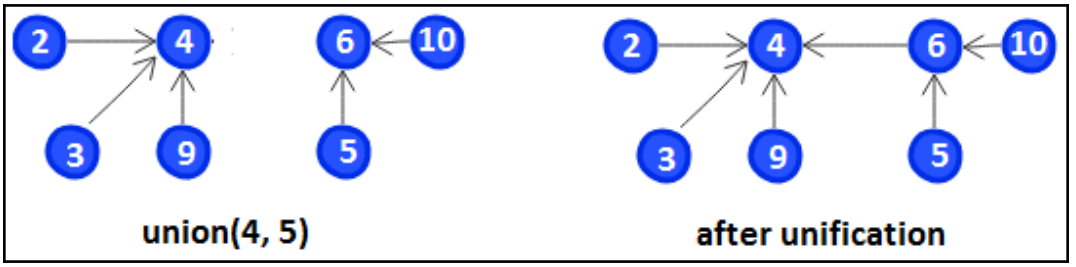

0
7
1
2
4
6
10
8
3
9
5


0 1 2 3 4 5 6 7 8 9 10
0 1 2 3 4 5 6 7 8 9 10

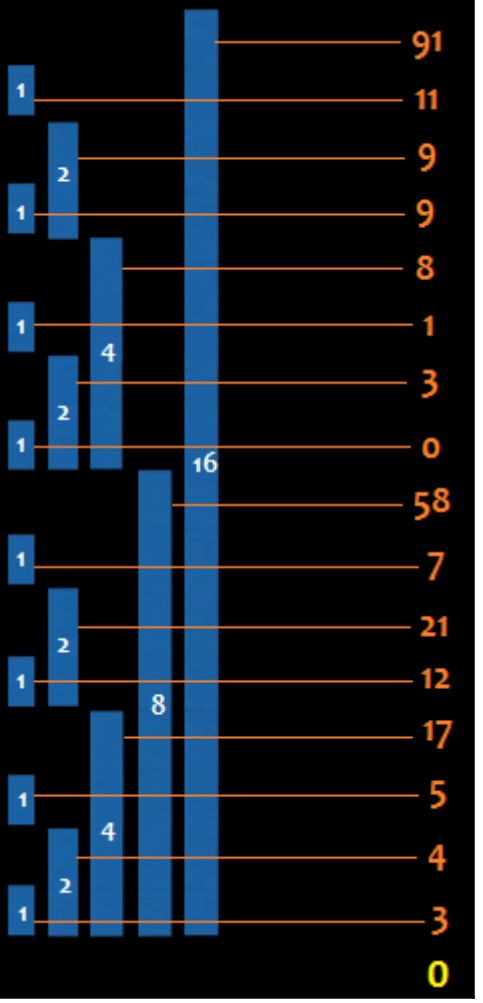

0 1 2 3 4 5 6 7 8 9 10
0 1 2 3 4 5 6 7 8 9 10
0 1 2 3 4 5 6 7 8 9 10

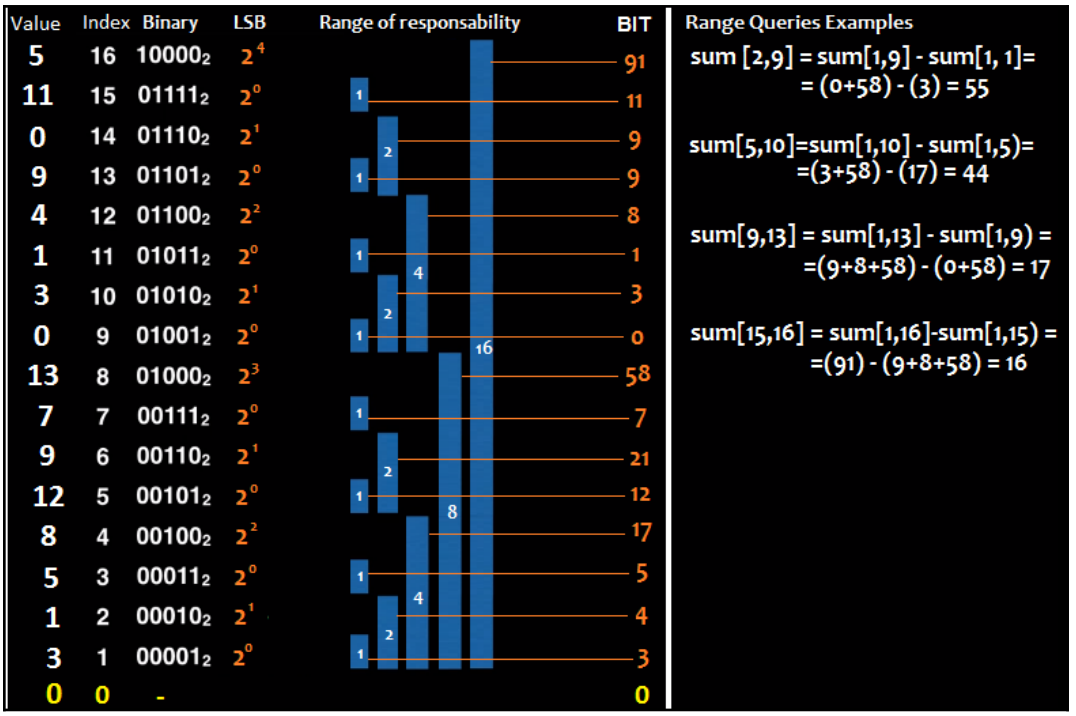

0 1 2 3 4 5 6 7 8 9 10
0 0 2 3 4 6 6 7 8 4 10
0 2 3 4 6 7 8 10
1 9 5

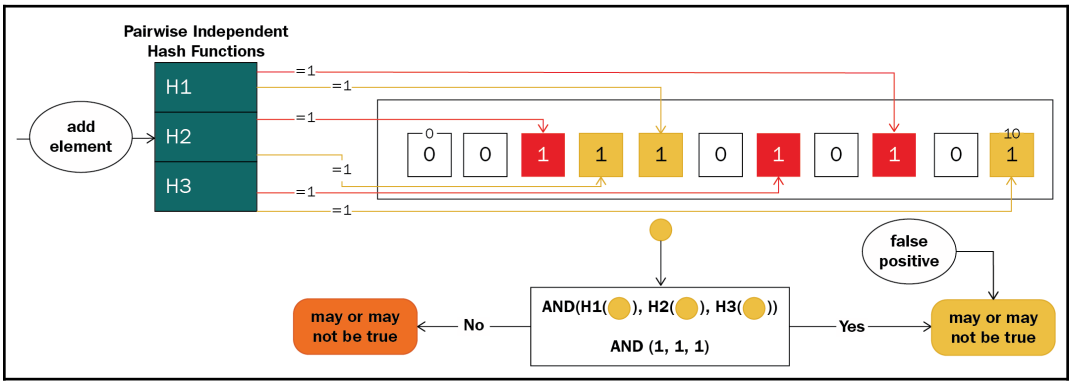

0 1 2 3 4 5 6 7 8 9 10
0 0 4 4 4 6 4 0 8 4 6
0 7 2 4 6 10 8
1 3 9 5


2 4 6 10
3 9 5
union(4, 5)
2 4 6 10
3 9 5
after unification

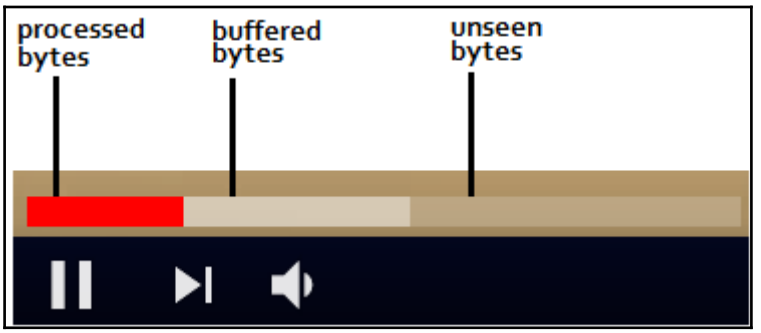
Value	Index	Binary	LSB
5	16	10000_2	2^4
11	15	01111_2	2^0
0	14	01110_2	2^1
9	13	01101_2	2^0
4	12	01100_2	2^2
1	11	01011_2	2^0
3	10	01010_2	2^1
0	9	01001_2	2^0
13	8	01000_2	2^3
7	7	00111_2	2^0
9	6	00110_2	2^1
12	5	00101_2	2^0
8	4	00100_2	2^2
5	3	00011_2	2^0
1	2	00010_2	2^1
3	1	00001_2	2^0
0	0	-	




Value Index Binary LSB
5 16 10000_2 2^4
11 15 01111_2 2^0
0 14 01110_2 2^1
9 13 01101_2 2^0
4 12 01100_2 2^2
1 11 01011_2 2^0
3 10 01010_2 2^1
0 9 01001_2 2^0
13 8 01000_2 2^3
7 7 00111_2 2^0
9 6 00110_2 2^1
12 5 00101_2 2^0
8 4 00100_2 2^2
5 3 00011_2 2^0
1 2 00010_2 2^1
3 1 00001_2 2^0
0 0 -
Range of responsability
BIT
91
11
9
9
8
1
3
0
58
7
21
12
17
5
4
3
0
Range Queries Examples
sum [2,9] = sum[1,9] - sum[1, 1]=
= (0+58) - (3) = 55
sum[5,10]=sum[1,10] - sum[1,5]=
=(3+58) - (17) = 44
sum[9,13] = sum[1,13] - sum[1,9) =
=(9+8+58) - (0+58) = 17
sum[15,16] = sum[1,16]-sum[1,15) =
=(91) - (9+8+58) = 16


Pairwise Independent
Hash Functions
add
element
H1
H2
H3
=1
0 0 1 1 1 0 1 0 1 0 1
0
10
false
positive
may or may
not be true
No
AND(H1( ), H2( ), H3( ))
AND (1, 1, 1)
Yes
may or may
not be true

Chapter 6: Java I/O Paths, Files, Buffers, Scanning, and Formatting



Raw JSON

```
{"type": "Gac", "weight": 2000}
{"type": "Hemi", "weight": 1200}
```

melons_raw.json

Array-like JSON

```
[{
        "type": "Gac",
        "weight": 2000
}, {
        "type": "Hemi",
        "weight": 1200
}]
```

melons_array.json

Map-like JSON

```
{
    "A": {
            "type": "Gac",
            "weight": 2000
    },
    "B": {
            "type": "Hemi",
            "weight": 1200
    }
}
```

melons_map.json

melon.csv

CSV

```
Gaac,2000
Hemi,1500
Cantaloupe,800
Golden Prize,2300
Crenshaw,3000
```

file1.txt

```
This is
a file for testing
mismatches between

two files!
```

file2.txt

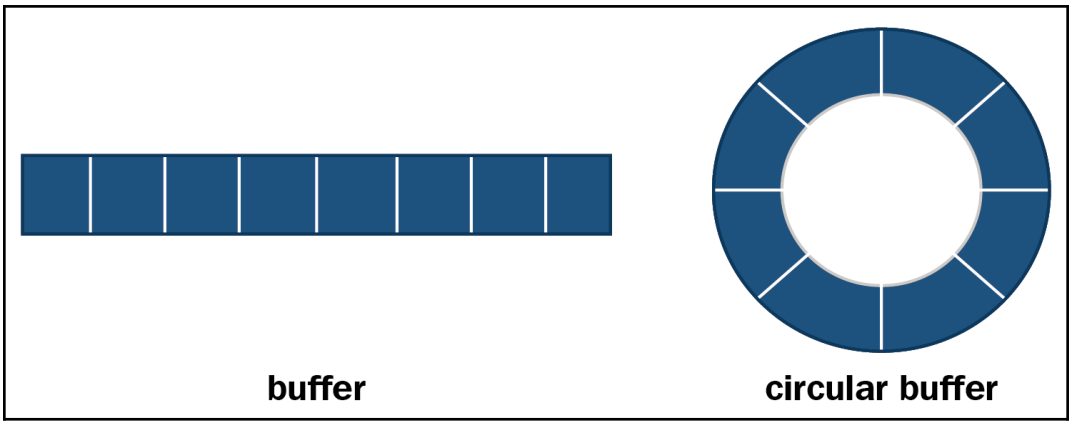
```
This is
a file for testing
mismatches between

two files!
```

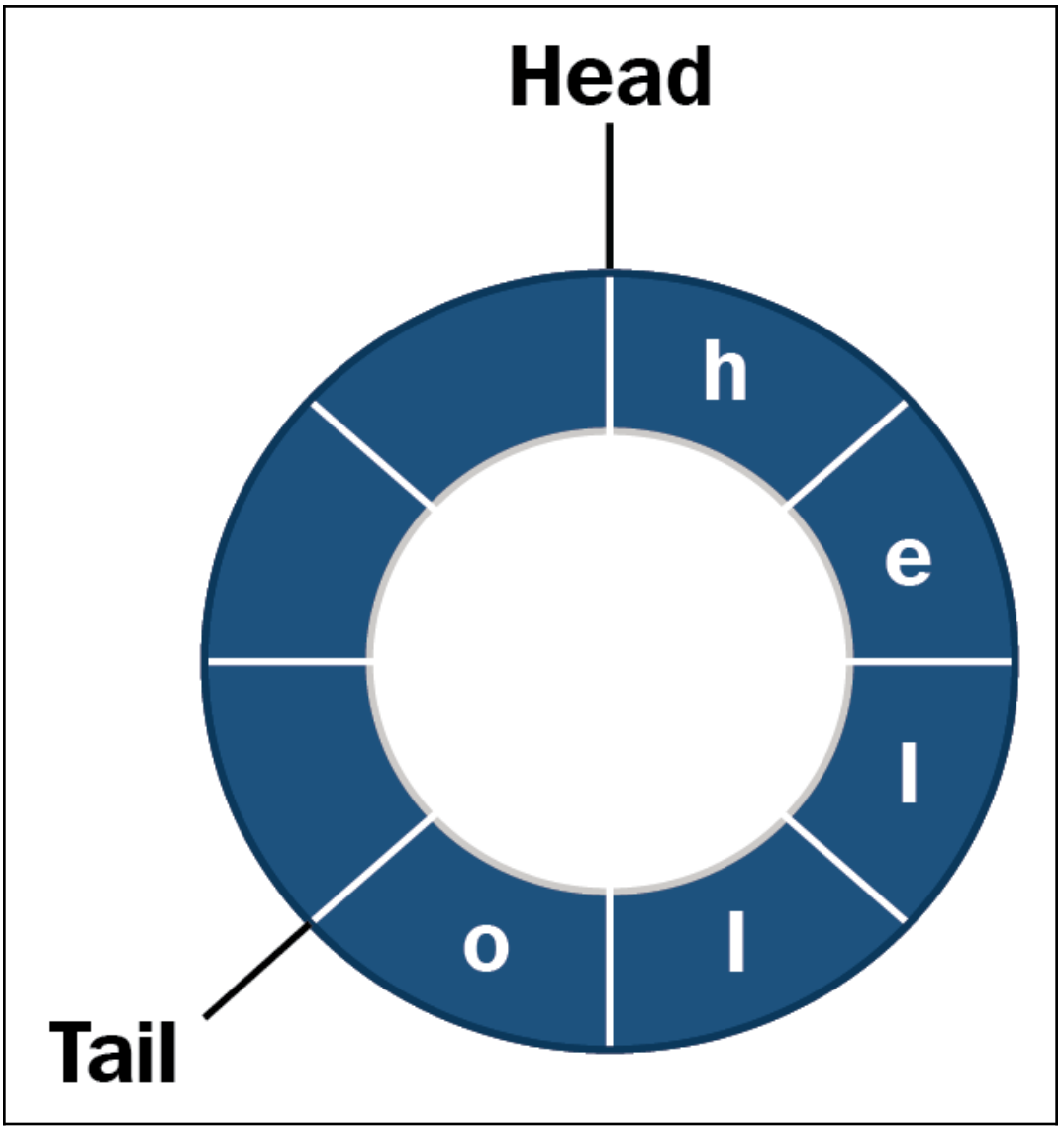
file3.txt

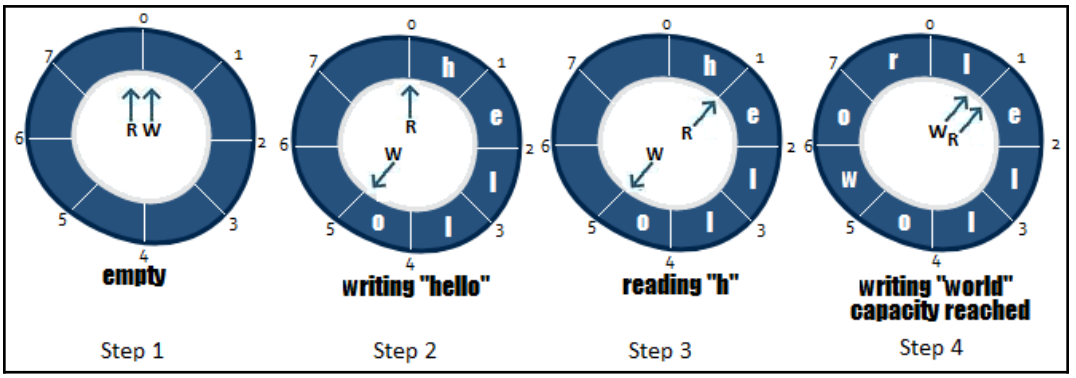
```
This is
a file for testing
mismatches between


two files!
```

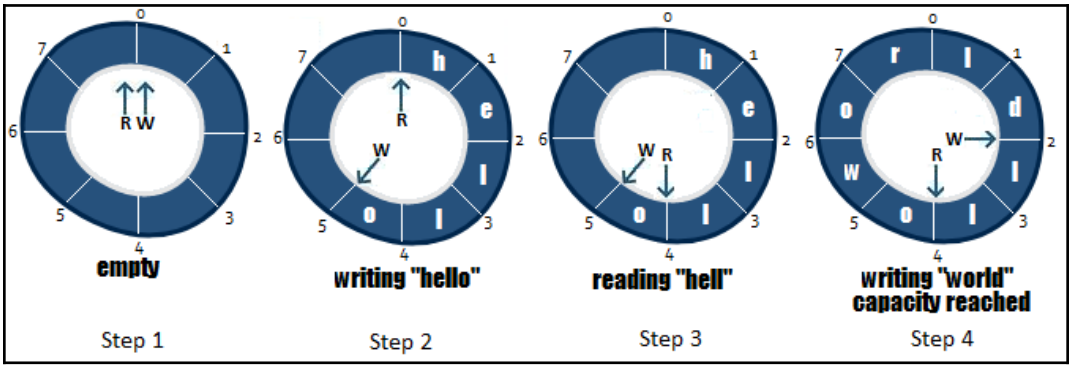
file4.txt

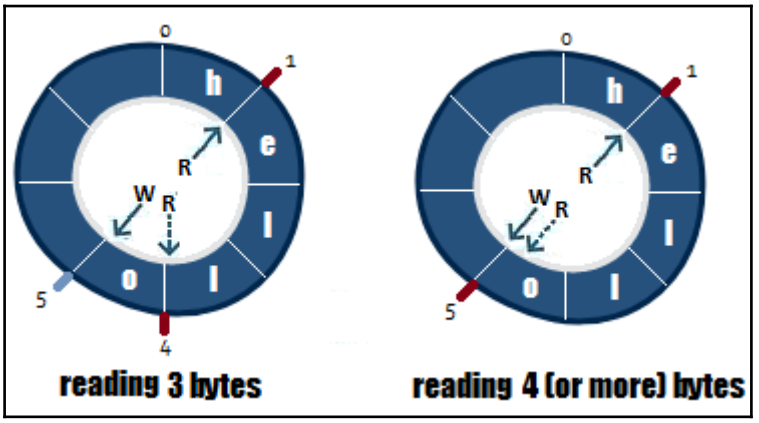
```
This is
a file for testing
mismatches between

two files.
```

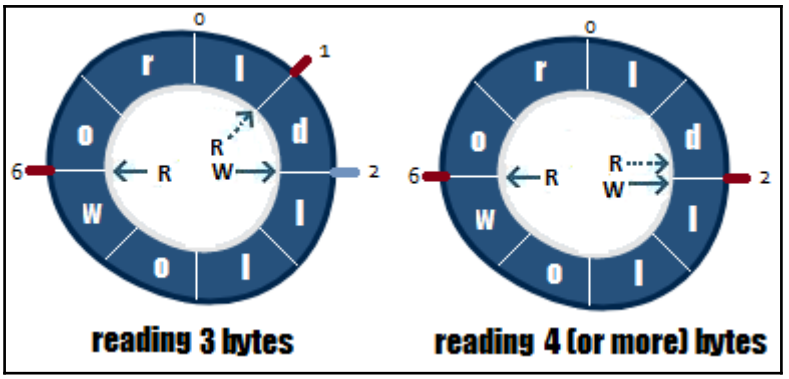

buffer
circular buffer

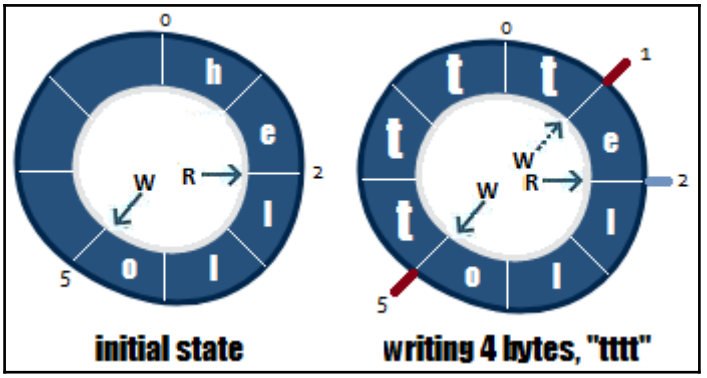

Head
h
e
l
l
o
Tail

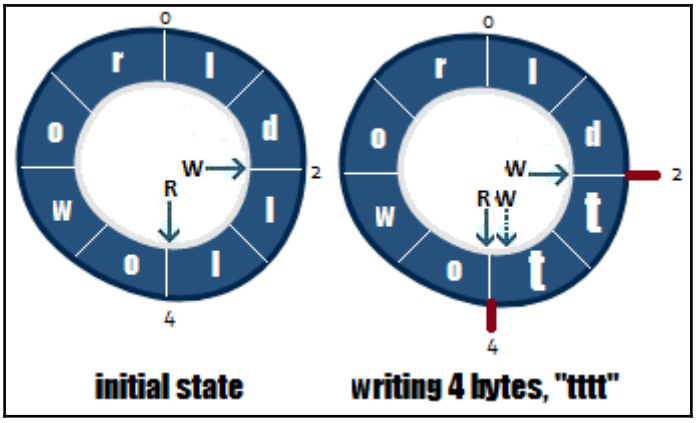

empty
writing "hello"
reading "h"
writing "world"
capacity reached
Step 1
Step 2
Step 3
Step 4


empty
writing "hello"
reading "hell"
writing "world"
capacity reached
Step 1
Step 2
Step 3
Step 4


reading 3 bytes
reading 4 (or more) bytes


reading 3 bytes
reading 4 (or more) bytes


initial state
writing 4 bytes, "tttt"


initial state
writing 4 bytes, "tttt"

Obtained

```
| 78910 | 0.9276730641526881 |
| 83222 | 0.28423903775300785 |
| 5593 | 0.866538798997145 |
| 57329 | 0.9145723363689985 |
| 61443 | 0.41527451214386724 |
| 9043 | 0.8442927124583571 |
| 474 | 0.9159122616950742 |
| 45763 | 0.04448867226365116 |
| 26671 | 0.4648636732351614 |
| 24096 | 0.12870733626570974 |
```

Wanted

```
|  78910 | 0.928 |
|  83222 | 0.284 |
|   5593 | 0.867 |
|  57329 | 0.915 |
|  61443 | 0.415 |
|   9043 | 0.844 |
|    474 | 0.916 |
|  45763 | 0.044 |
|  26671 | 0.465 |
|  24096 | 0.129 |
```

```
78,910 bytes
83,222 bytes
 5,593 bytes
57,329 bytes
61,443 bytes
 9,043 bytes
   474 bytes
45,763 bytes
26,671 bytes
24,096 bytes
```

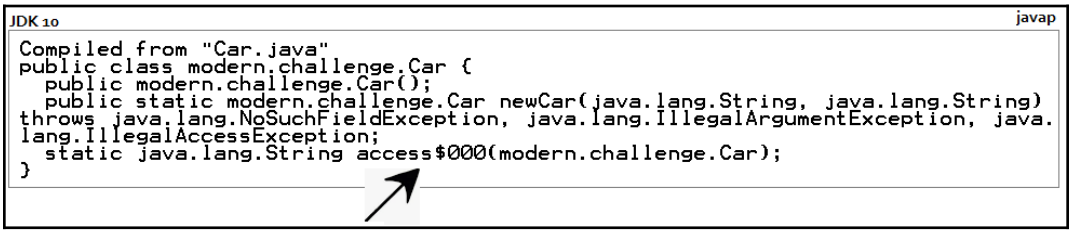
doubles.txt

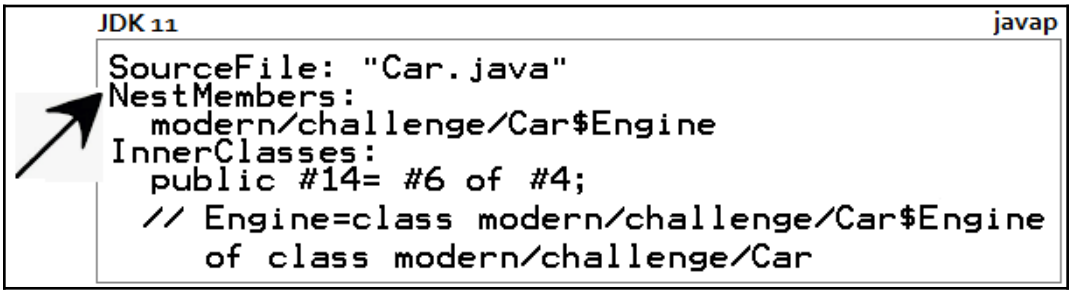
```
23.4556 1.23 4.55 2.33
5.663 956.34343 23.2333
0.3434 0.788
```

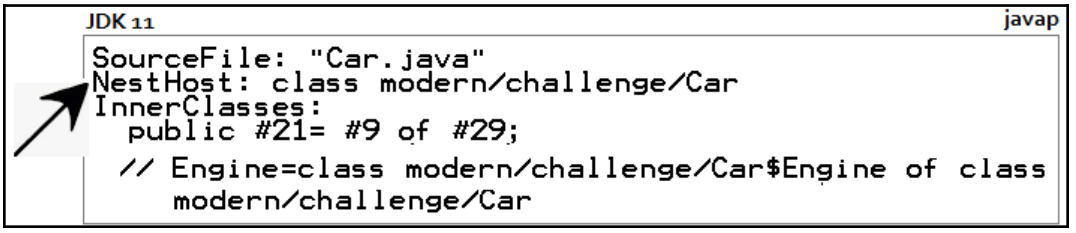
people.txt

```
Matt,Kyle,23,San Franciso;
Darel,Der,50,New York;Sandra,Hui,40,Dallas;
Leonard,Vurt,43,Bucharest;Mark,Seil,19,Texas;Ulm,Bar,43,Kansas
```

Chapter 7: Java Reflection Classes, Interfaces, Constructors, Methods, and Fields








```
@Packt
package modern.challenge;
```

```
@Fruit(name = "melon", value = "delicious")
public class Melon extends @Family Cucurbitaceae
        implements @ByWeight Comparable {
```

```
@Ripe(true)
public void eat() throws @Runtime IllegalStateException {
}
```

```
@Ripe(true)
public void eat() throws @Runtime IllegalStateException {
}
```

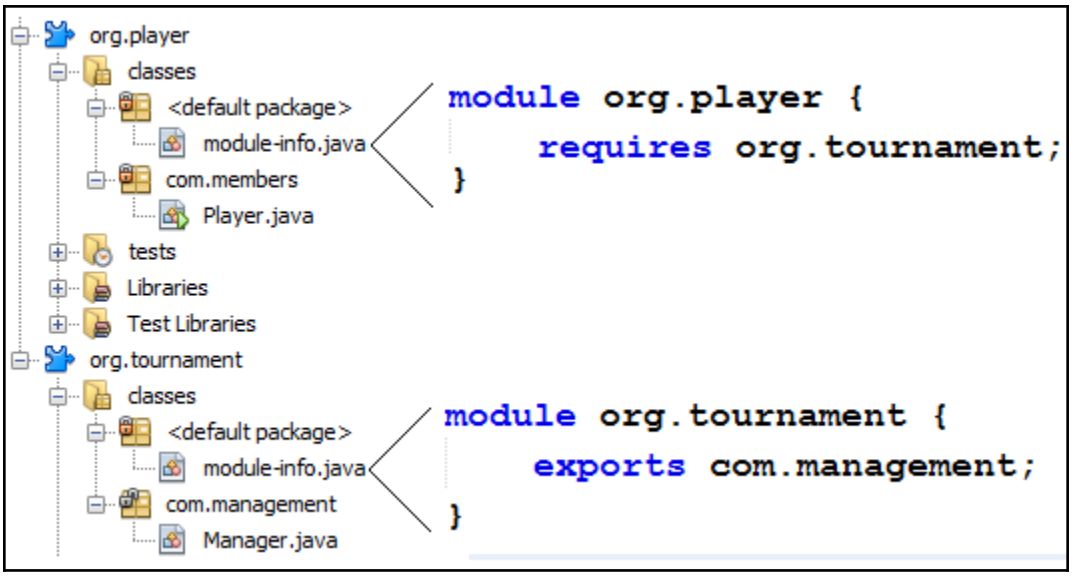
```
public @Shape("oval") List<Seed> seeds() {
    return Collections.emptyList();
}
```

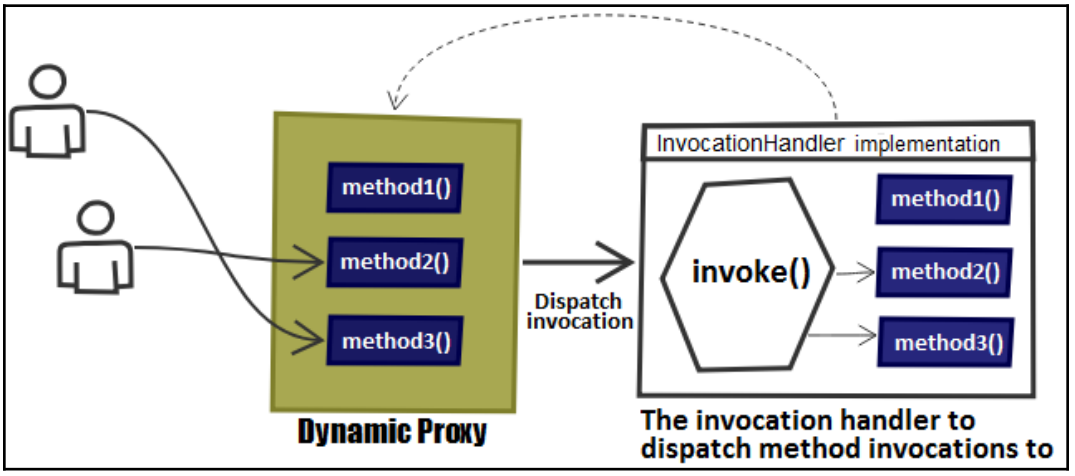
```
public void slice(@Ripe(true) @Shape("square") int noOfSlices) {
}
```

```
@Unit
private final int weight;
```

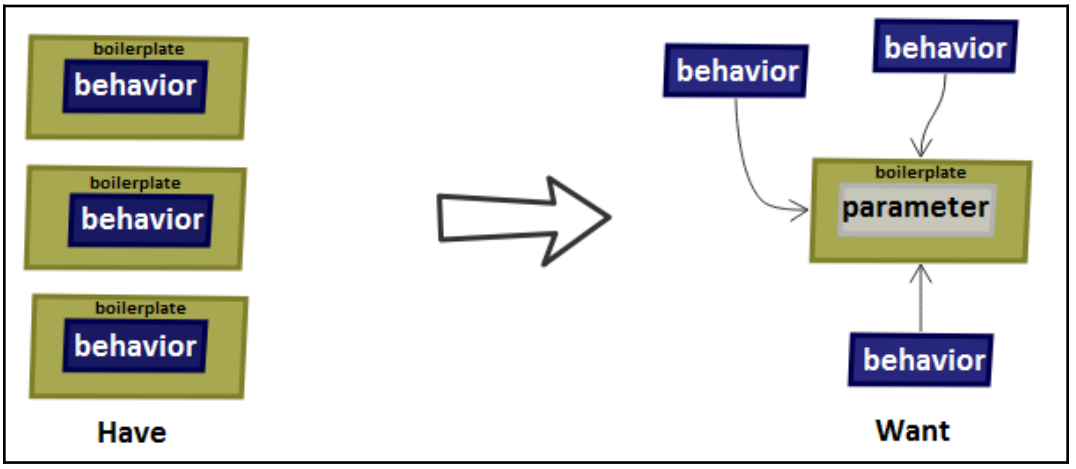
```
@Fruit(name = "melon", value = "delicious")
public class Melon extends @Family Cucurbitaceae
        implements @ByWeight Comparable {
```

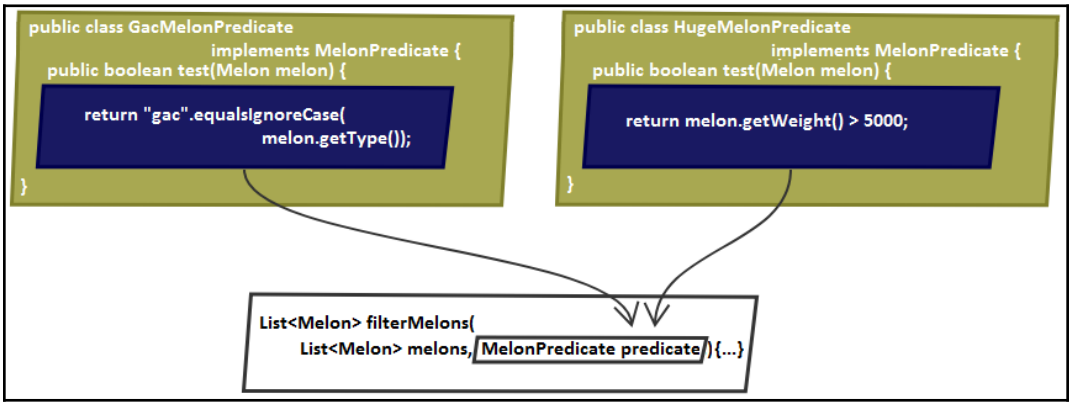
```
@Fruit(name = "melon", value = "delicious")
public class Melon extends @Family Cucurbitaceae
        implements @ByWeight Comparable {
```


org.player
classes
<default package>
module-info.java
com.members
Player.java
tests
Libraries
Test Libraries
org.tournament
classes
<default package>
module-info.java
com.management
Manager.java
module org.player {
requires org.tournament;
}
module org.tournament {
exports com.management;
}

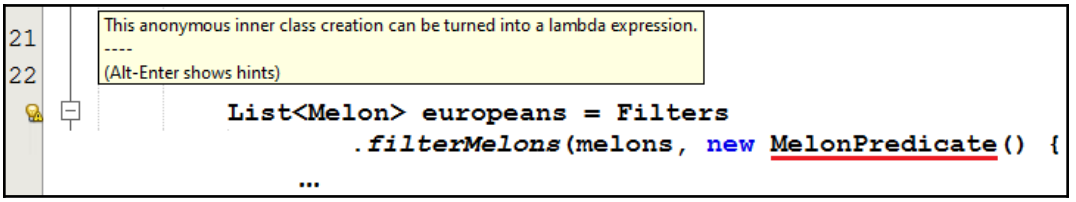

method1()
method2()
method3()
Dynamic Proxy
Dispatch invocation
InvocationHandler implementation
invoke()
method1()
method2()
method3()
The invocation handler to dispatch method invocations to

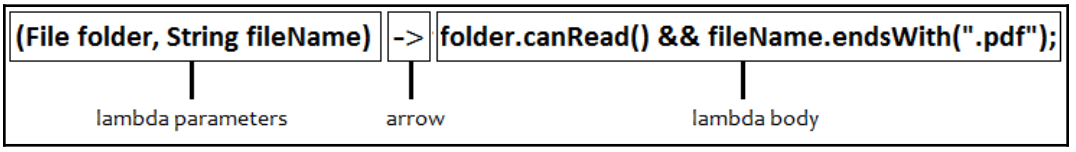
Chapter 8: Functional Style Programming - Fundamentals and Design Patterns

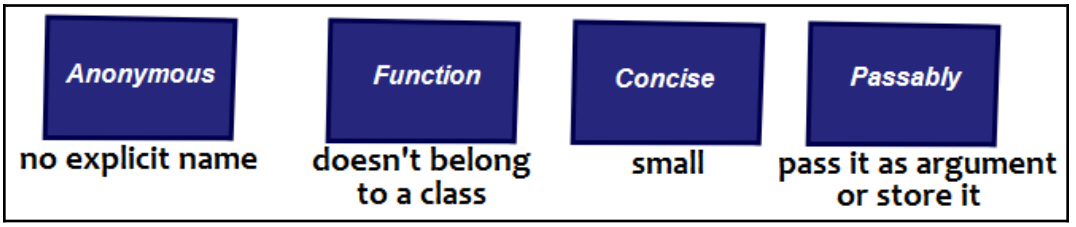




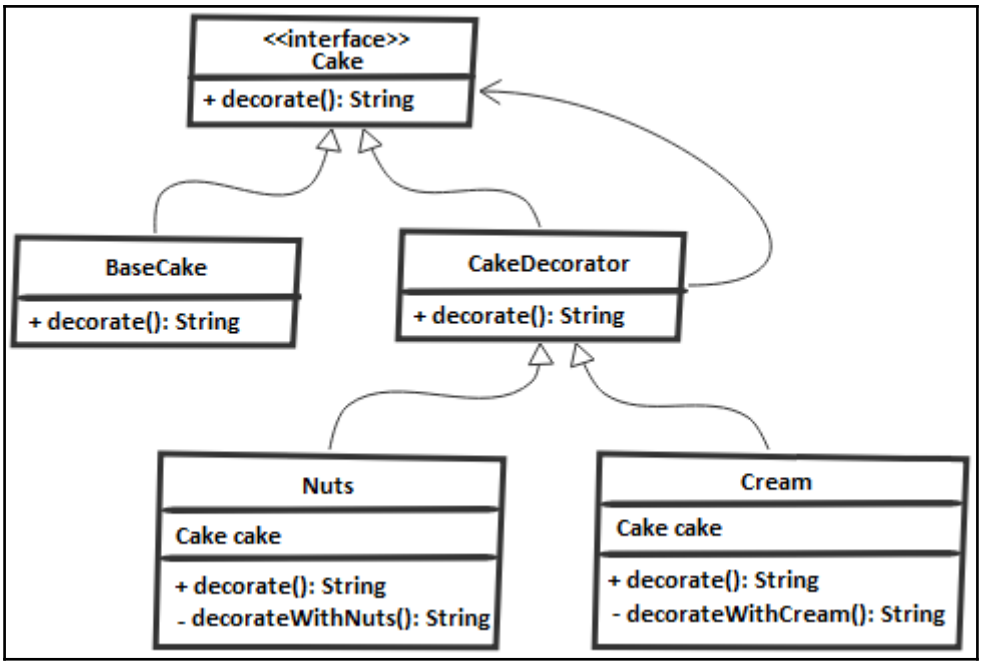
```
List<Melon> europeans = Filters.filterMelons(
                                melons, new MelonPredicate() {
    @Override
     public boolean test(Melon melon) {

         return "europe".equalsIgnoreCase(melon.getOrigin());

     }
});
```

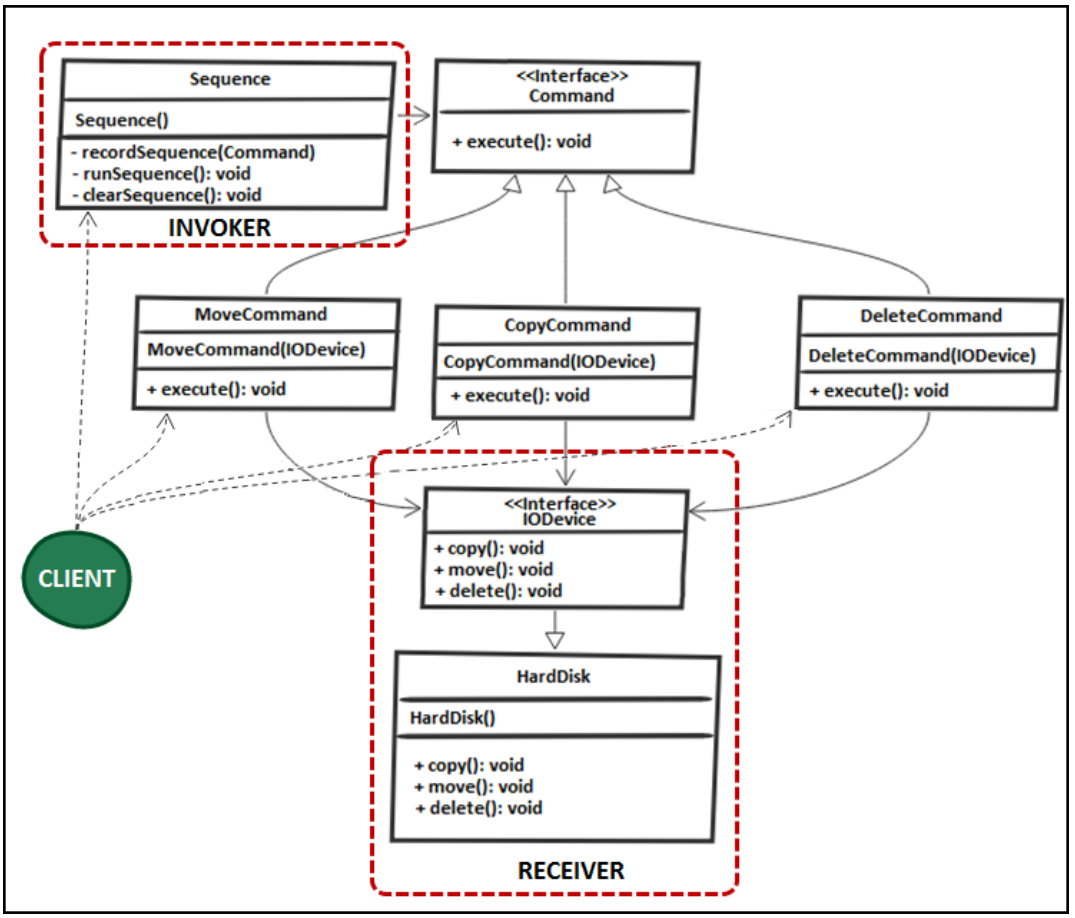






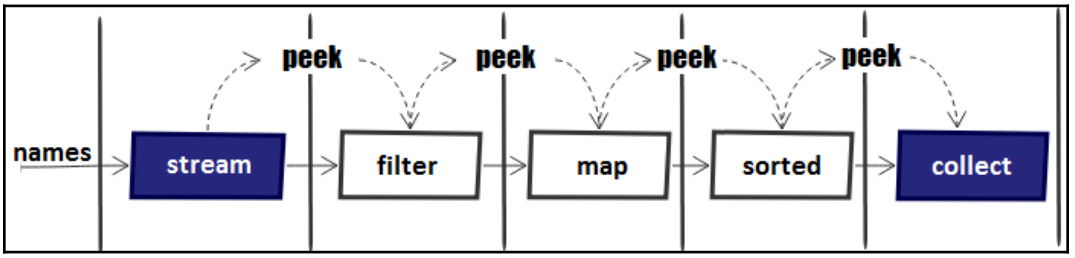

```
try (Scanner scanner = new Scanner(
                Path.of("doubles.txt"), StandardCharsets.UTF_8)) {
}
```


<<interface>>
Cake
+ decorate(): String
BaseCake
+ decorate(): String
CakeDecorator
+ decorate(): String
Nuts
Cake cake
+ decorate(): String
- decorateWithNuts(): String
Cream
Cake cake
+ decorate(): String
- decorateWithCream(): String


Sequence
Sequence()
- recordSequence(Command)
- runSequence(): void
- clearSequence(): void
INVOKER
<<Interface>>
Command
+ execute(): void
MoveCommand
MoveCommand(IODevice)
+ execute(): void
CopyCommand
CopyCommand(IODevice)
+ execute(): void
DeleteCommand
DeleteCommand(IODevice)
+ execute(): void
CLIENT
<<Interface>>
IODevice
+ copy(): void
+ move(): void
+ delete(): void
HardDisk
HardDisk()
+ copy(): void
+ move(): void
+ delete(): void
RECEIVER

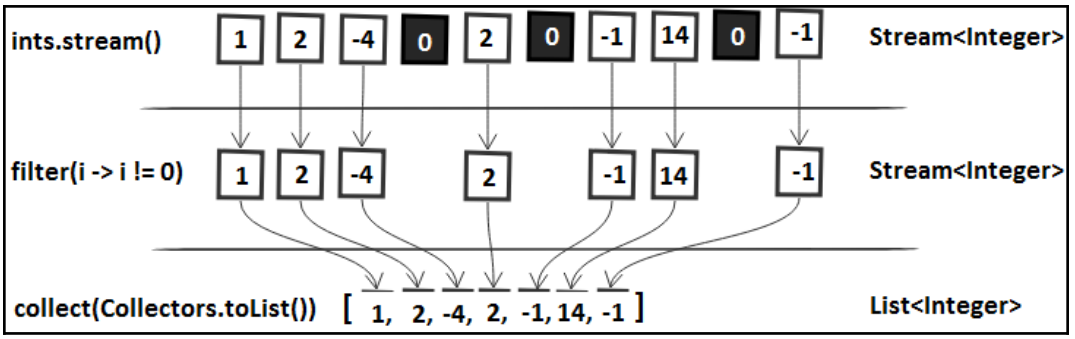
Chapter 9: Functional Style Programming - a Deep Dive

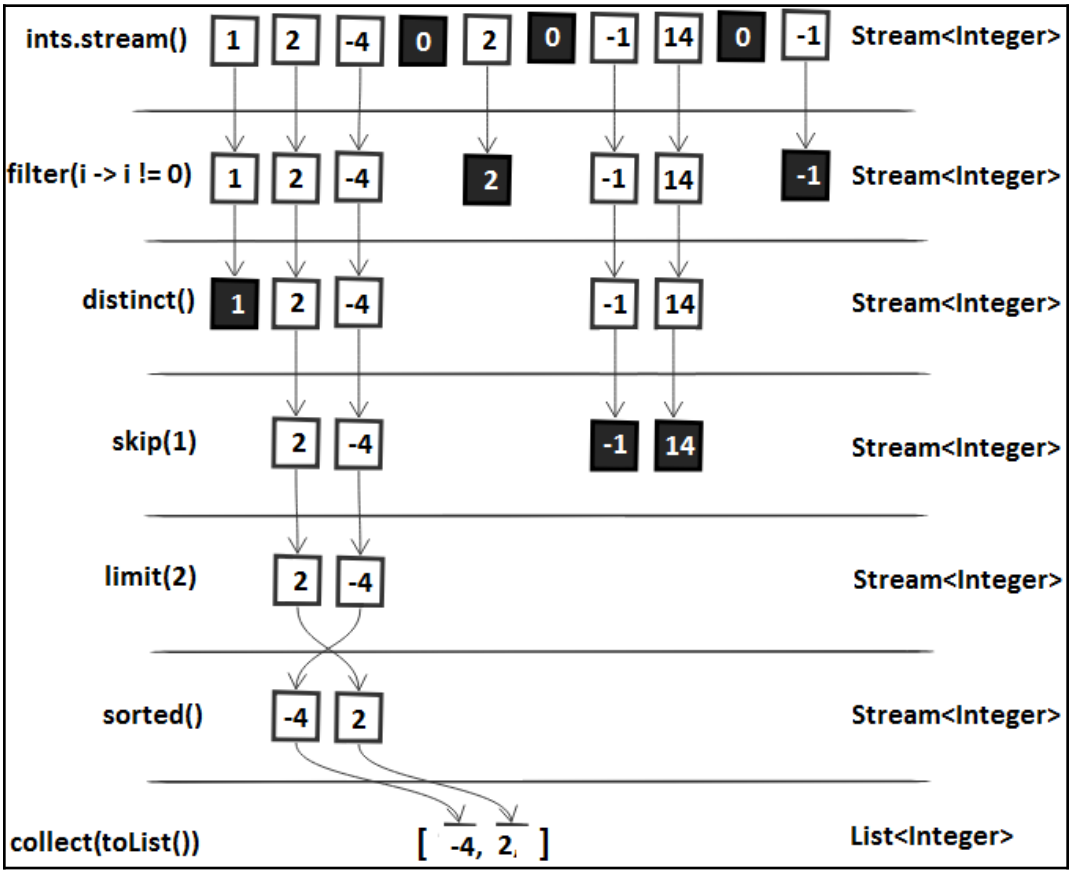
```
Exception in thread "main" java.lang.NullPointerException
        at modern.challenge.Main.lambda$main$5(Main.java:28)
        at java.base/java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:195)
        at java.base/java.util.Spliterators$ArraySpliterator.forEachRemaining(Spliterators.java:948)
        at java.base/java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:484)
        at java.base/java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:474)
        at java.base/java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.java:913)
        at java.base/java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)
        at java.base/java.util.stream.ReferencePipeline.collect(ReferencePipeline.java:578)
        at modern.challenge.Main.main(Main.java:29)
```

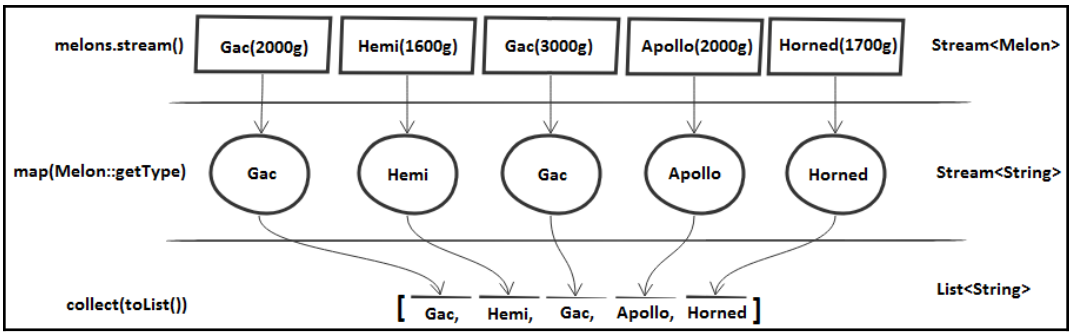


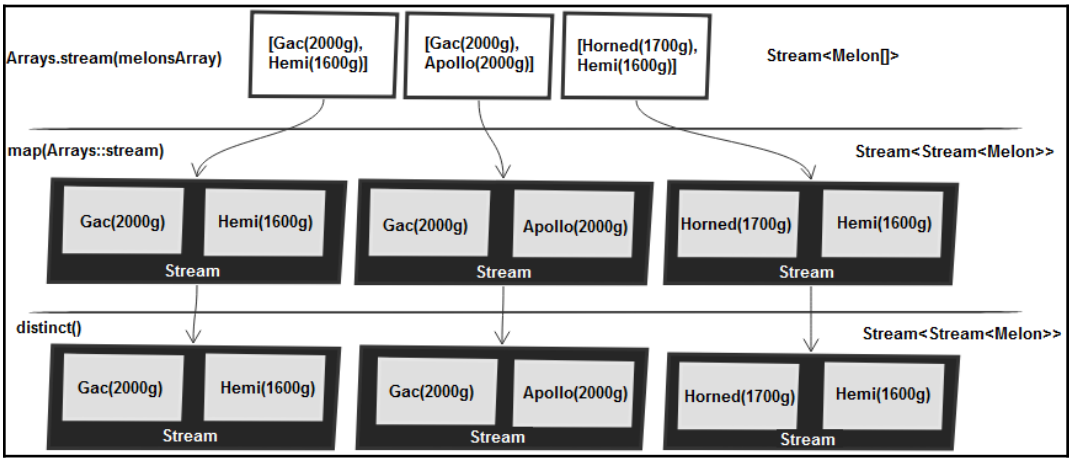
```
After:
        stream(): anna
        stream(): bob
        stream(): christian
        filter(): christian
        map(): CHRISTIAN
        stream(): carmen
        filter(): carmen
        map(): CARMEN
        stream(): rick
        stream(): carla
        filter(): carla
        map(): CARLA
        sorted(): CARLA
        sorted(): CARMEN
        sorted(): CHRISTIAN
```

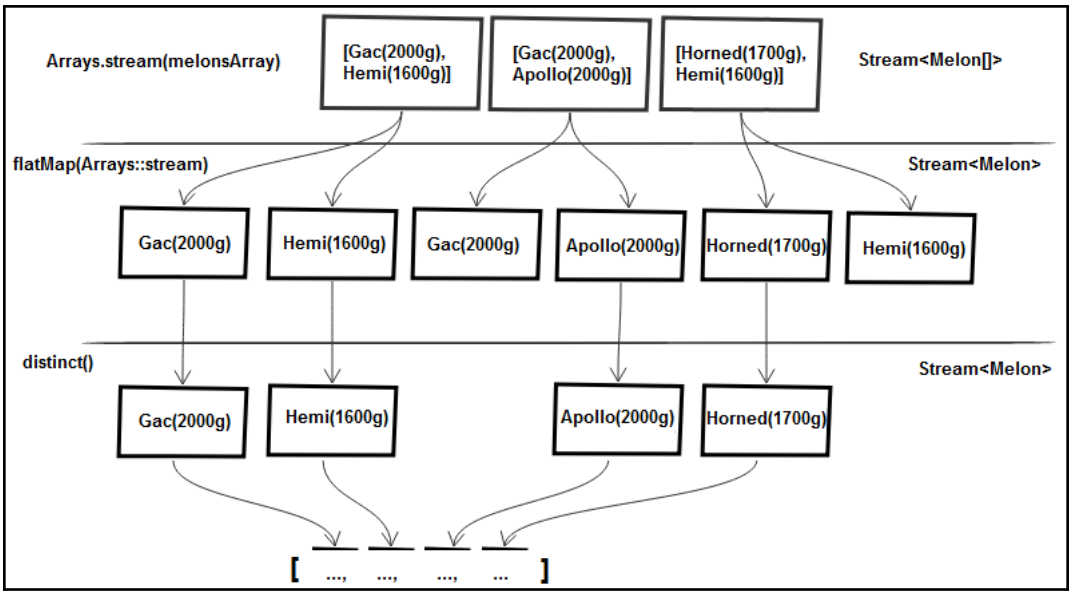
```
After:
        stream(): anna
        stream(): bob
        stream(): christian
        filter(): christian
        map(): CHRISTIAN
        stream(): null
Exception in thread "main" java.lang.NullPointerException
        at modern.challenge.Main.lambda$main$1(Main.java:16)
        ...
```

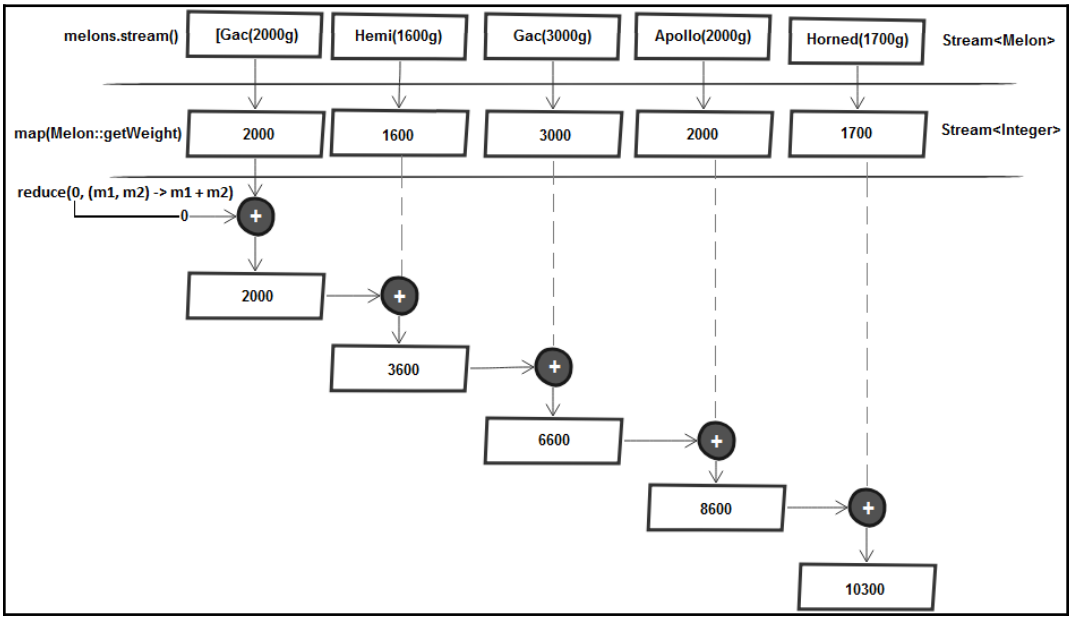

ints.stream()
1 2 -4 0 2 0 -1 14 0 -1
Stream<Integer>
filter(i -> i != 0)
1 2 -4 2 -1 14 -1
Stream<Integer>
collect(Collectors.toList())
[1, 2, -4, 2, -1, 14, -1]
List<Integer>


ints.stream()
1 2 -4 0 2 0 -1 14 0 -1
Stream<Integer>
filter(i -> i != 0)
1 2 -4 2 -1 14 -1
Stream<Integer>
distinct()
1 2 -4 -1 14
Stream<Integer>
skip(1)
2 -4 -1 14
Stream<Integer>
limit(2)
2 -4
Stream<Integer>
sorted()
-4 2
Stream<Integer>
collect(toList())
[-4, 2]
List<Integer>

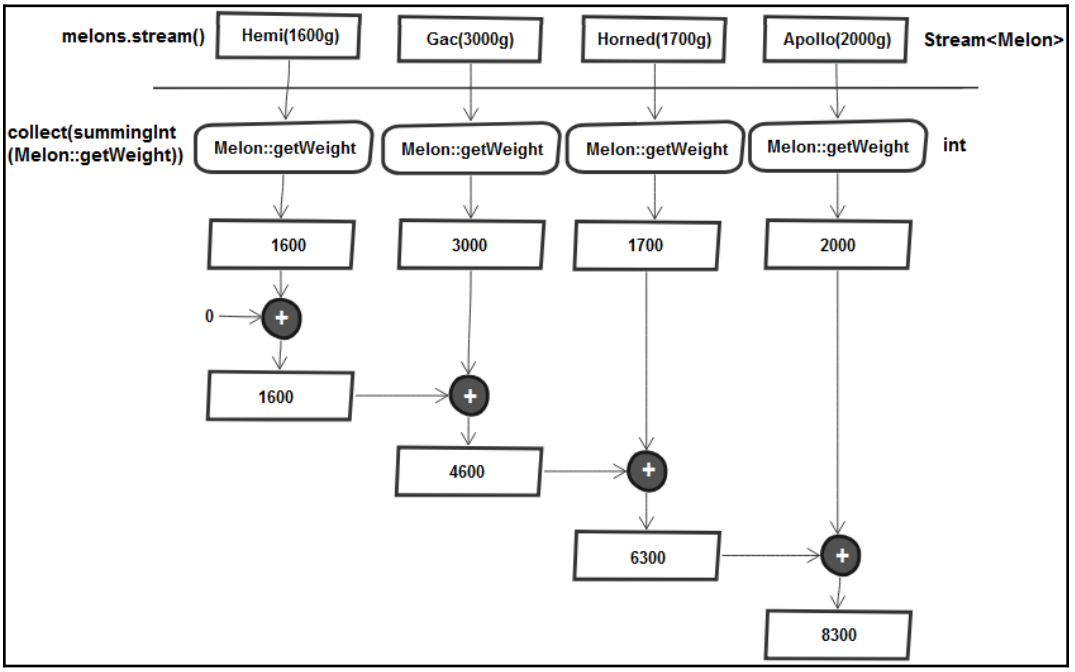

melons.stream()
Gac(2000g)
Hemi(1600g)
Gac(3000g)
Apollo(2000g)
Horned(1700g)
Stream<Melon>
map(Melon::getType)
Gac
Hemi
Gac
Apollo
Horned
Stream<String>
collect(toList())
[Gac, Hemi, Gac, Apollo, Horned]
List<String>

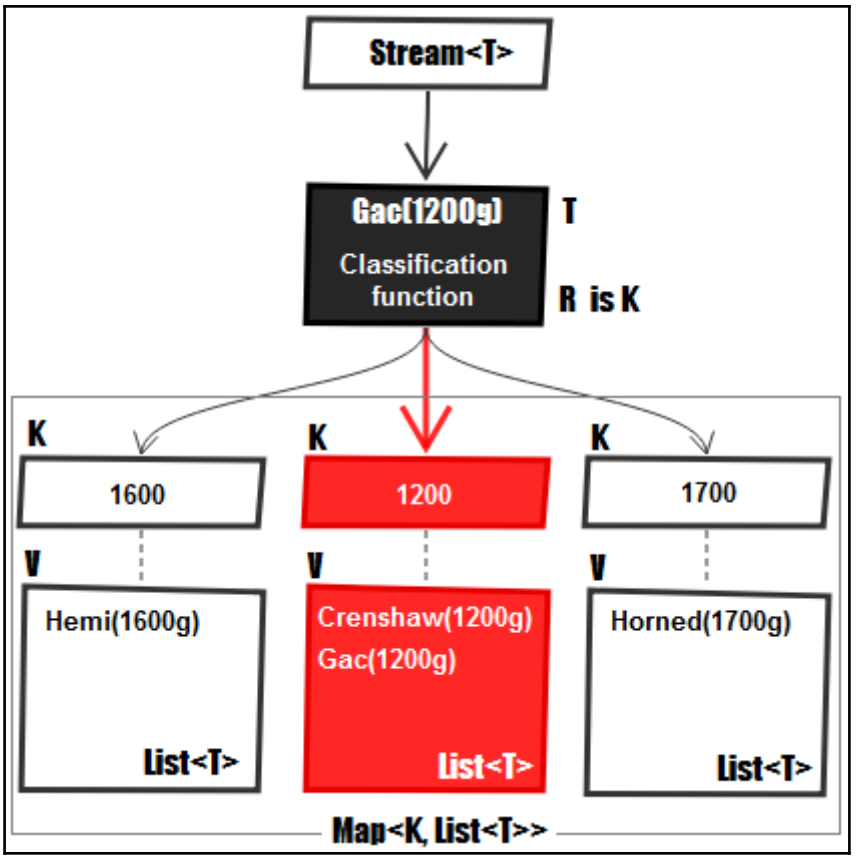

Arrays.stream(melonsArray)
[Gac(2000g), Hemi(1600g)]
[Gac(2000g), Apollo(2000g)]
[Horned(1700g), Hemi(1600g)]
Stream<Melon[]>
map(Arrays::stream)
Stream<Stream<Melon>>
Gac(2000g)
Hemi(1600g)
Stream
Gac(2000g)
Apollo(2000g)
Stream
Horned(1700g)
Hemi(1600g)
Stream
distinct()
Stream<Stream<Melon>>
Gac(2000g)
Hemi(1600g)
Stream
Gac(2000g)
Apollo(2000g)
Stream
Horned(1700g)
Hemi(1600g)
Stream


Arrays.stream(melonsArray)
[Gac(2000g), Hemi(1600g)]
[Gac(2000g), Apollo(2000g)]
[Horned(1700g), Hemi(1600g)]
Stream<Melon[]>
flatMap(Arrays::stream)
Stream<Melon>
Gac(2000g)
Hemi(1600g)
Gac(2000g)
Apollo(2000g)
Horned(1700g)
Hemi(1600g)
distinct()
Stream<Melon>
Gac(2000g)
Hemi(1600g)
Apollo(2000g)
Horned(1700g)
[..., ..., ..., ...]

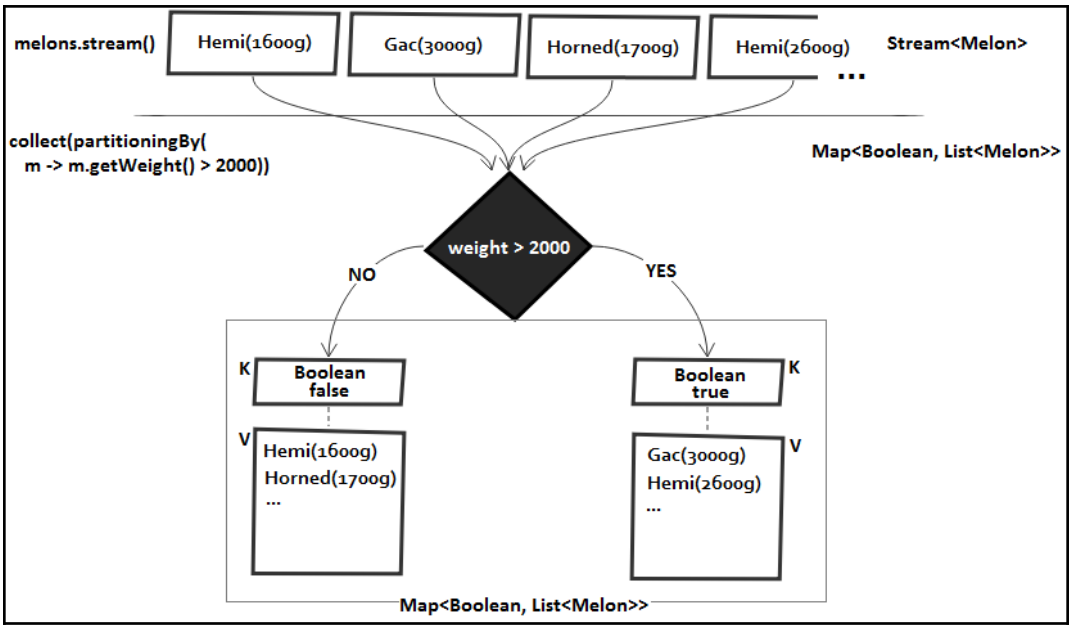

melons.stream()
[Gac(2000g)
Hemi(1600g)
Gac(3000g)
Apollo(2000g)
Horned(1700g)
Stream<Melon>
map(Melon::getWeight)
2000
1600
3000
2000
1700
Stream<Integer>
reduce(0, (m1, m2) -> m1 + m2)
0
2000
3600
6600
8600
10300

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} = \left(\frac{\sum_{i=1}^{n} x_i^{-1}}{n} \right)^{-1}$$

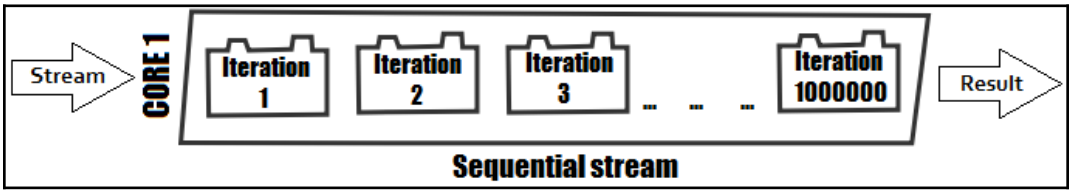


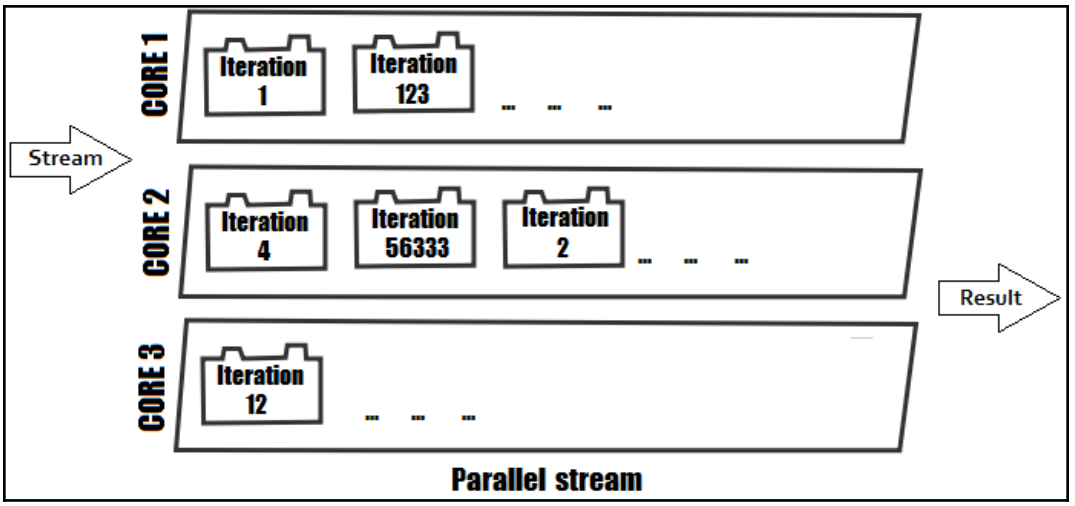


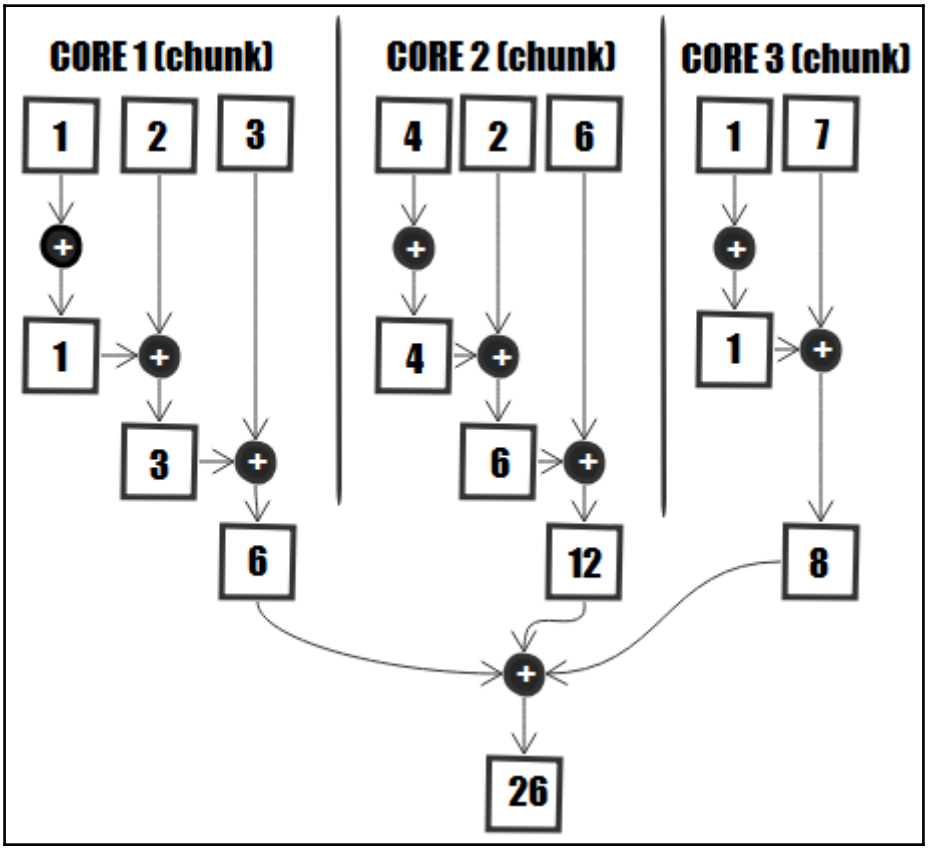
sugar / weight	LOW	MEDIUM	HIGH	UNKNOWN
2600		Apollo Cantaloupe	Hemi	
3600		Cantaloupe		
1200	Gac		Crenshaw Horned	
1600				Hemi
3000	Gac			

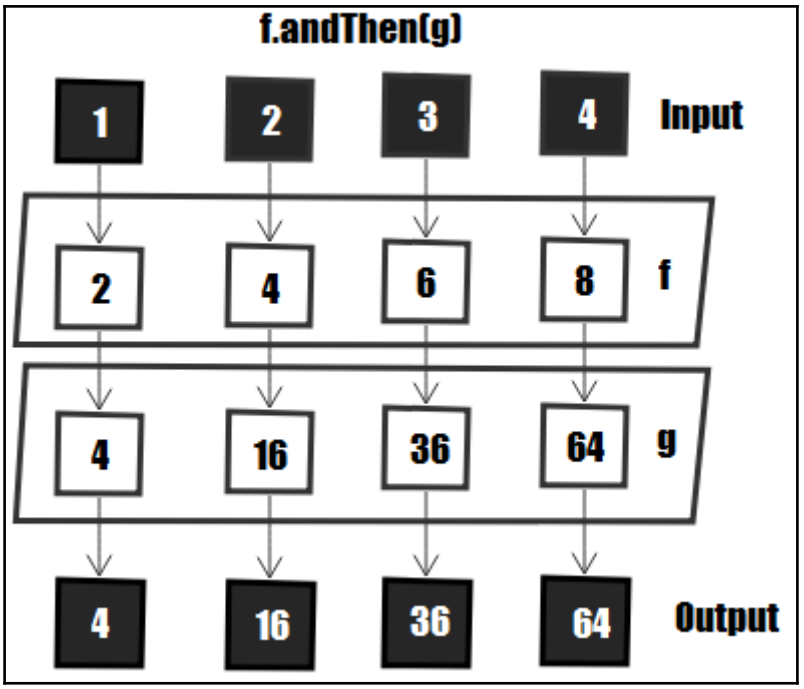

melons.stream()
Hemi(1600g)
Gac(3000g)
Horned(1700g)
Hemi(2600g)
...
Stream<Melon>
collect(partitioningBy(
m -> m.getWeight() > 2000))
Map<Boolean, List<Melon>>
weight > 2000
NO
YES
K
Boolean false
V
Hemi(1600g)
Horned(1700g)
...
Boolean true
K
Gac(3000g)
Hemi(2600g)
...
V
Map<Boolean, List<Melon>>

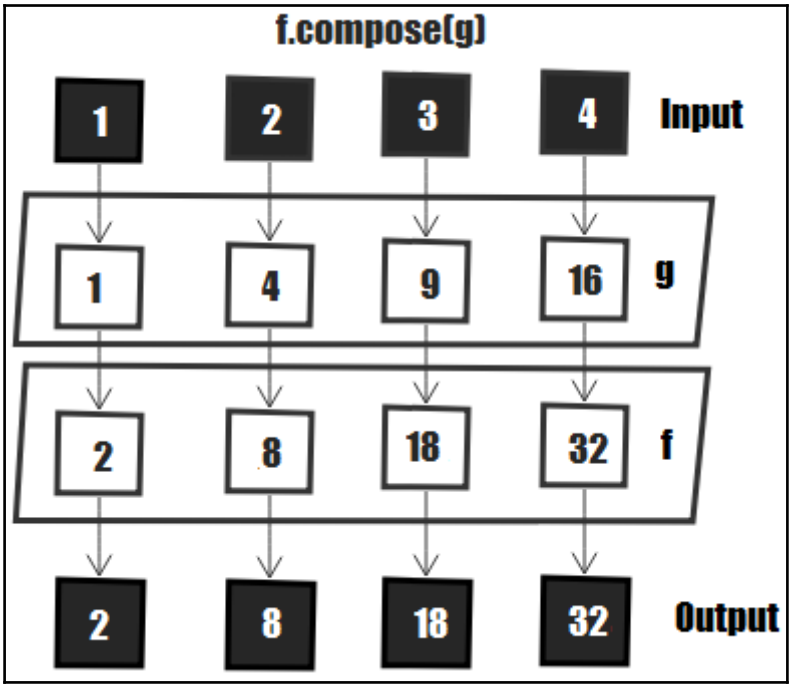
Collector 2
Collector 1
Collector result
aggregator


Stream
CORE 1
Iteration 1
Iteration 2
Iteration 3
...
Iteration 1000000
Result
Sequential stream

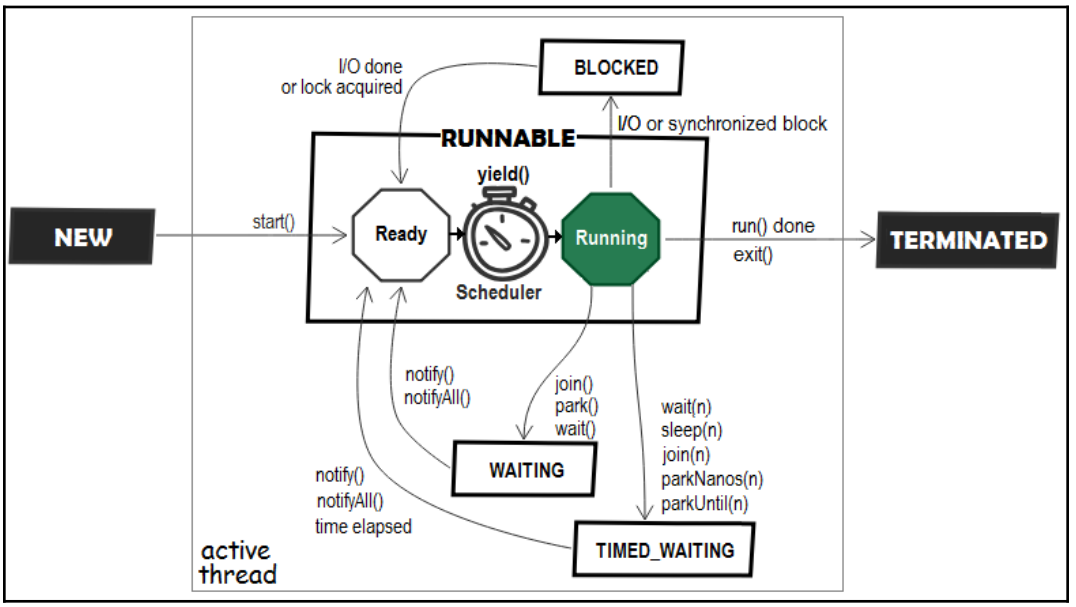

CORE 1
Iteration 1
Iteration 123
Stream
CORE 2
Iteration 4
Iteration 56333
Iteration 2
Result
CORE 3
Iteration 12
Parallel stream

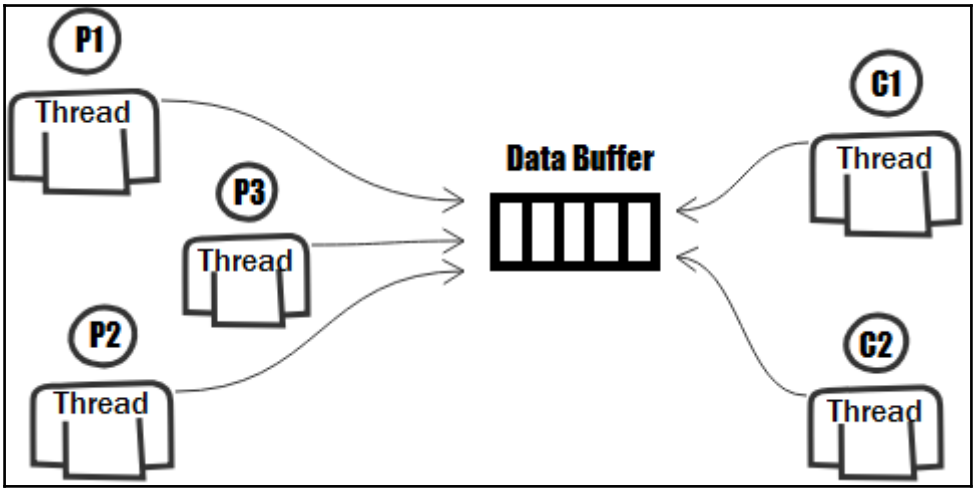

CORE 1 (chunk)
CORE 2 (chunk)
CORE 3 (chunk)
1
2
3
4
2
6
1
7
1
4
3
6
1
6
12
8
26

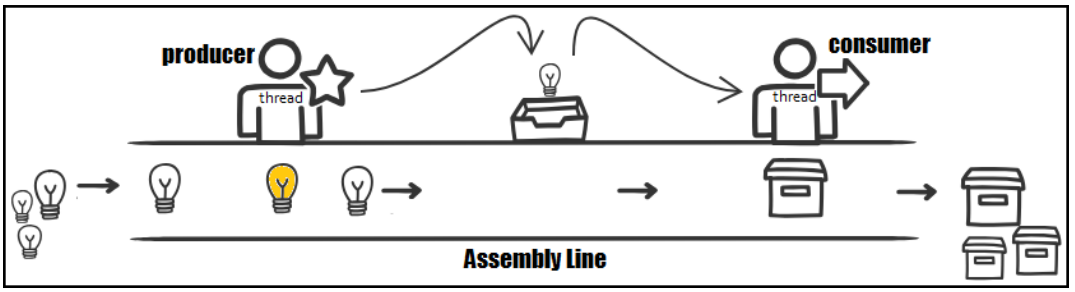

f.andThen(g)
1
2
3
4
Input
2
4
6
8
f
4
16
36
64
g
4
16
36
64
Output

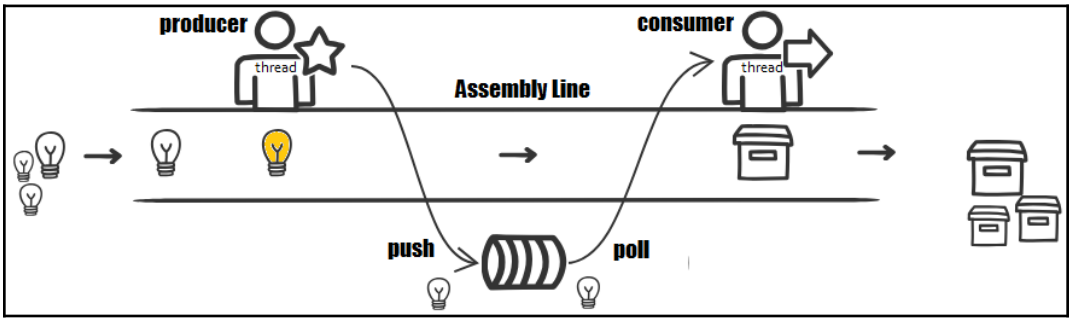

f.compose(g)
1
2
3
4
Input
1
4
9
16
g
2
8
18
32
f
2
8
18
32
Output

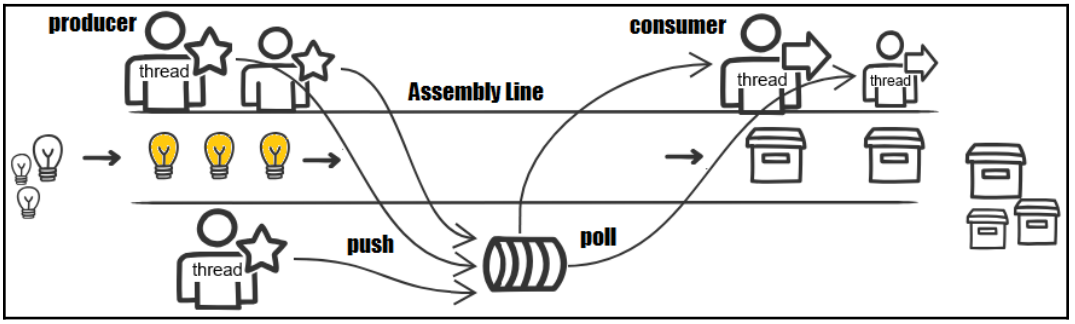
Chapter 10: Concurrency - Thread Pools, Callables, and Synchronizers

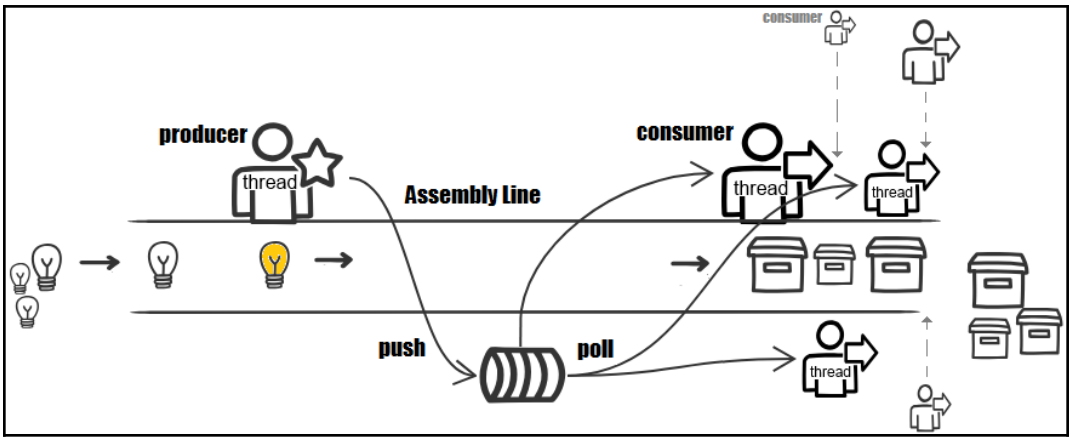


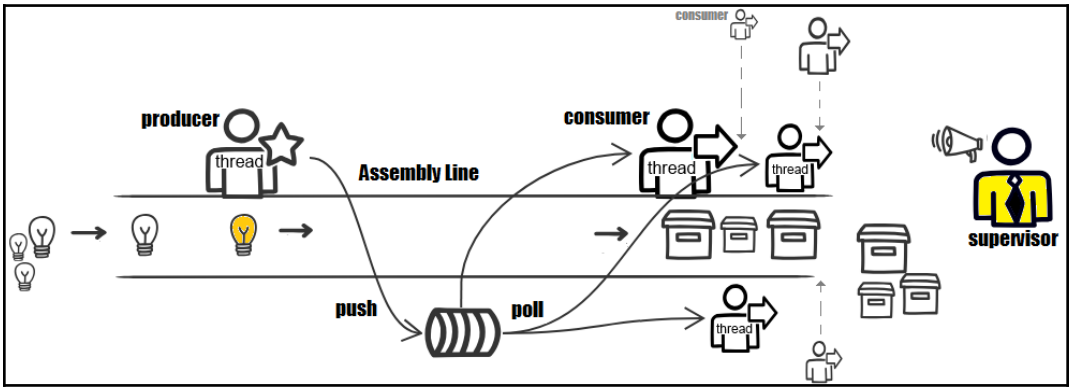


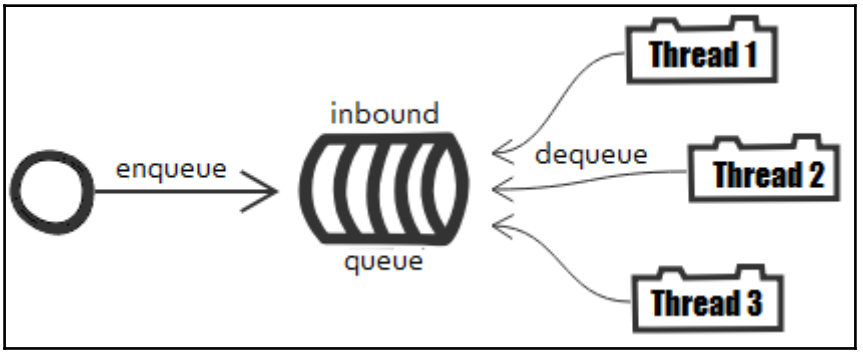

producer
thread
consumer
thread
Assembly Line

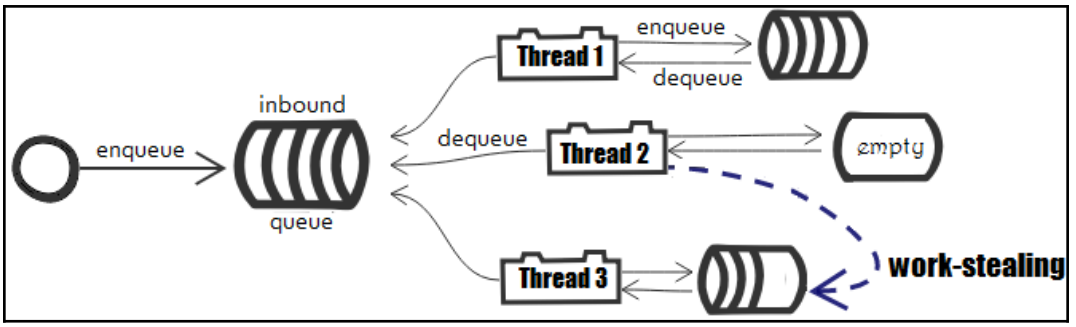

producer
thread
consumer
thread
Assembly Line
push
poll

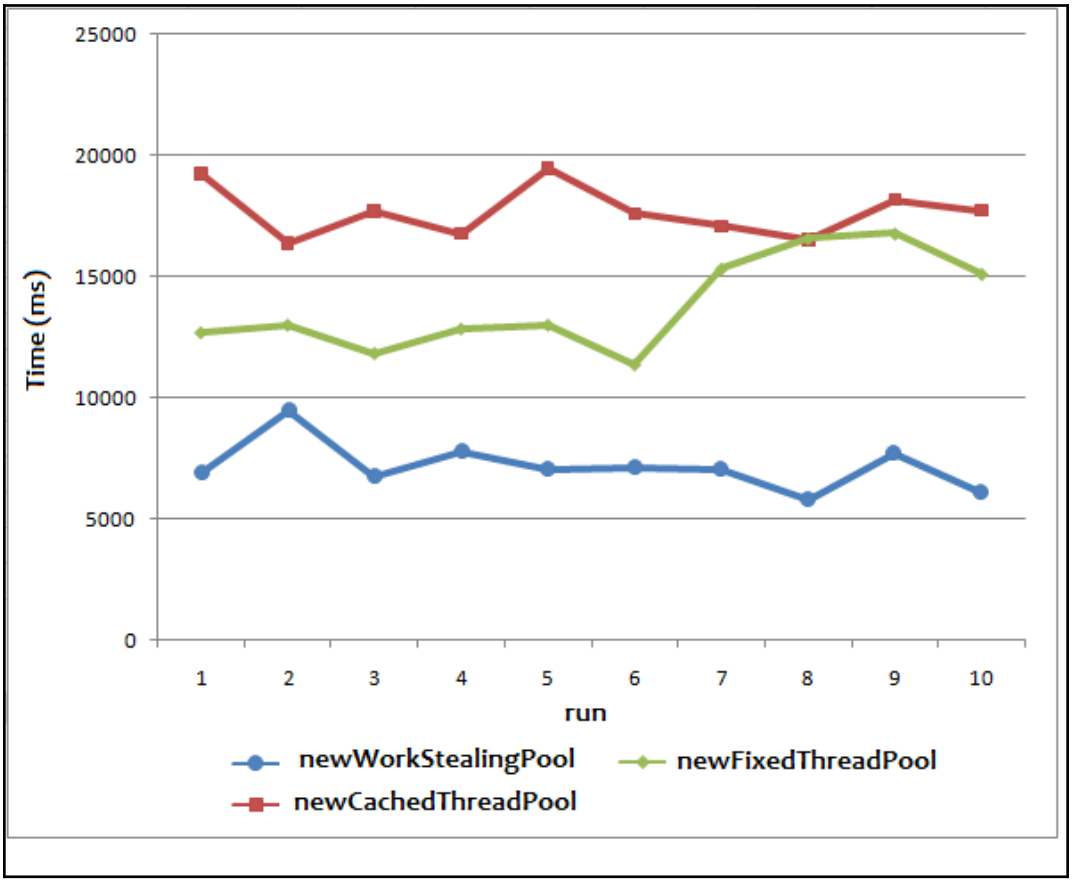

producer
thread
consumer
thread
thread
Assembly Line
thread
push
poll


consumer
producer
thread
Assembly Line
consumer
thread
thread
push
poll
thread

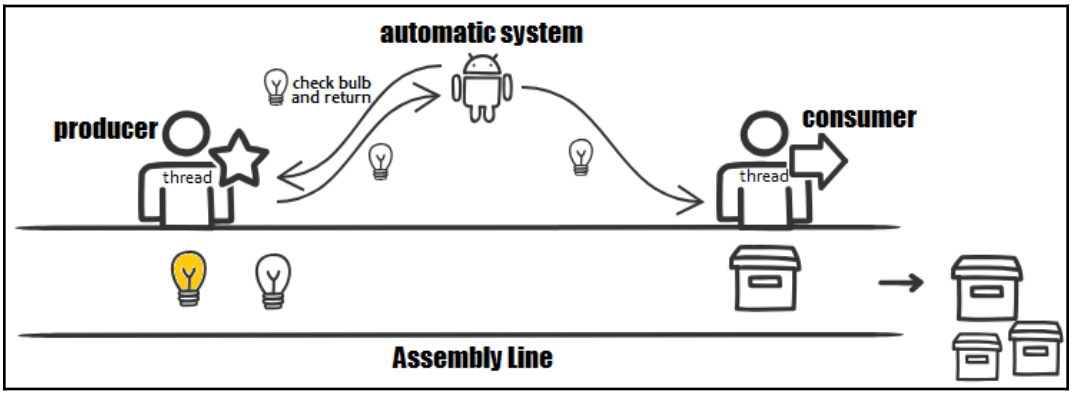

consumer
producer
thread
Assembly Line
consumer
thread
thread
supervisor
push
poll
thread

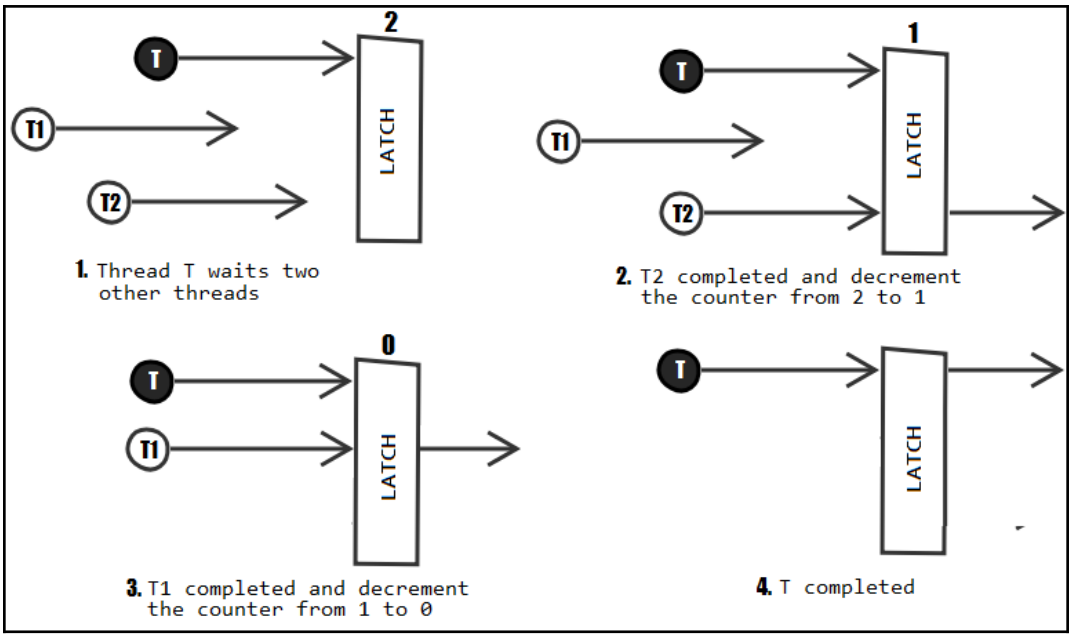

Thread 1
inbound
enqueue
dequeue
Thread 2
queue
Thread 3

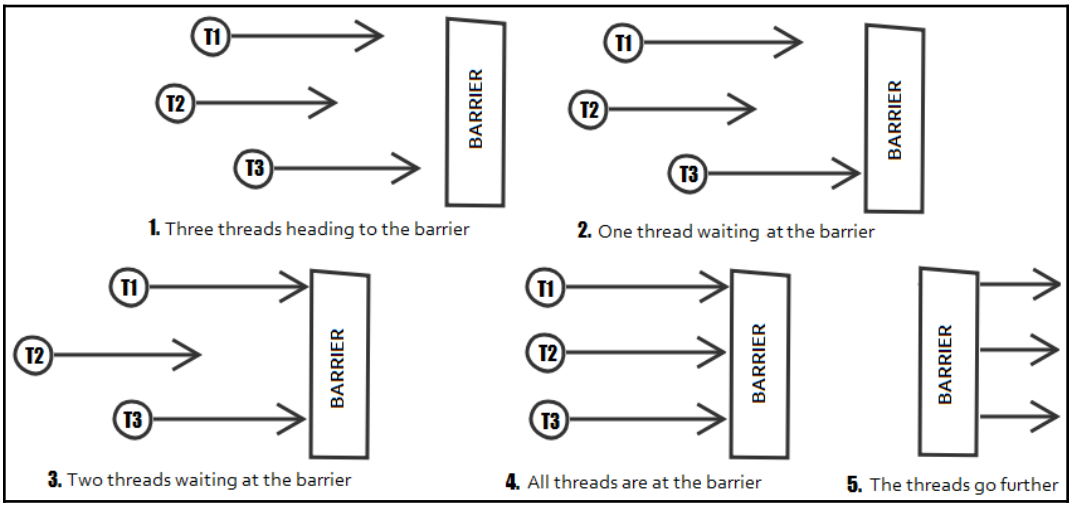

Thread 1
enqueue
dequeue
inbound
enqueue
dequeue
Thread 2
empty
queue
Thread 3
work-stealing

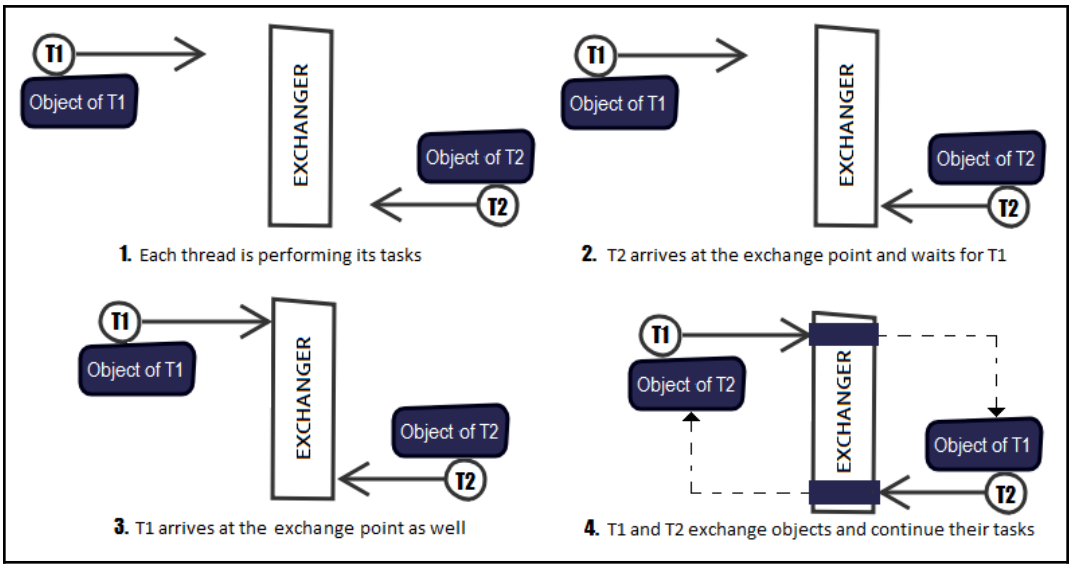

25000
20000
15000
10000
5000
0
Time (ms)
1
2
3
4
5
6
7
8
9
10
run
newWorkStealingPool
newFixedThreadPool
newCachedThreadPool

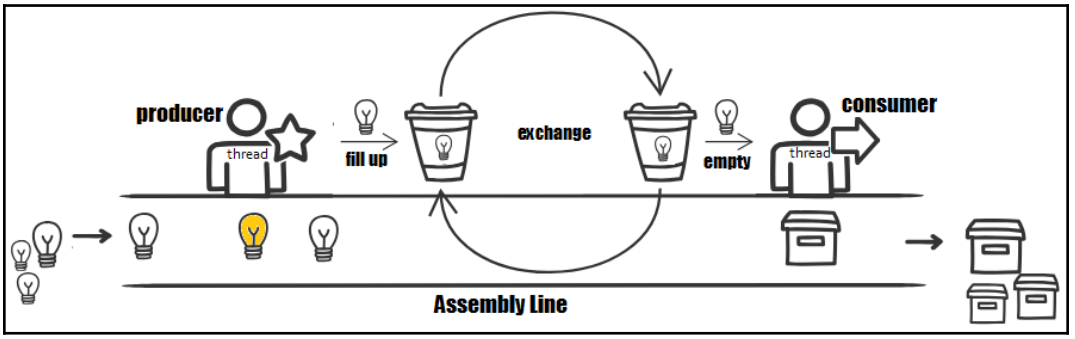
900
800
700
600
500
400
300
200
100
0
Time (ms)
1
2
3
4
5
6
7
8
9
10
run
newWorkStealingPool
newFixedThreadPool
newCachedThreadPool

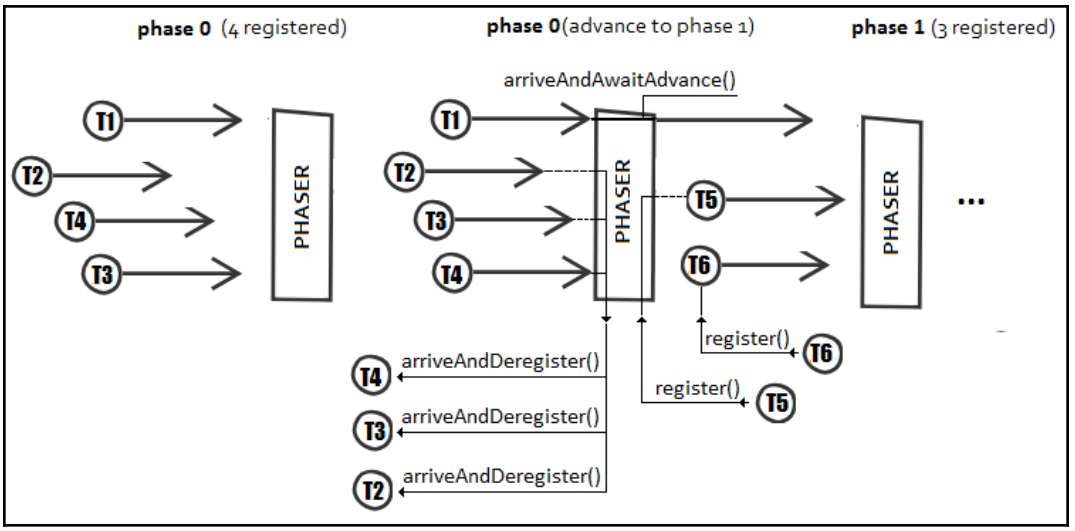

automatic system
check bulb and return
producer
thread
consumer
thread
Assembly Line


2
T
T1
T2
LATCH
1. Thread T waits two other threads
1
T
T1
T2
LATCH
2. T2 completed and decrement the counter from 2 to 1
0
T
T1
LATCH
3. T1 completed and decrement the counter from 1 to 0
T
LATCH
4. T completed

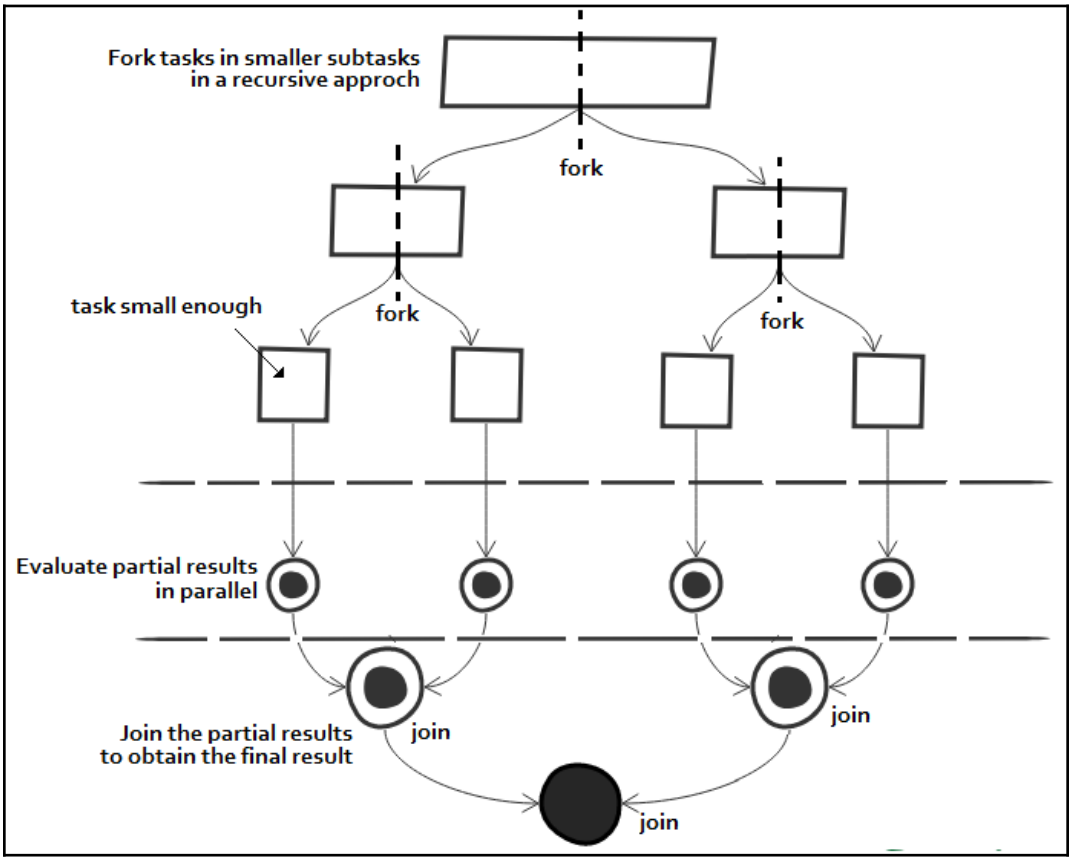

T1
T2
T3
BARRIER
1. Three threads heading to the barrier
T1
T2
T3
BARRIER
2. One thread waiting at the barrier
T1
T2
T3
BARRIER
3. Two threads waiting at the barrier
T1
T2
T3
BARRIER
4. All threads are at the barrier
BARRIER
5. The threads go further

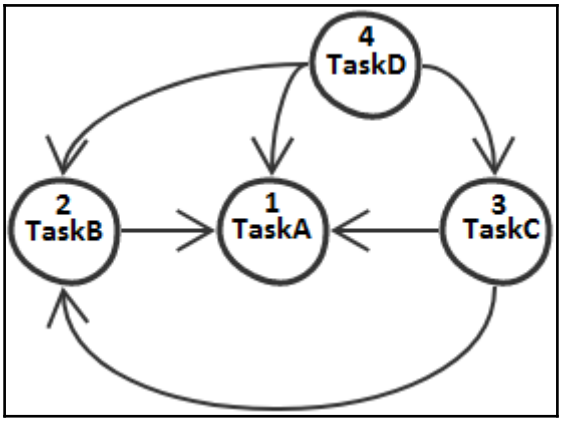

T1
Object of T1
EXCHANGER
Object of T2
T2
1. Each thread is performing its tasks
2. T2 arrives at the exchange point and waits for T1
3. T1 arrives at the exchange point as well
4. T1 and T2 exchange objects and continue their tasks


producer
thread
fill up
exchange
empty
consumer
thread
Assembly Line


phase 0 (4 registered)
phase 0(advance to phase 1)
phase 1 (3 registered)
arriveAndAwaitAdvance()
T1
T2
T4
T3
PHASER
T5
T6
...
register()
arriveAndDeregister()
arriveAndDeregister()
arriveAndDeregister()

Chapter 11: Concurrency - Deep Dive





```
CompletableFuture.supplyAsync(() -> {
     // Code prone to exception
     return "result1";
}).thenApply(r1 -> {
     // Code prone to exception
     return "result2";
}).thenApply(r2 -> {
     // Code prone to exception
     return "result3";
}).thenAccept(r3 -> {
     // Code prone to exception
});
```

Exception in supplyAsync()

```
CompletableFuture.supplyAsync(() -> {
     // Code prone to exception
     return "result1";
}).thenApply(r1 -> {
     // Code prone to exception
     return "result2";
}).thenApply(r2 -> {
     // Code prone to exception
     return "result3";
}).thenAccept(r3 -> {
     // Code prone to exception
});
```

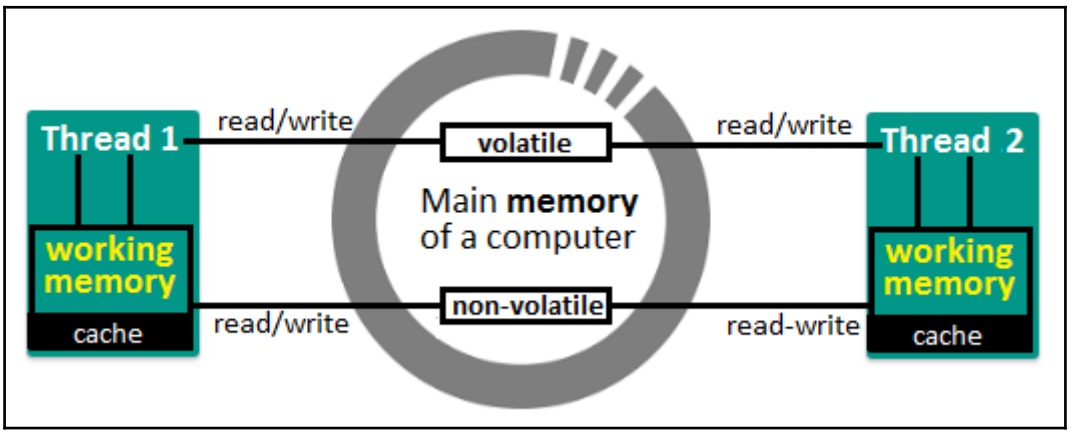
Exception in 1st thenApply()

```
CompletableFuture.supplyAsync(() -> {
     // Code prone to exception
     return "result1";
}).thenApply(r1 -> {
     // Code prone to exception
     return "result2";
}).thenApply(r2 -> {
     // Code prone to exception
     return "result3";
}).thenAccept(r3 -> {
     // Code prone to exception
});
```

Exception in 2nd thenApply()

```
CompletableFuture.supplyAsync(() -> {
     // Code prone to exception
     return "result1";
}).thenApply(r1 -> {
     // Code prone to exception
     return "result2";
}).thenApply(r2 -> {
     // Code prone to exception
     return "result3";
}).thenAccept(r3 -> {
     // Code prone to exception
});
```

Exception in thenAccept()

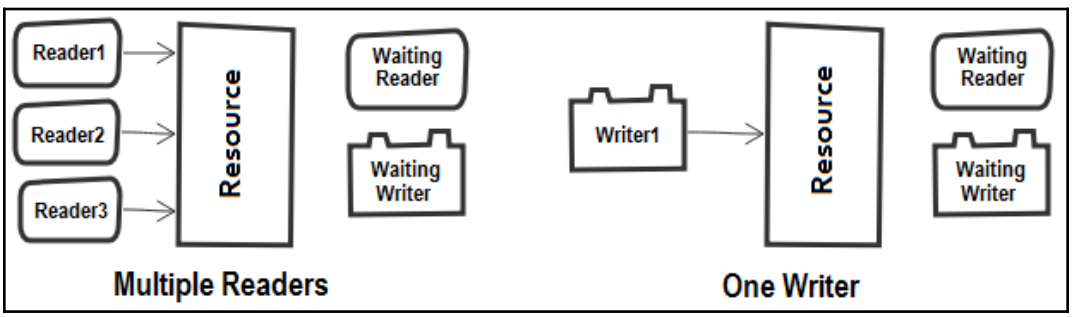


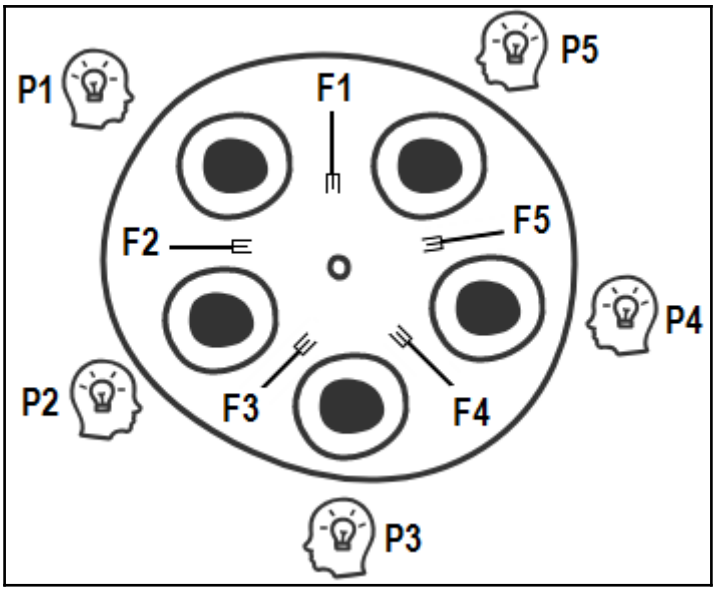
newCondition

```
Lock/ReentrantLock lock = new ReentrantLock();
Condition condition = lock.newCondition();

public void execute() throws InterruptedException {
  lock.lock();
  try {
     ...
     while/if(some_condition) {
       condition.await();
     }
  } finally {
     lock.unlock();
  }
}
```

```
lock.lock();
try {
  condition.signal(All)();
} finally {
   lock.unlock();
}
```

When await() is called the thread releases the lock. After getting the signal to continue the thread must acquire the lock again.

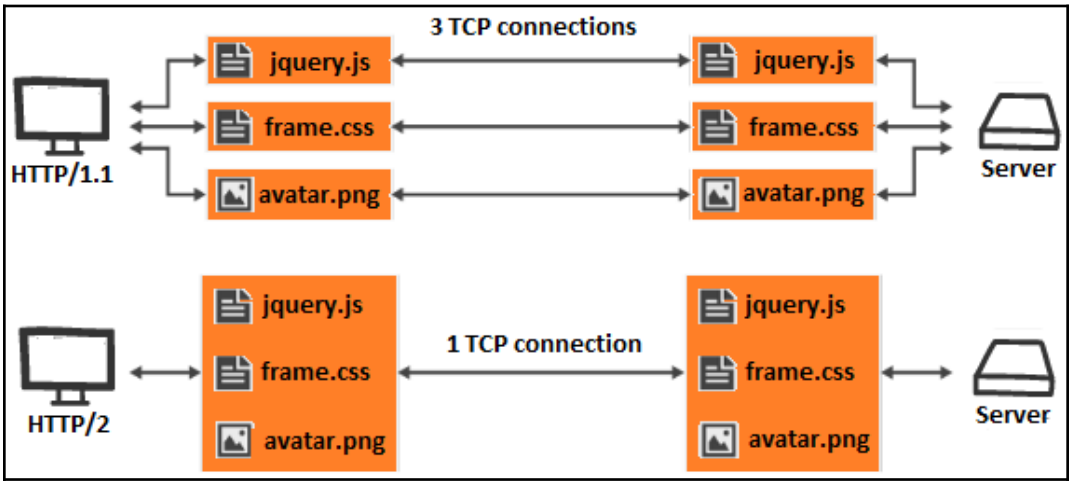



P1
P5
F1
F2
F5
P4
P2
F3
F4
P3

Chapter 12: Optional

No images

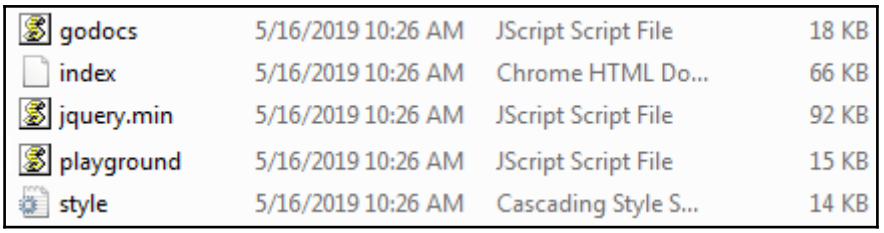
Chapter 13: The HTTP Client and WebSocket APIs




```
--779d334bbfa749fdb1f4d115cd18a0cd
Content-Disposition: form-data; name="author"

Lorem Ipsum Generator
--779d334bbfa749fdb1f4d115cd18a0cd
Content-Disposition: form-data; name="filefield"; filename="figure.png"
Content-Type: image/png

%PNG

¯ Øæ™>ïïA-  8   -ò }Ÿ>)wð{þûó  °Ÿç÷Kt Ï| ³ç Ï Øå™>ïÓ½}'ûÚ SÁhOcáu~¿ ¾o9Û÷¹ß:ë÷¸
¯'Ó 4V>ÇØ/th^\œ£¦ Ä ûÐÏÑ `ŸÎ} Bý «"è iaQ_Q »{«á i¸>Y_O )yàLeÊÁ|®#+N ¼4 ô}¹V|eú
¯Ê ›à>  Ö;+?Q~æg‹t_Çª%¢ Ôƒ* ÊÇ0ðIÓô§Ü äÿ  Vx#^^Ö3Çm ~Ú9~Ç(W·Ù
¯zí ãà°Á ã rŸR ^  TÉ«øŸô}š·p  .c· O"-
--779d334bbfa749fdb1f4d115cd18a0cd--
```

part 1

part2


godocs 5/16/2019 10:26 AM JScript Script File 18 KB
index 5/16/2019 10:26 AM Chrome HTML Do... 66 KB
jquery.min 5/16/2019 10:26 AM JScript Script File 92 KB
playground 5/16/2019 10:26 AM JScript Script File 15 KB
style 5/16/2019 10:26 AM Cascading Style S... 14 KB