

Codigo Fuente

javac

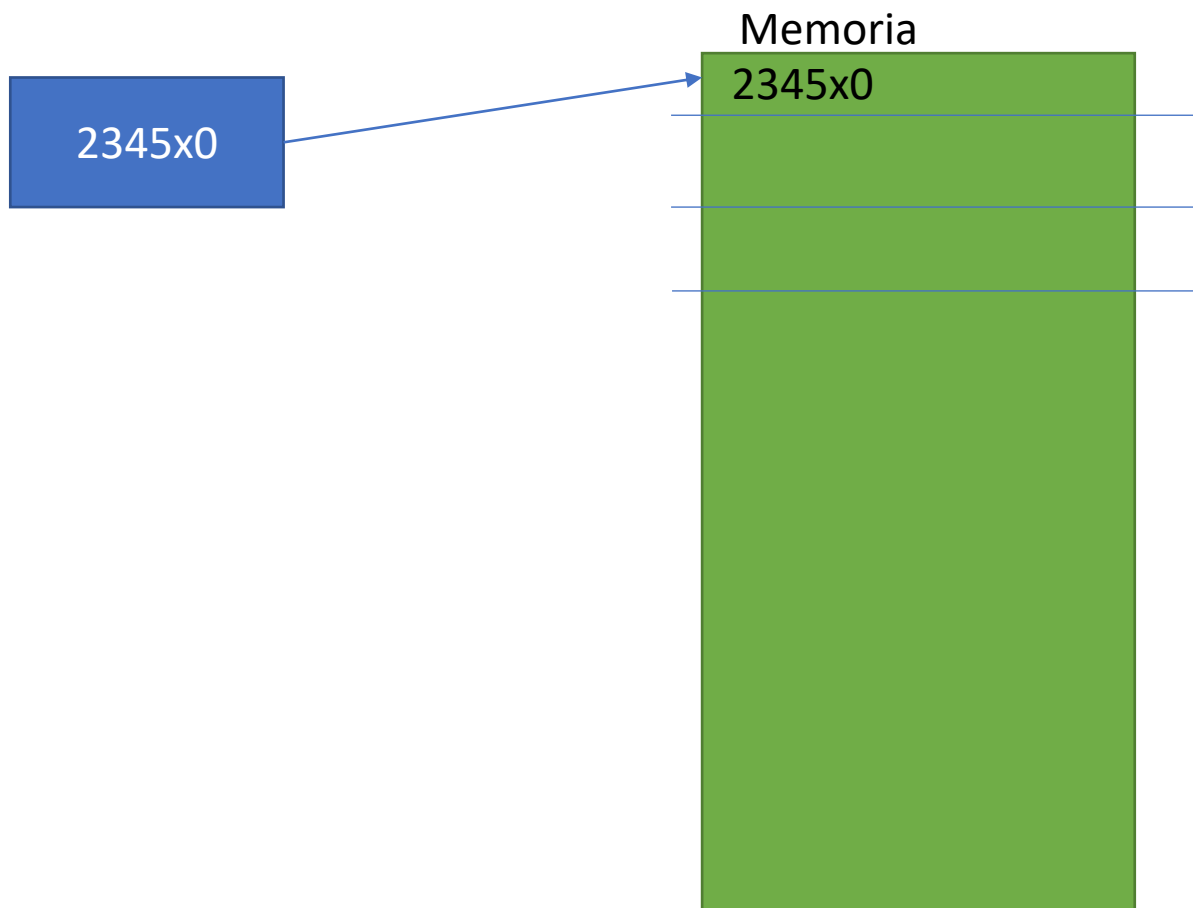
bytecode
.class

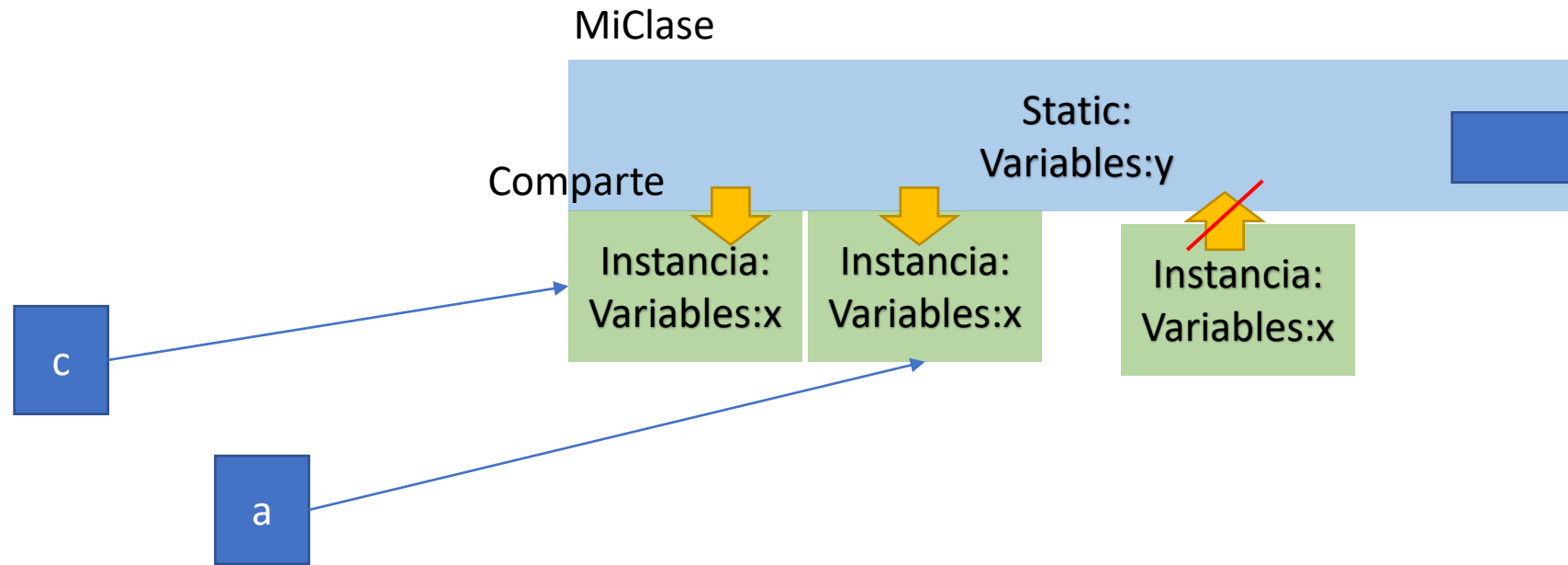
Vendor:
Oracle: JVM hotspot
IBM: J9
AWS: cordela
BLUE

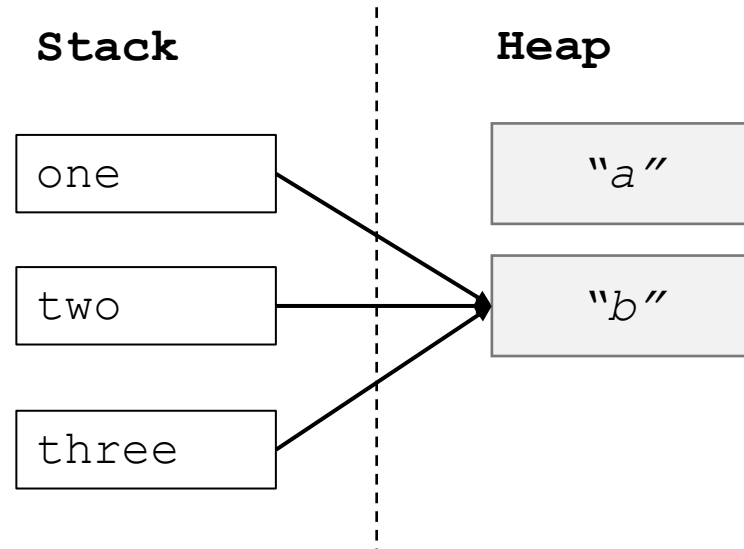
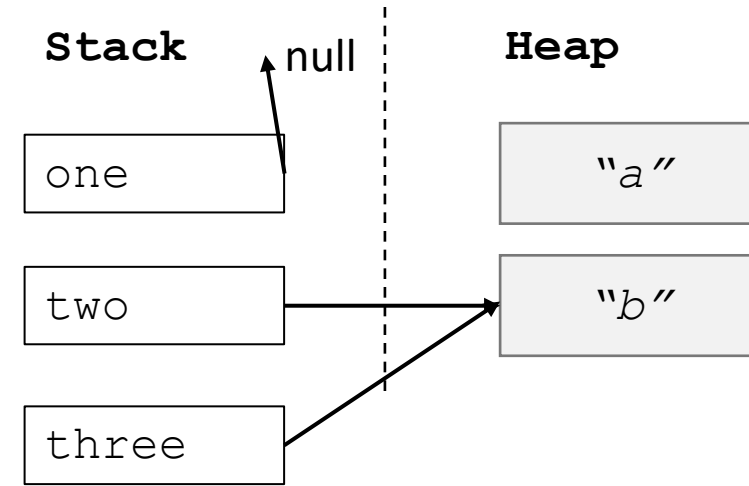
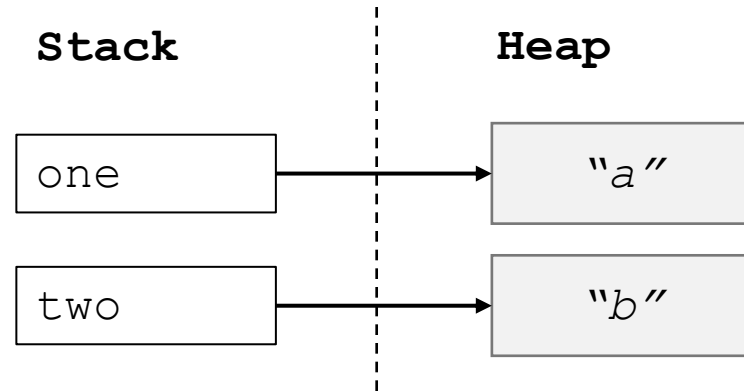
JVM

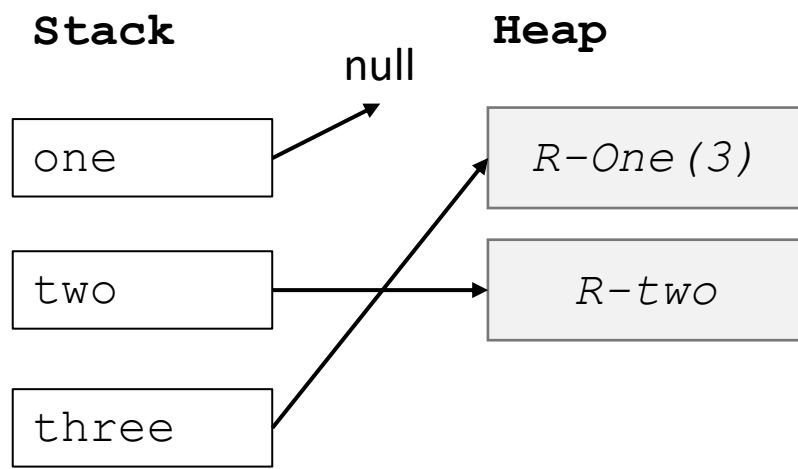
S.O

HW -> instrucciones

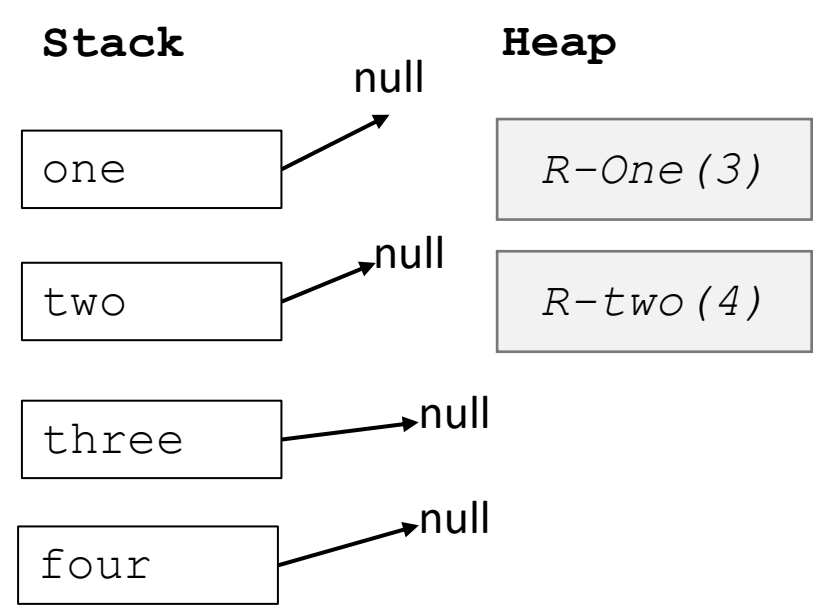




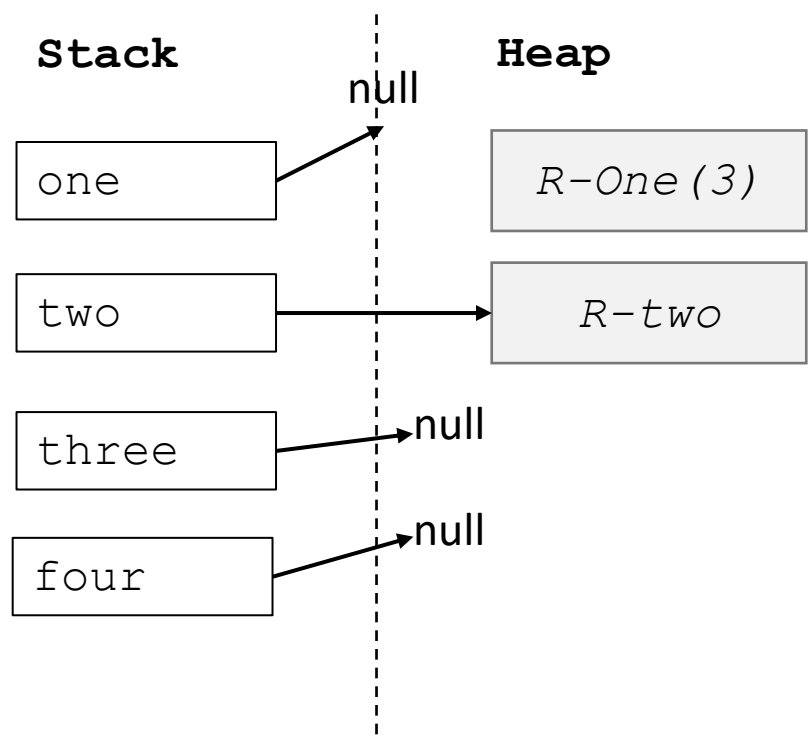




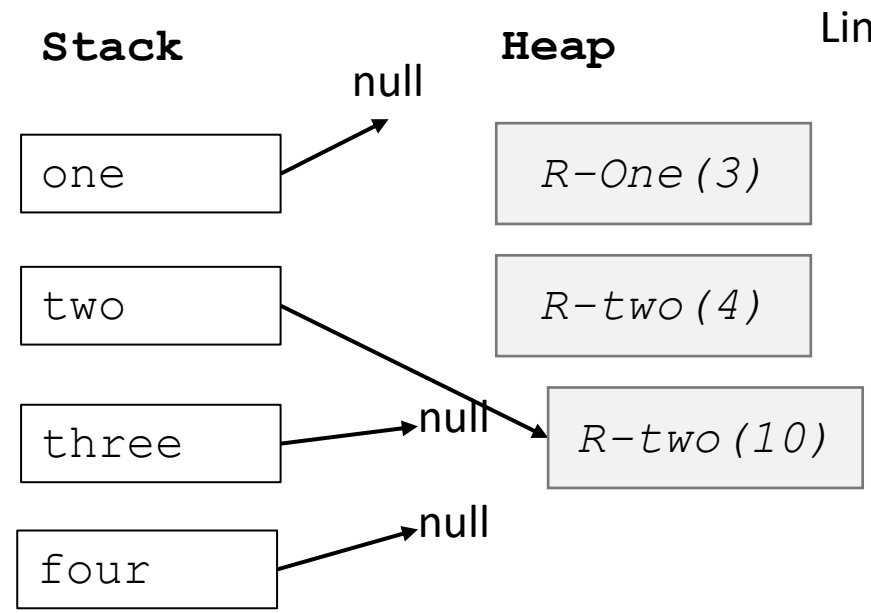
Linea 6



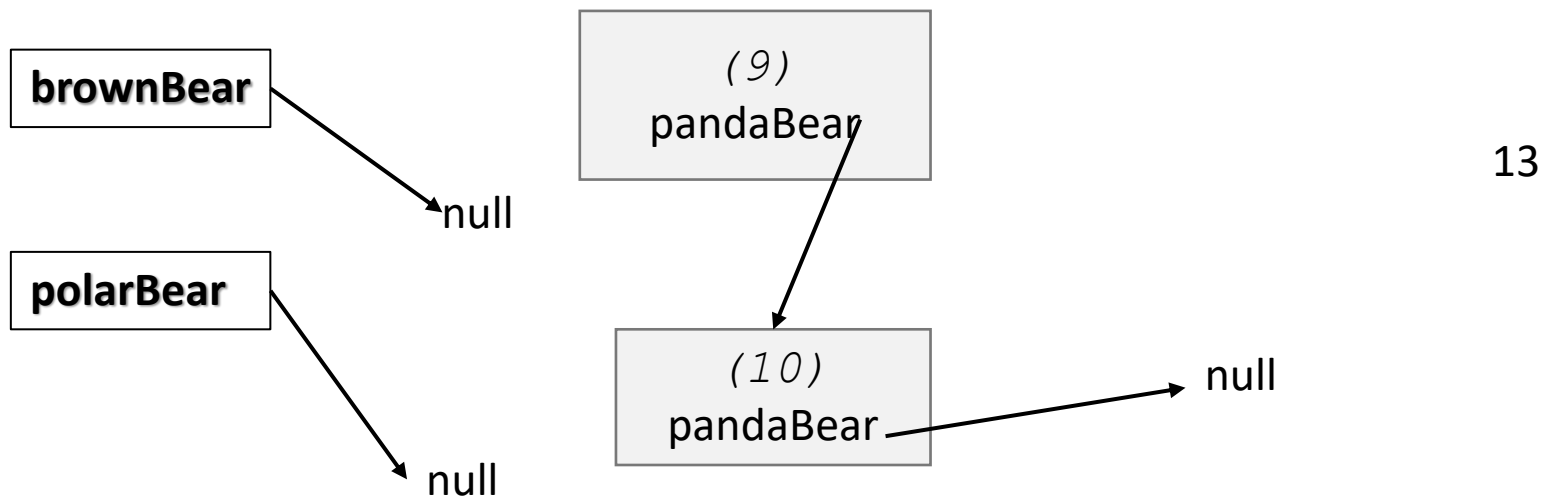
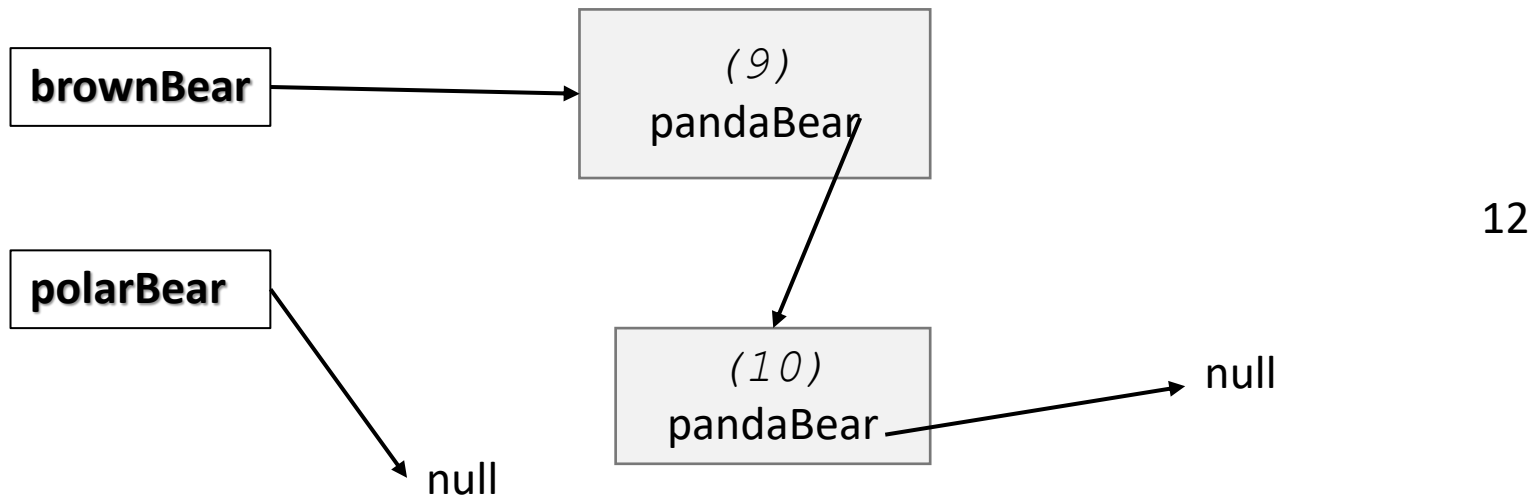
Linea 9

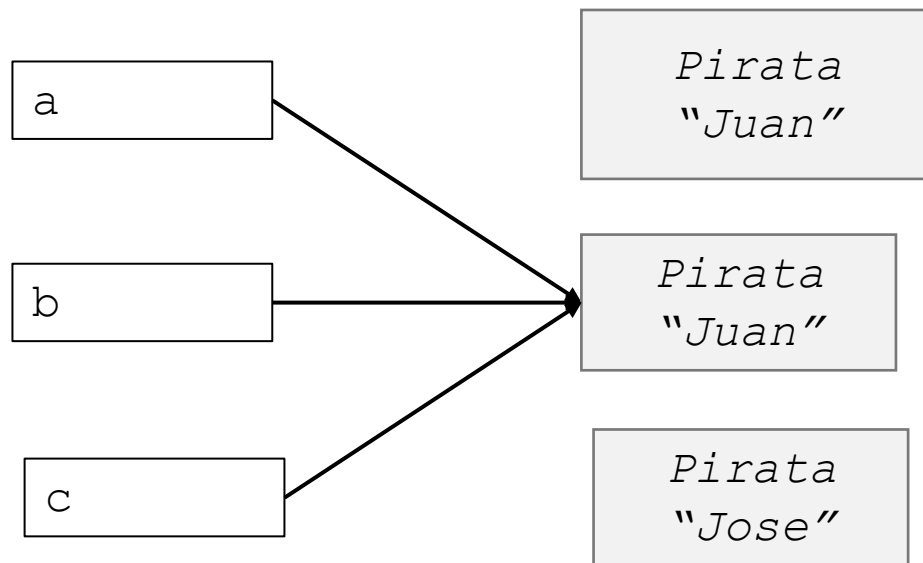
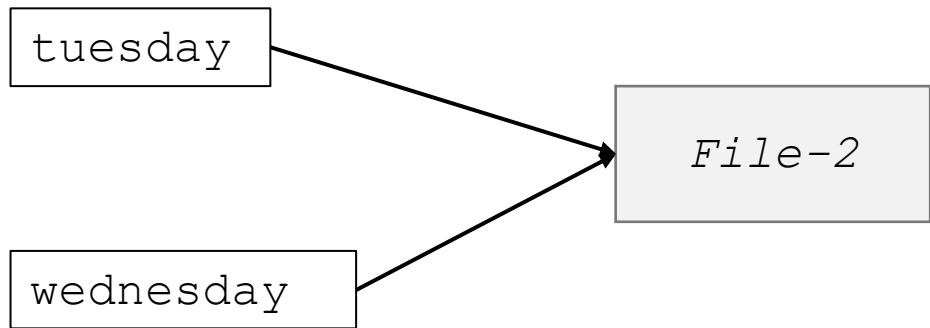
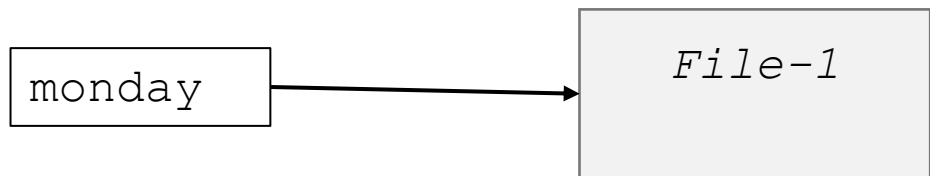


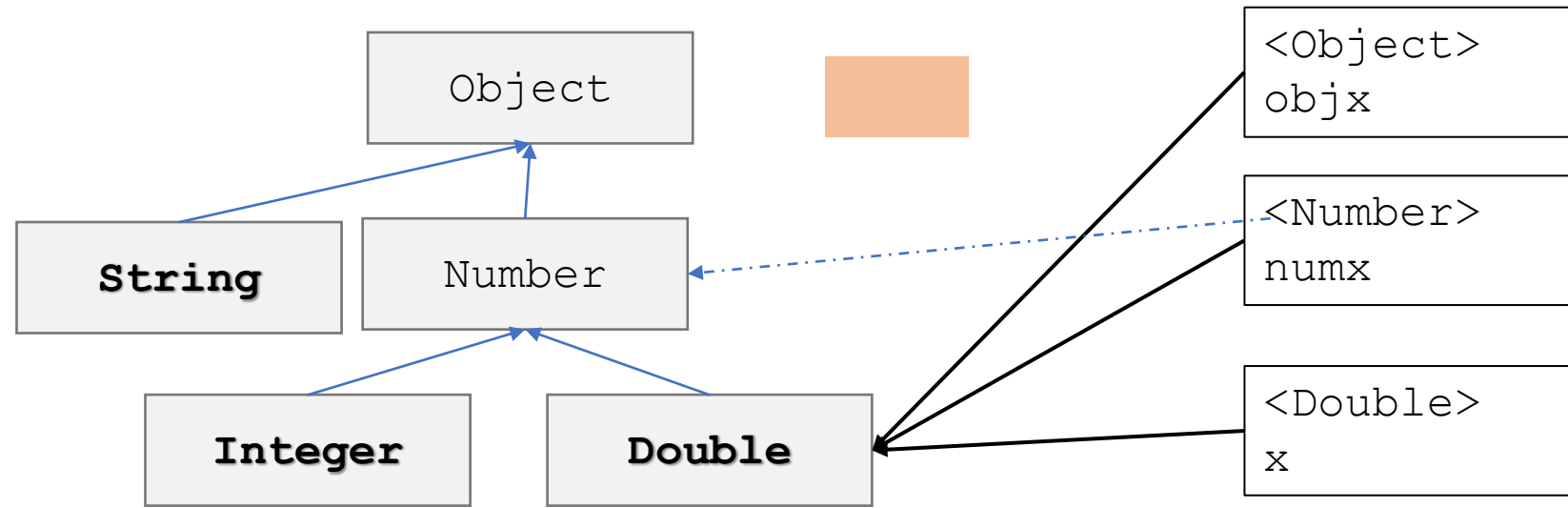
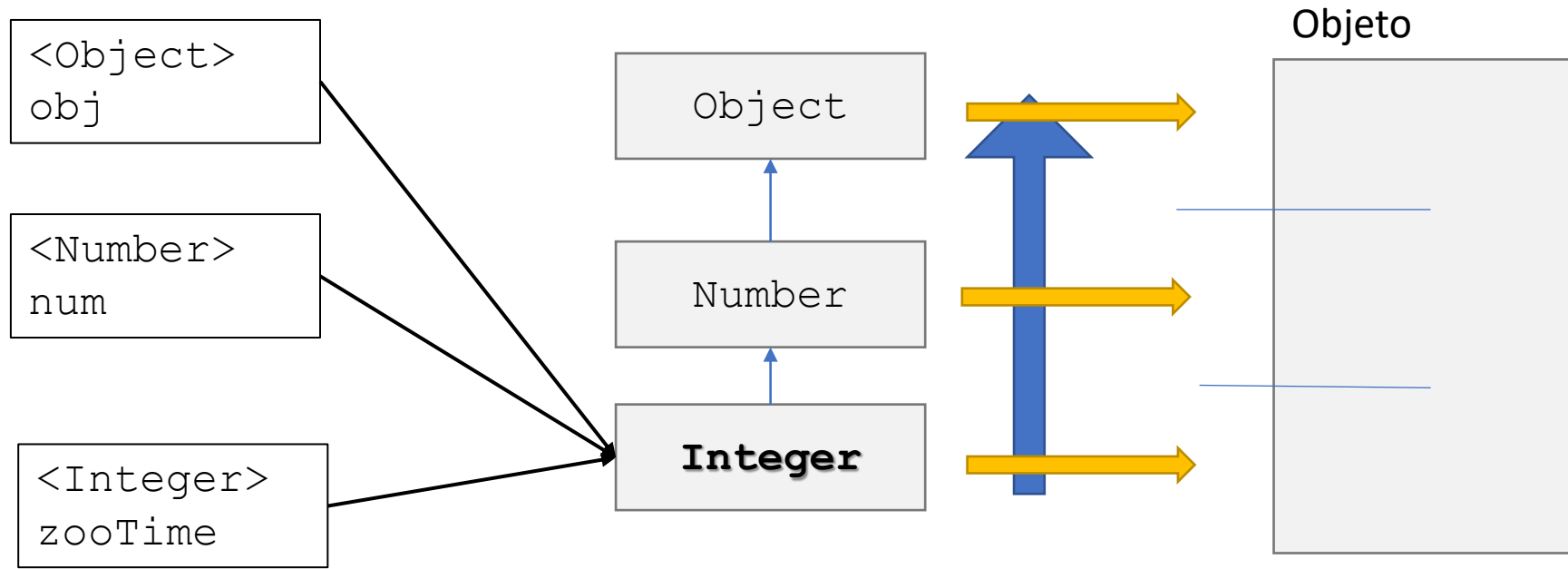
Linea 8



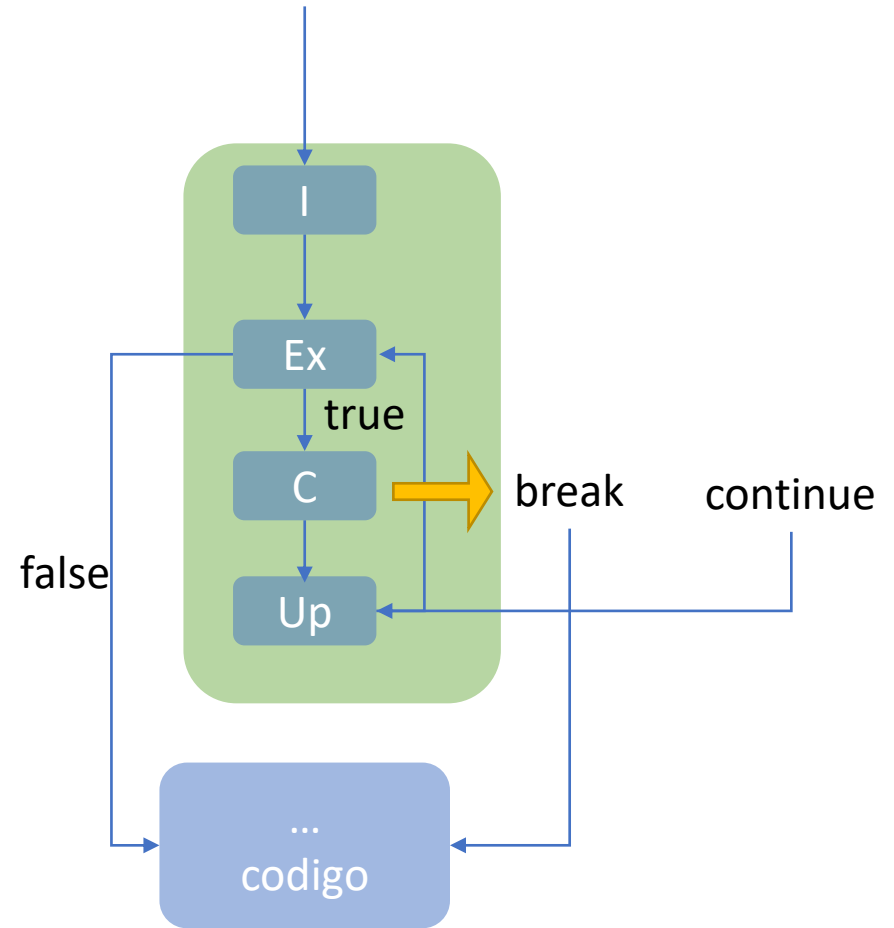
Linea 11

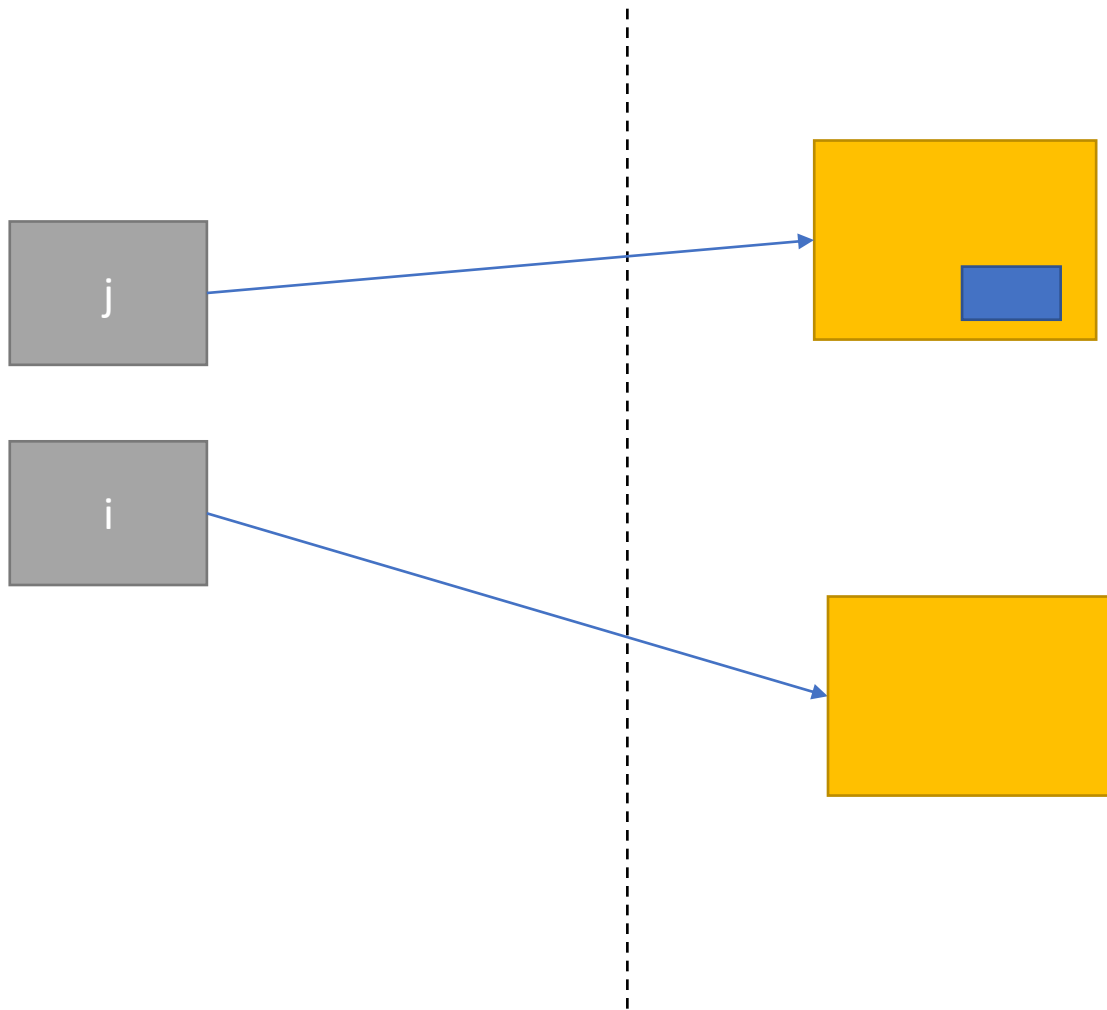


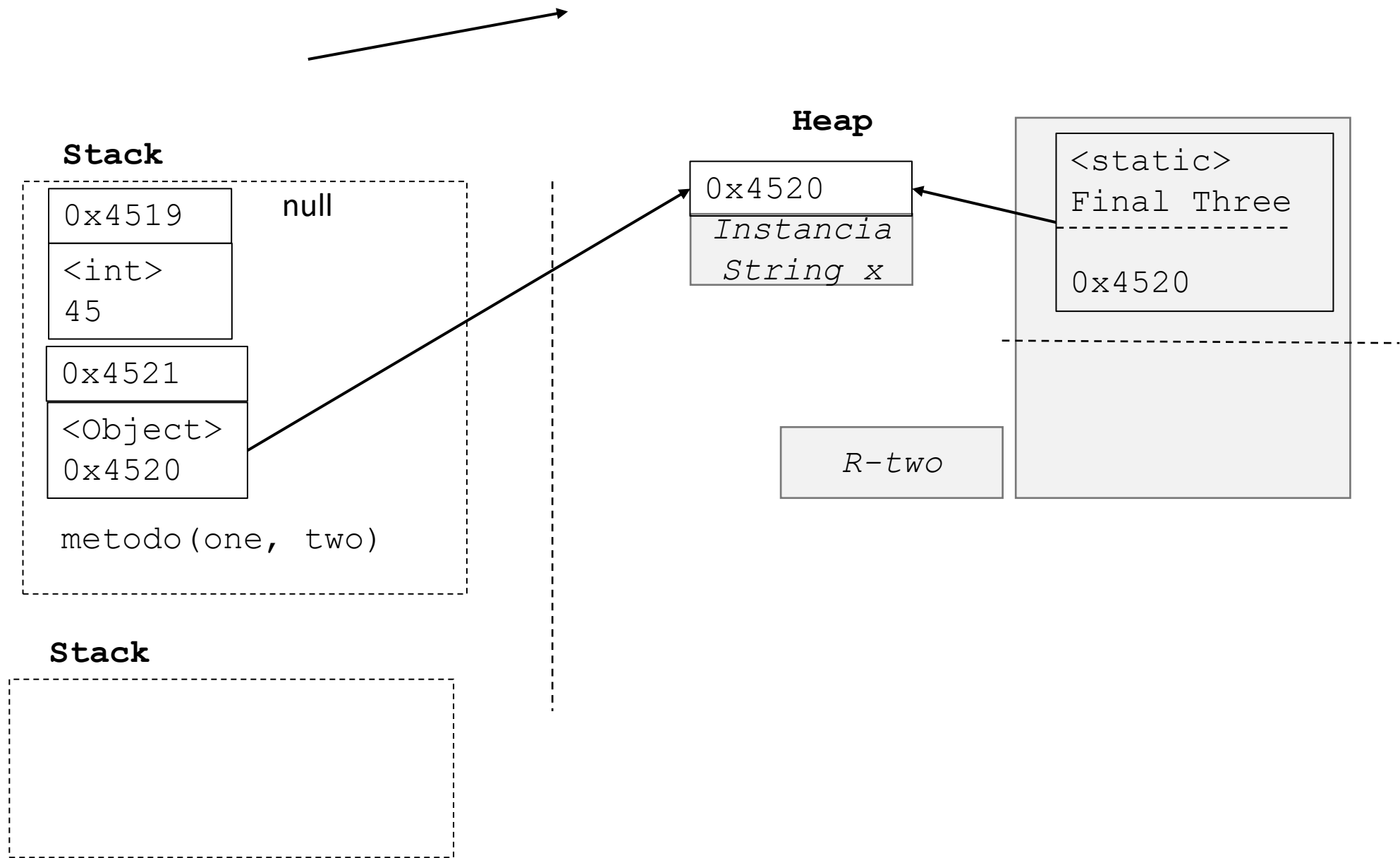


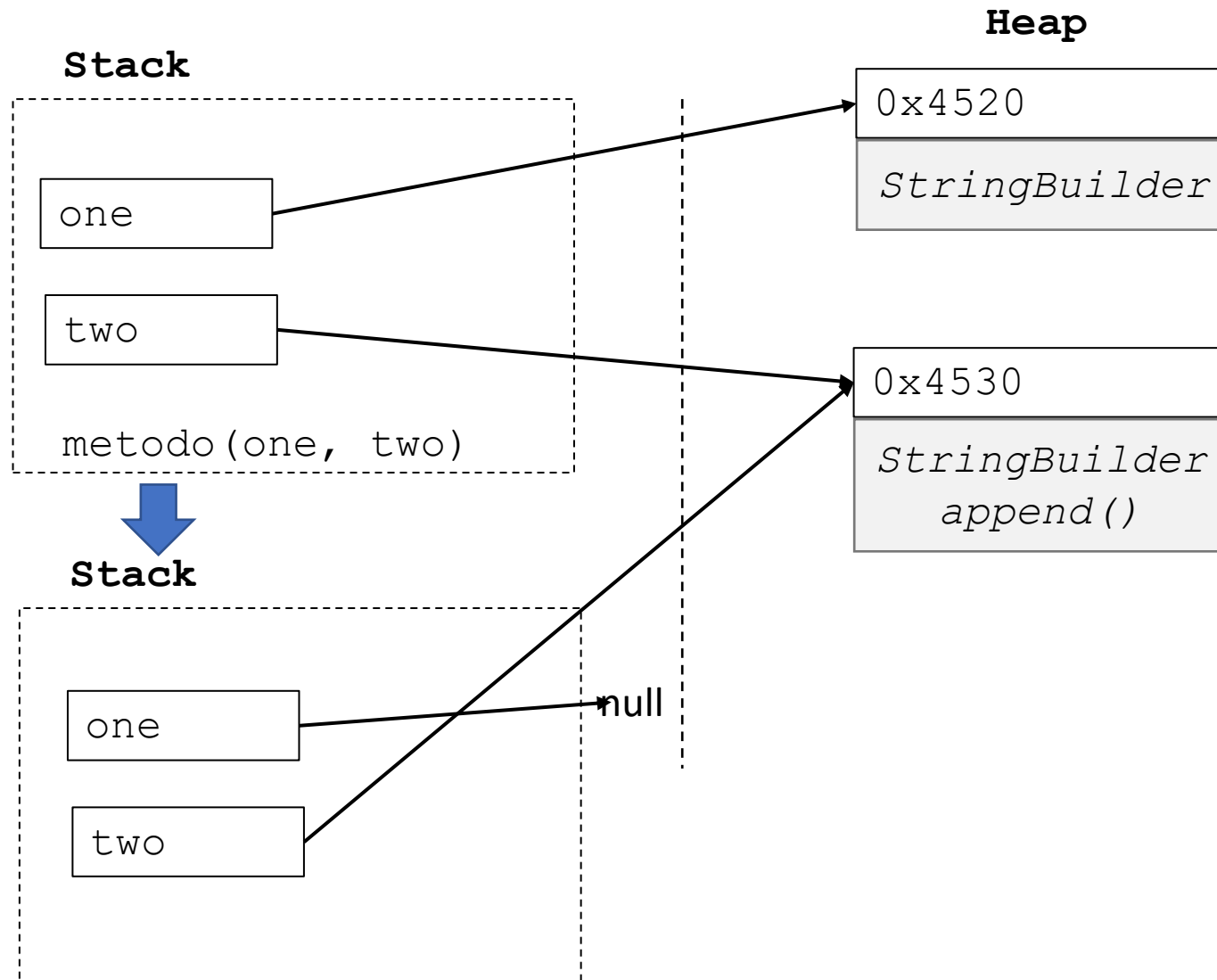


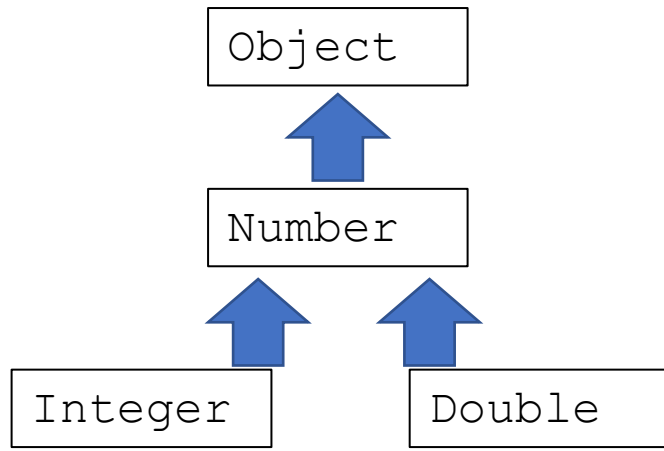
```
For(I; Ex; Up){  
    C;  
    ....;  
}
```



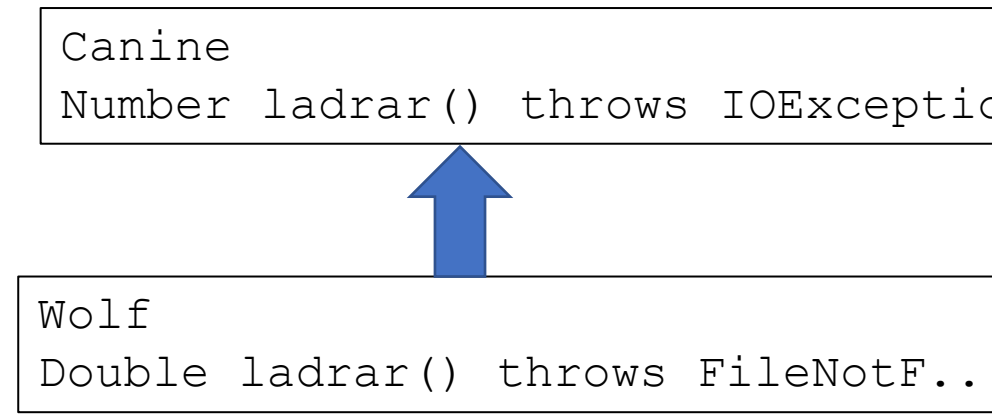
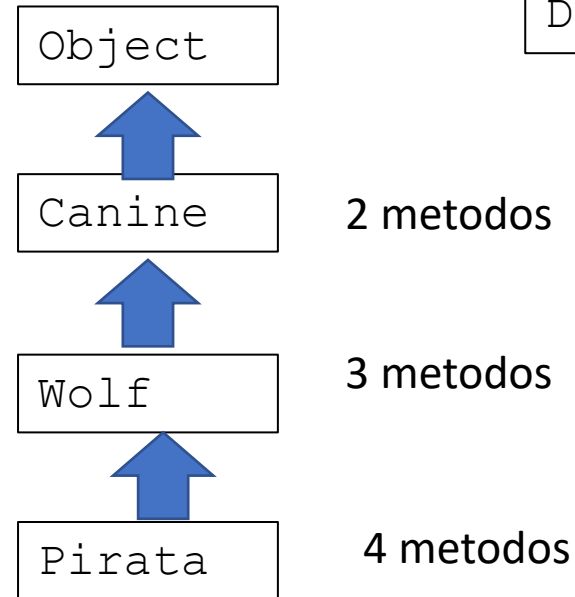


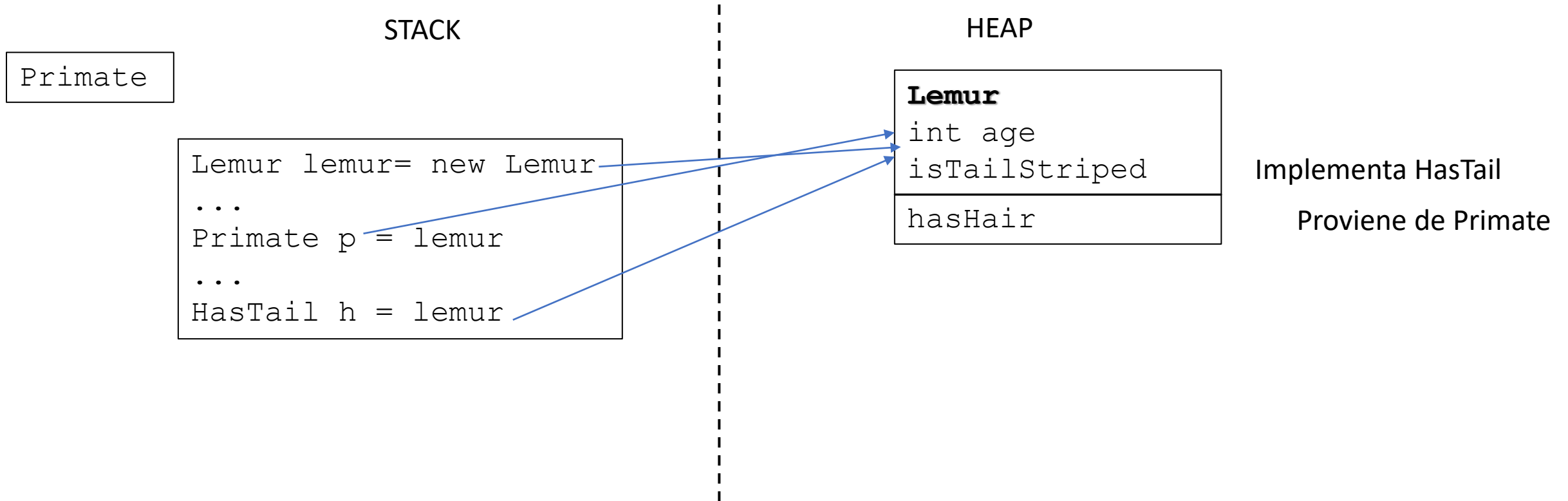
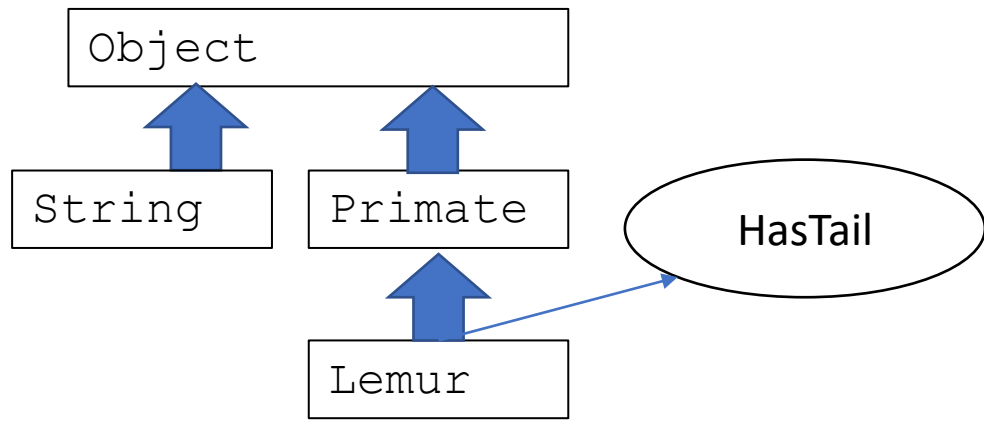


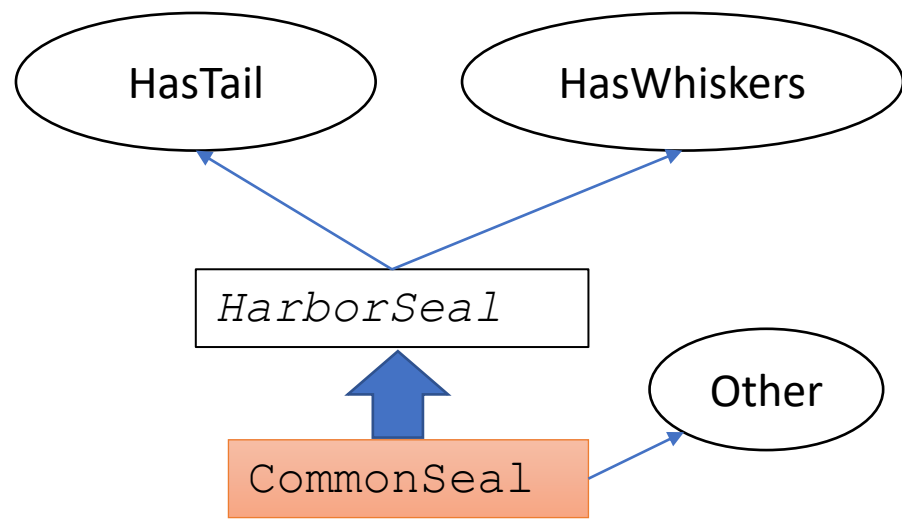




Canine c = [redacted]
Number n = c.ladrar();







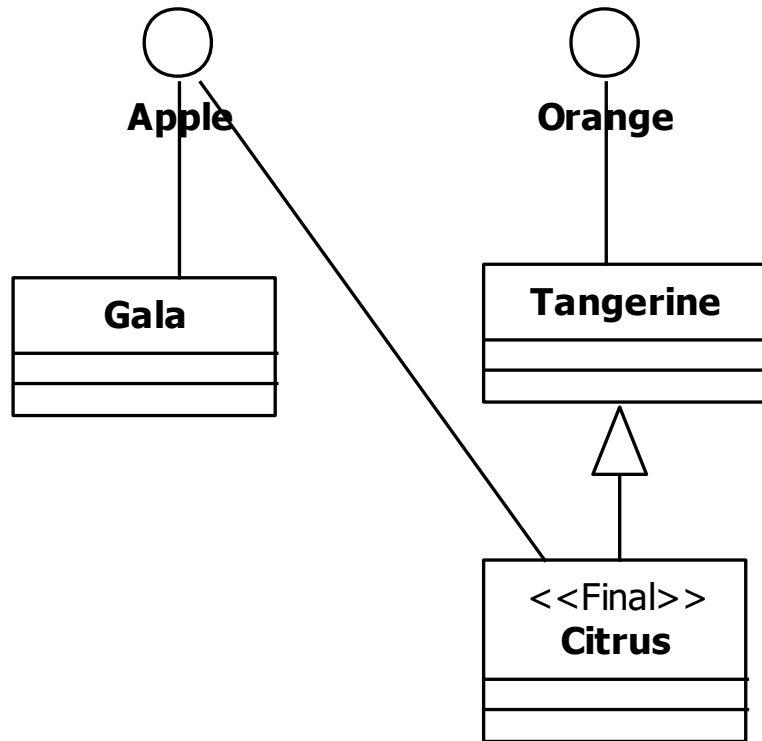
String

Stack

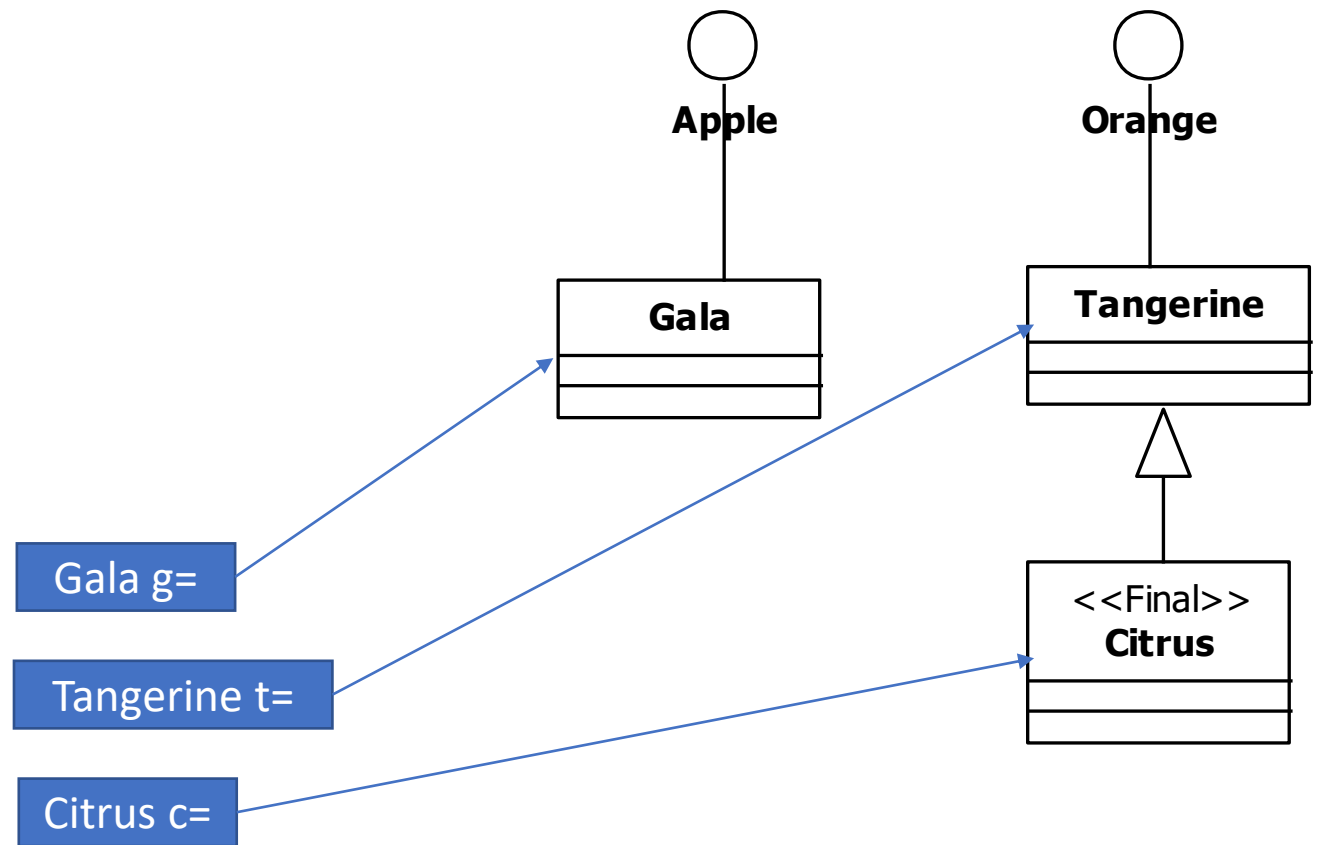
HasTail h = ...

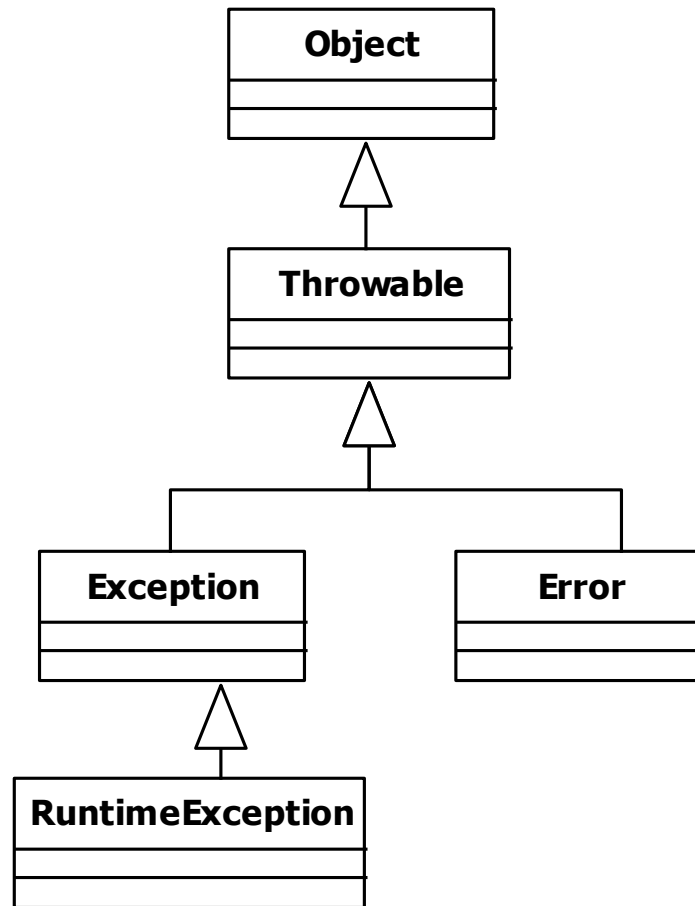
Heap

CommonSeal



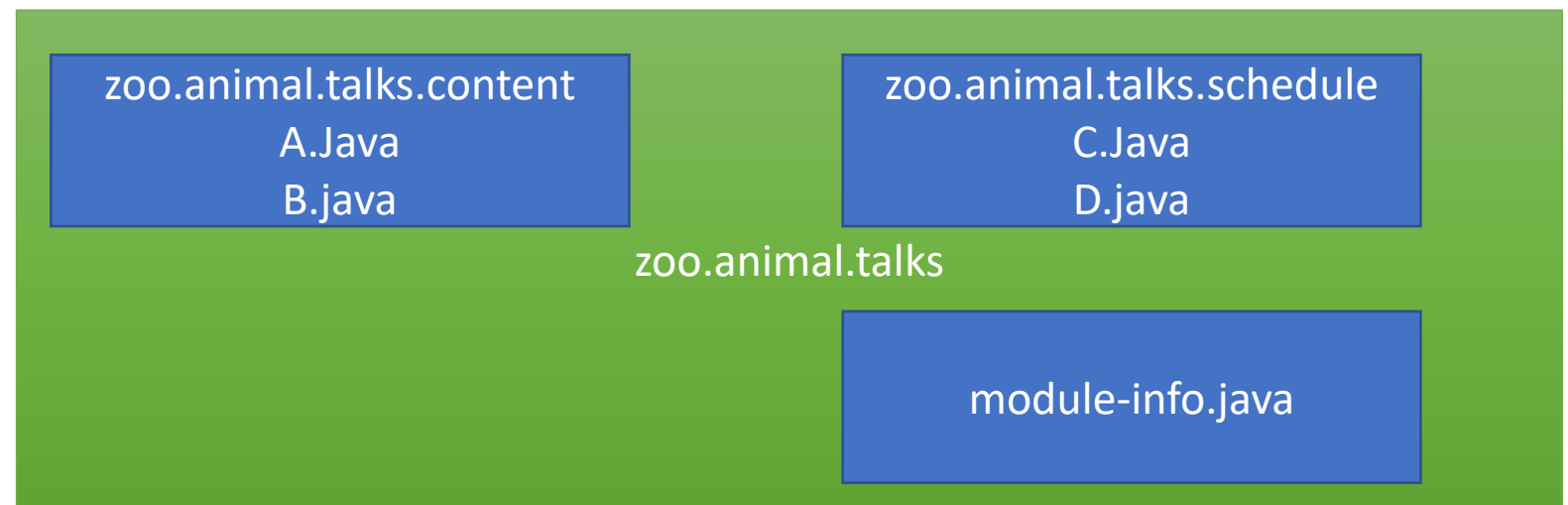
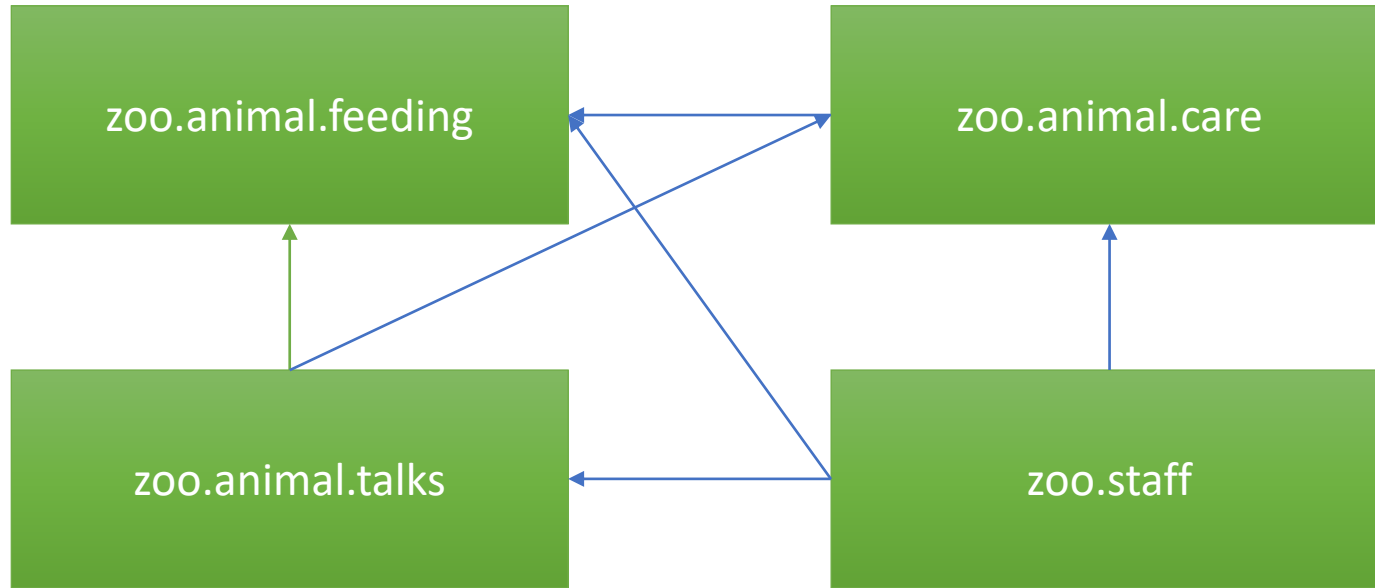
```
Tangerine t=new Citrus()
```





Checked : Exception y las que extiendan de esta
(sin pasar RuntimeException)

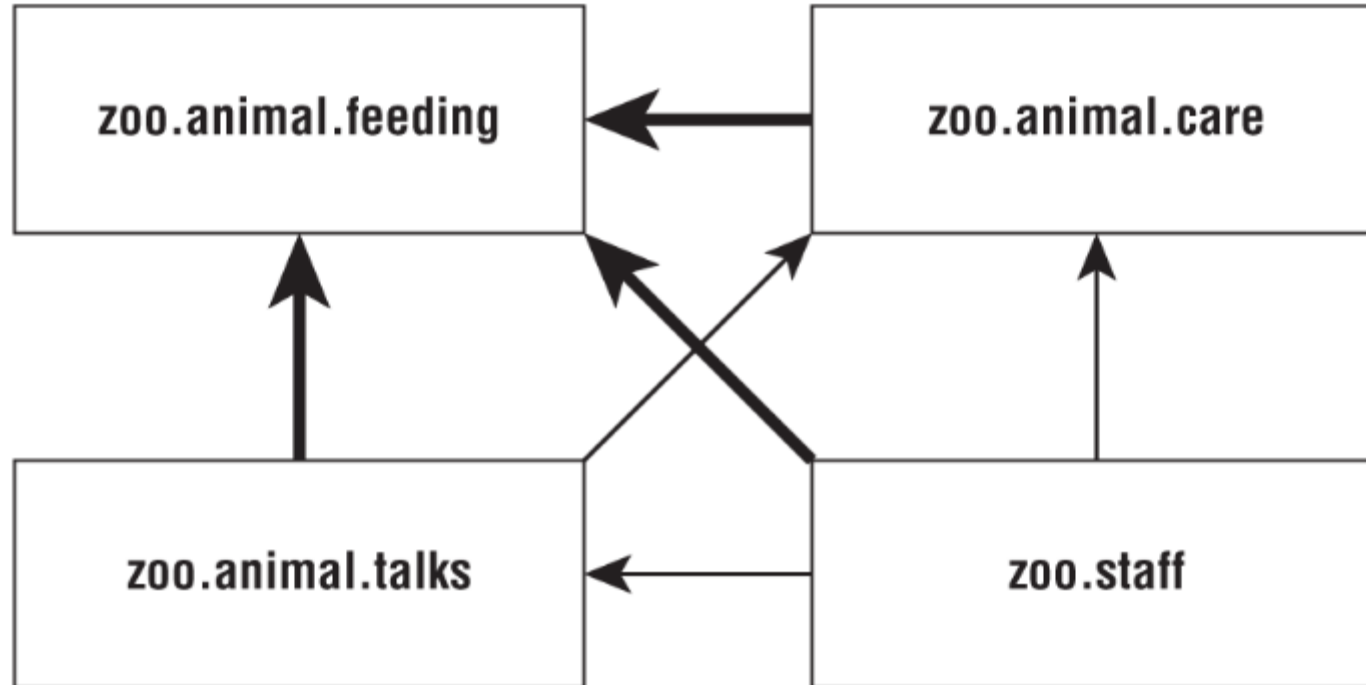
Unchecked: RuntimeException y las que extiendan de estas



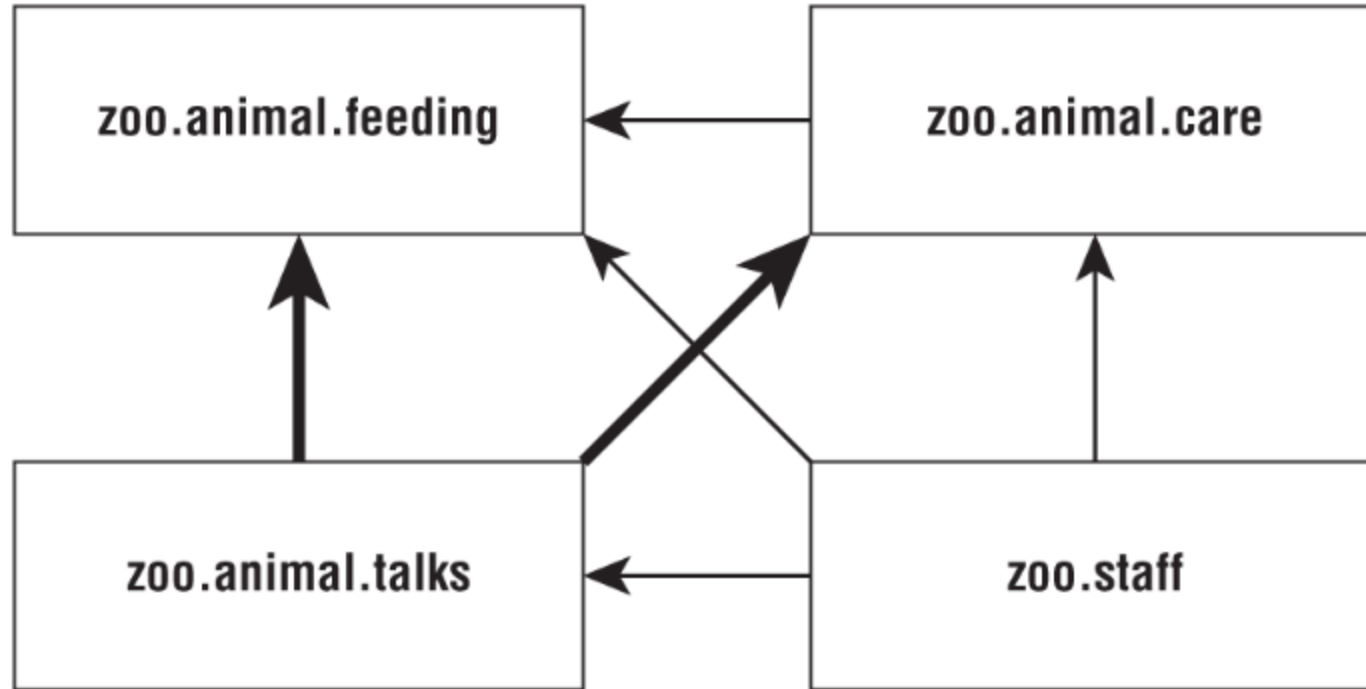
Beneficios:

1. Mejor control de Accesos
2. Clara gestión de dependencia
3. Builds Personalizados
4. Mejora Performance
5. Package Unificado

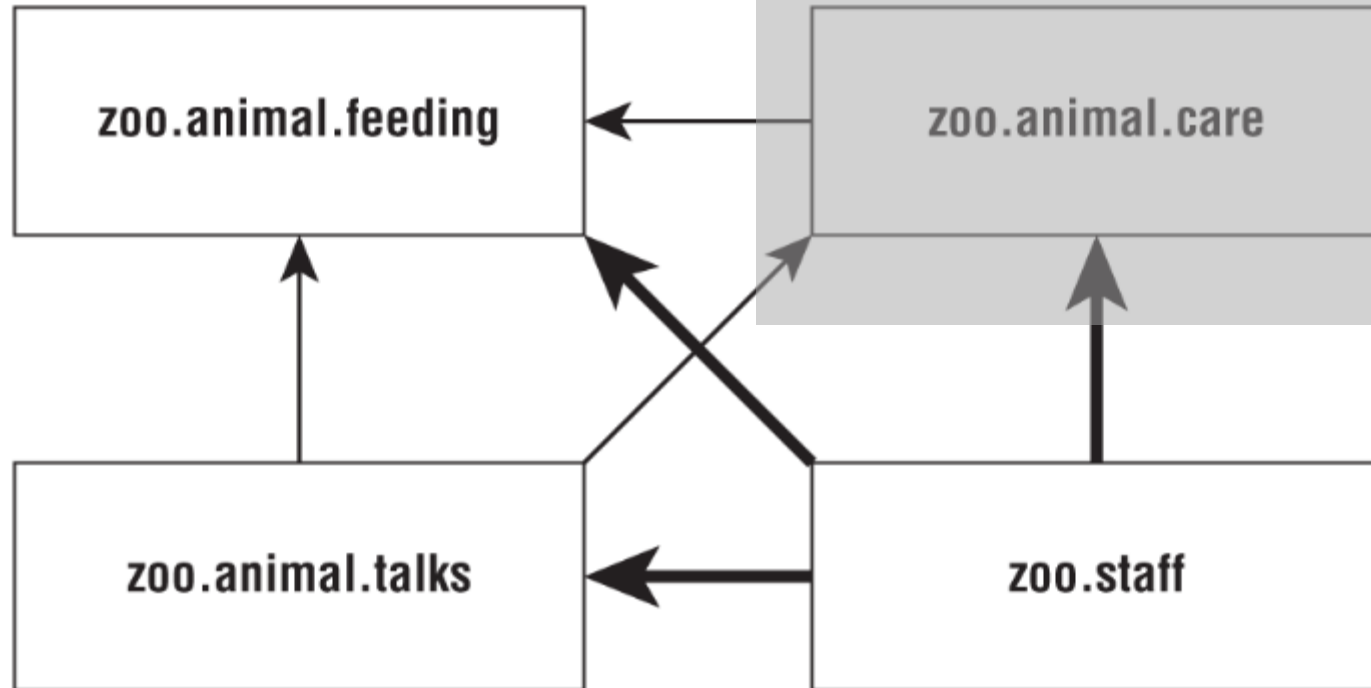
Zoo

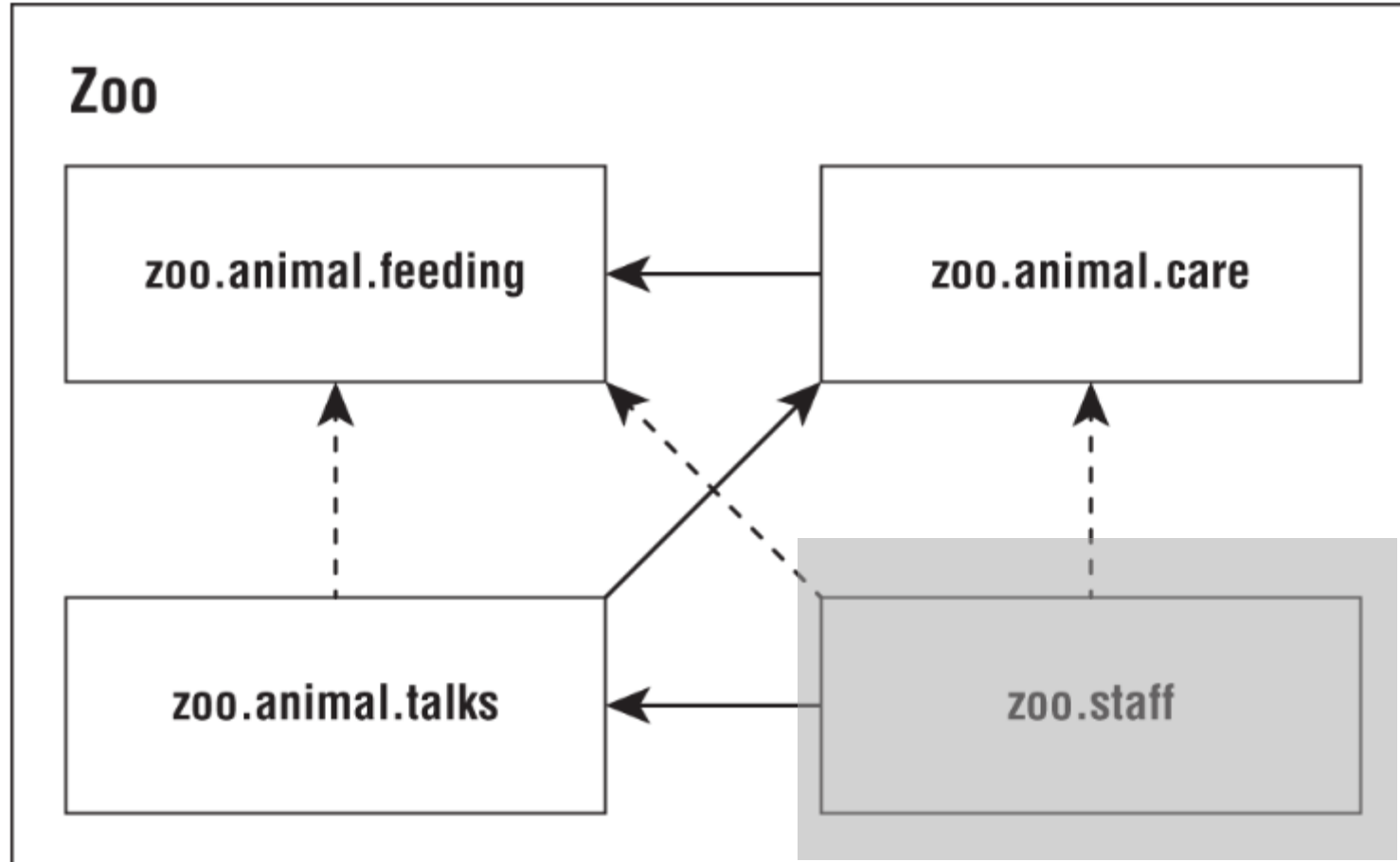


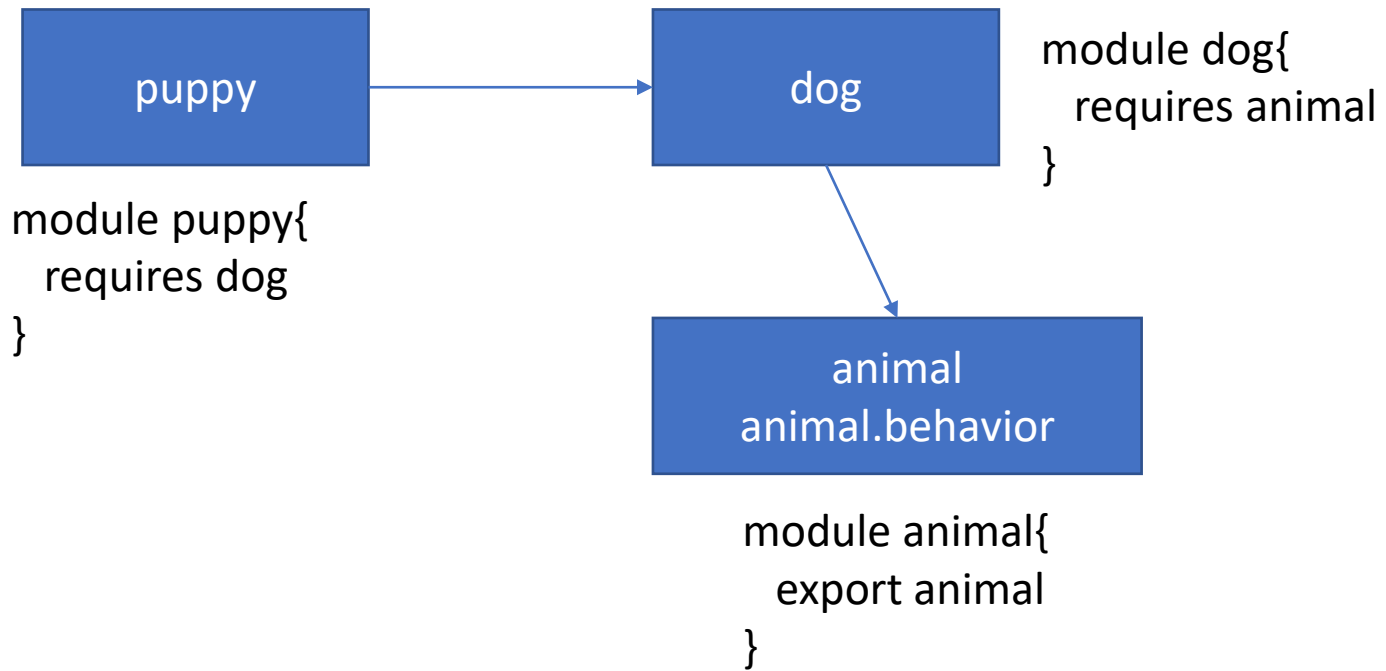
Zoo



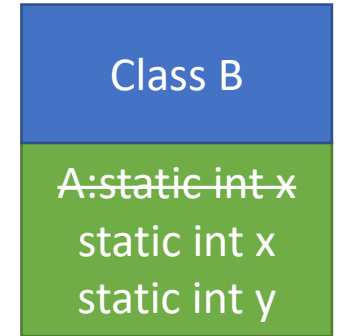
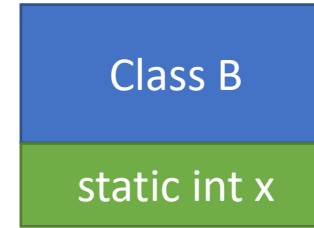
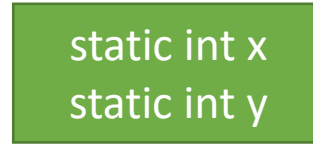
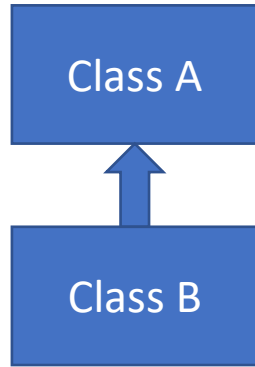
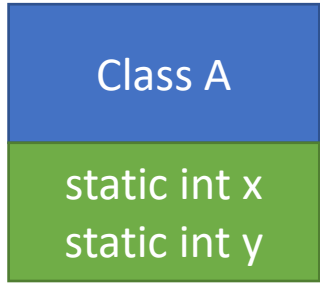
Zoo



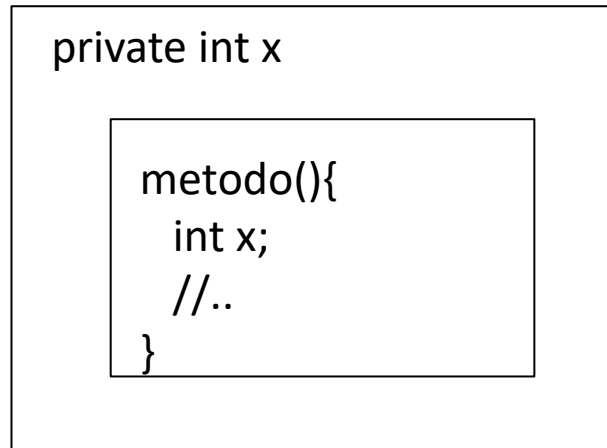




CH12

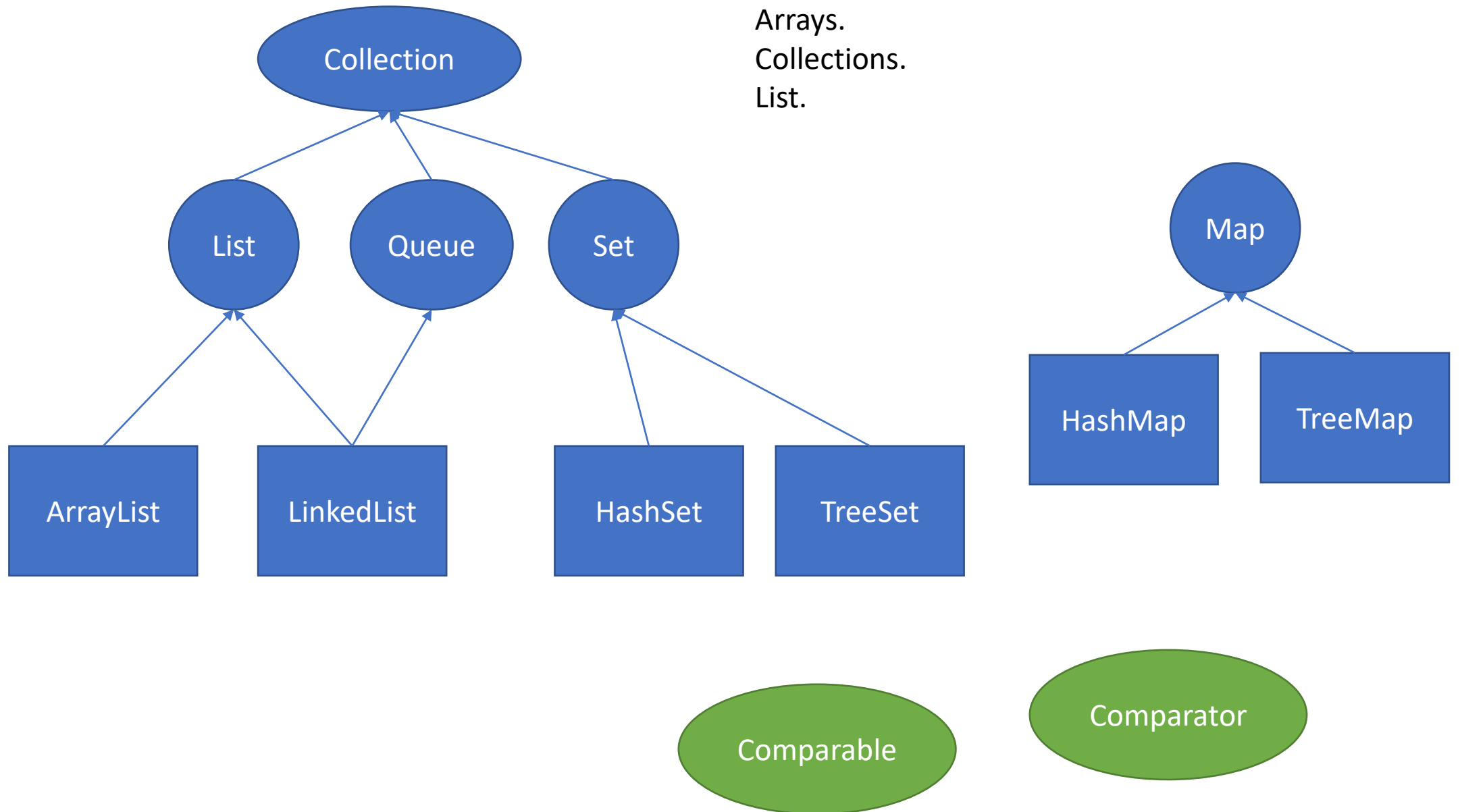


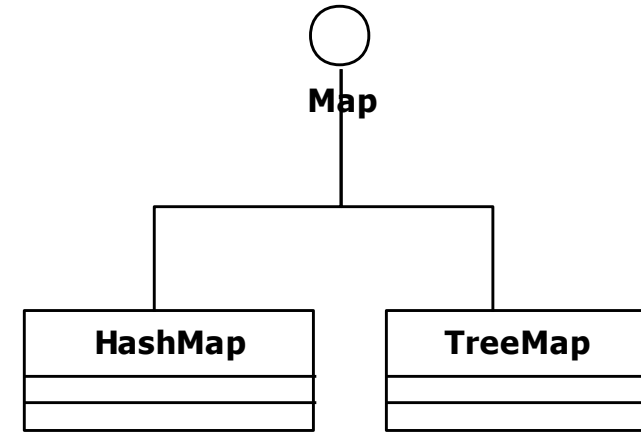
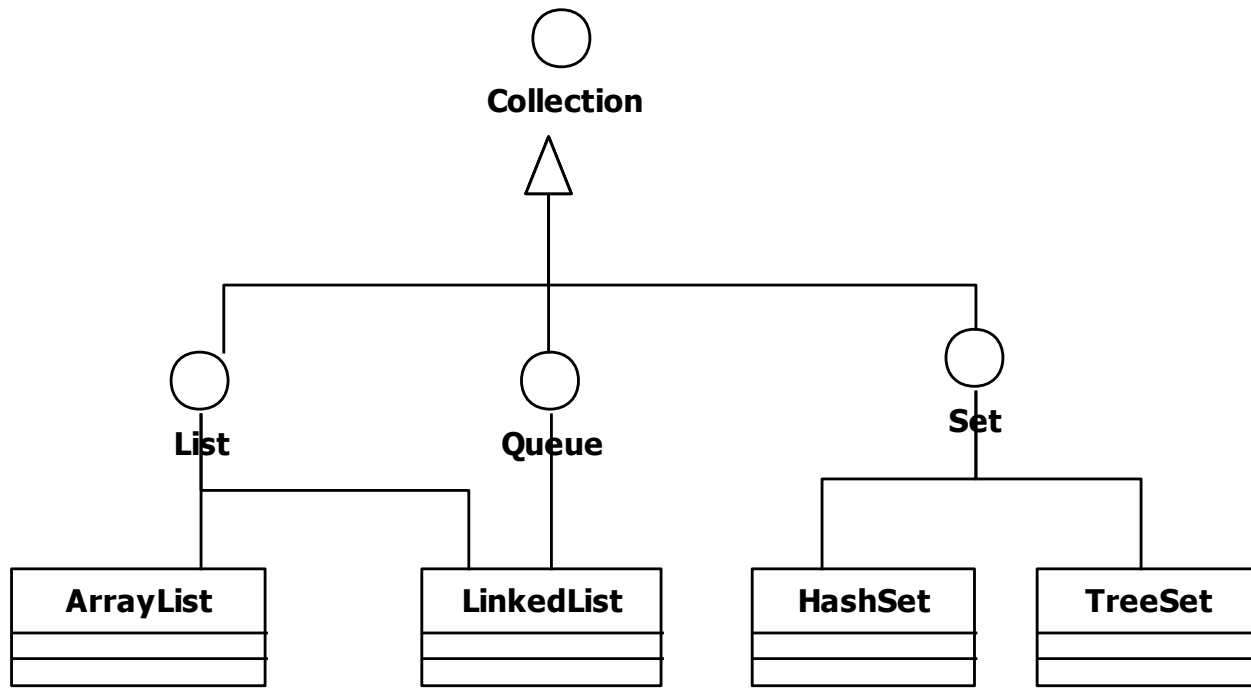
Class A



Repaso

Equals -> hashCode
Arrays.
Collections.
List.





Class A<T>

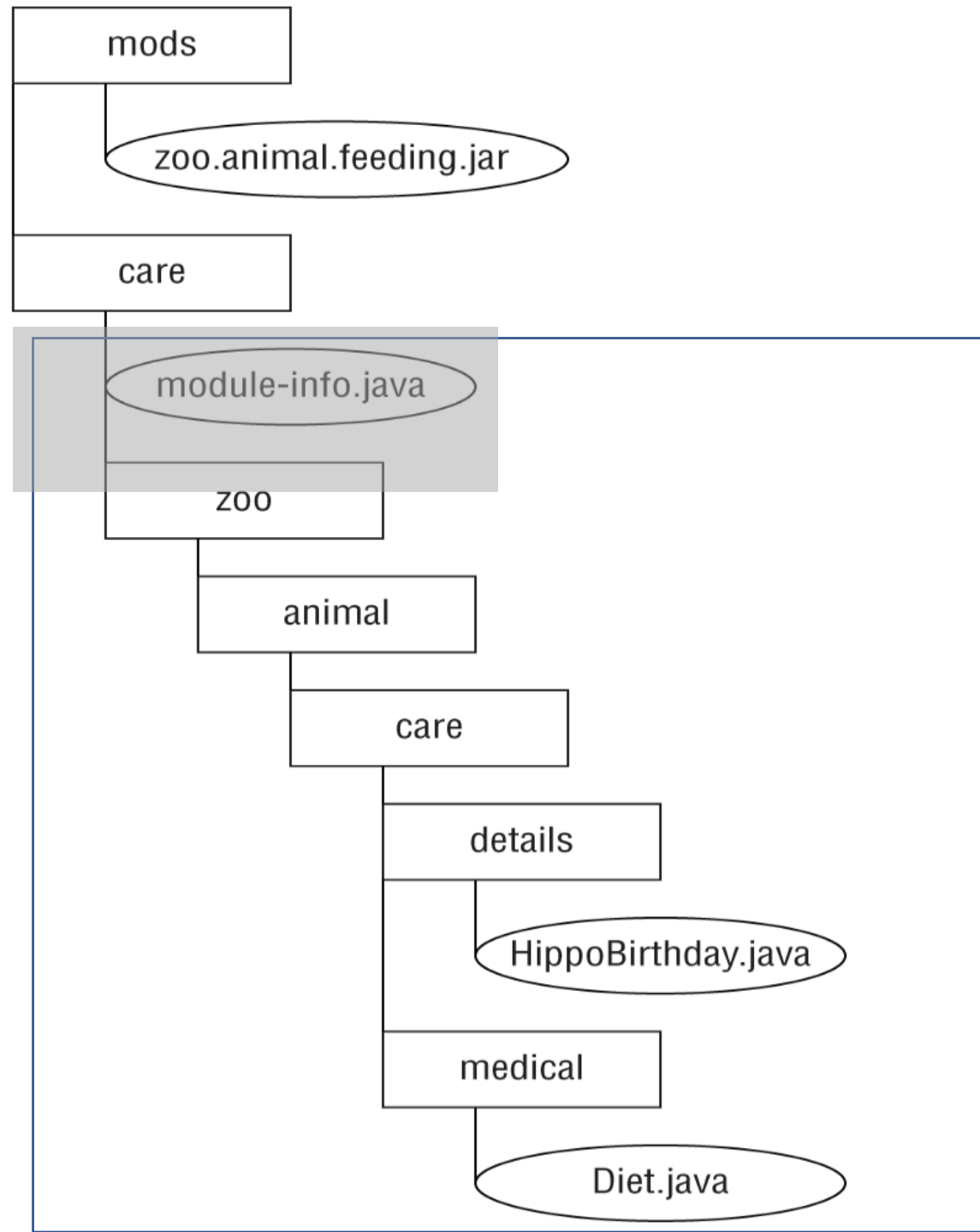
Class A<? extends String>

Class A<? super String>

public <T> T método(T x){}

List<String> lista = new ArrayList<>();

17 Modular Applications



Derivative	Description
exports <package>	Allows all modules to access the package
exports <package> to <module>	Allows a specific module to access the package
requires <module>	Indicates module is dependent on another module
requires transitive <module>	Indicates the module and that all modules that use this module are dependent on another module
uses <interface>	Indicates that a module uses a service
provides <interface> with <class>	Indicates that a module provides an implementation of a service

Module path

zoo.tickets.jar

zoo.tickets.cost

zoo.tickets.types

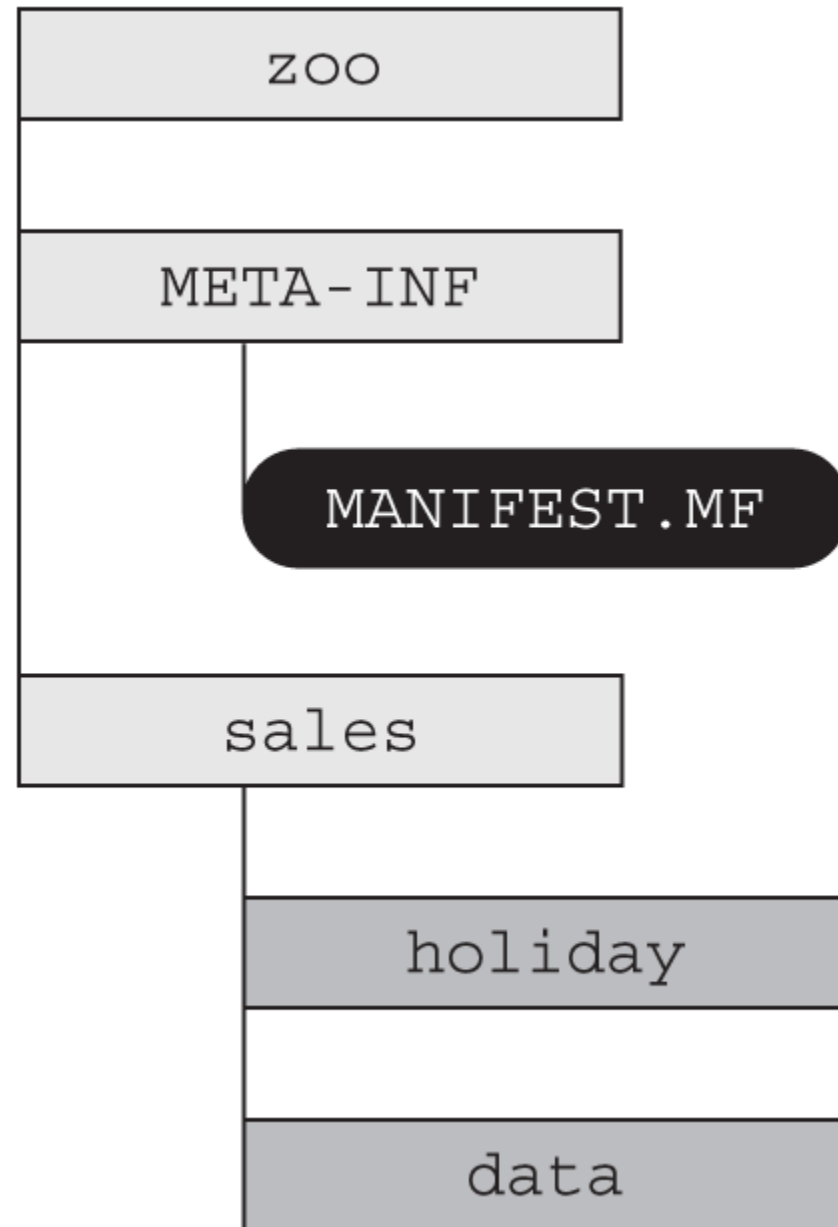
module-info.class

Module path

zoo.sales.jar

zoo.sales.holiday

zoo.sales.data



modulepath

classpath



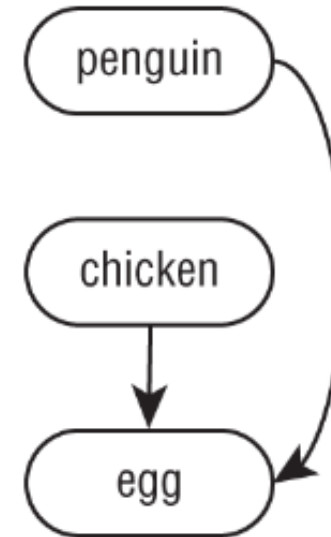
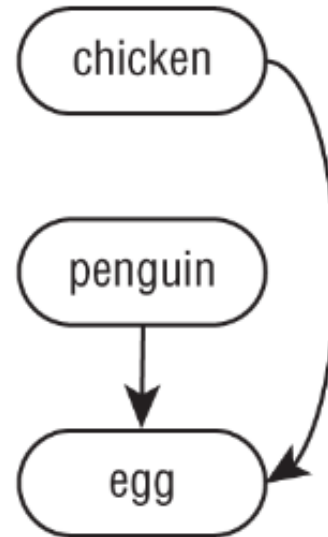
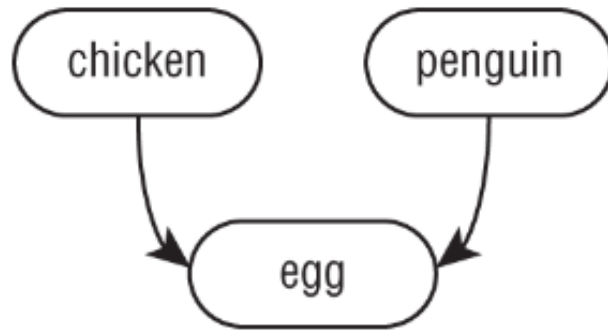
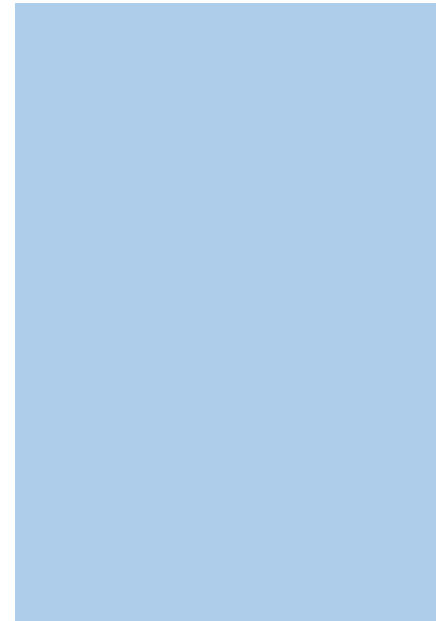
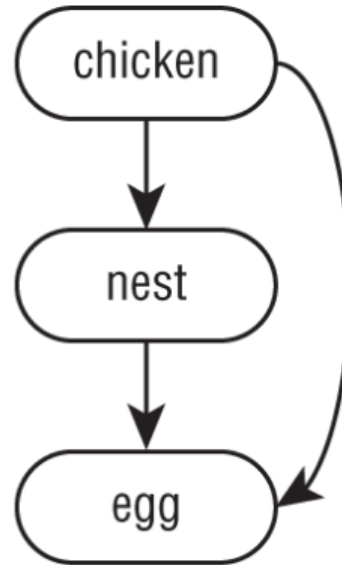
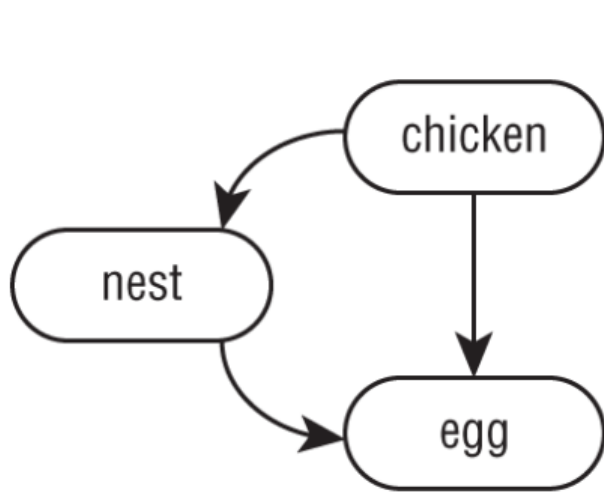
Property	Named	Automatic	Unnamed
A _____ module contains a module-info file?	Yes	No	Ignored if present
A _____ module exports which packages to other modules?	Those in the module-info file	All packages	No packages
A _____ module is readable by other modules on the module path?	Yes	Yes	No
A _____ module is readable by other JARs on the classpath?	Yes	Yes	Yes

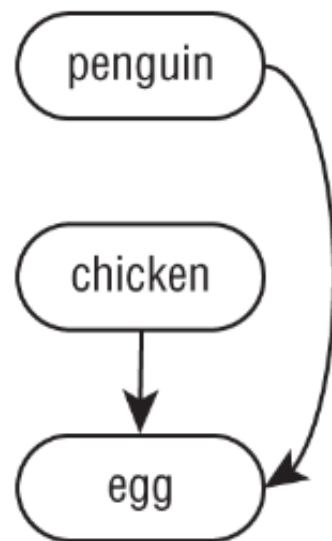
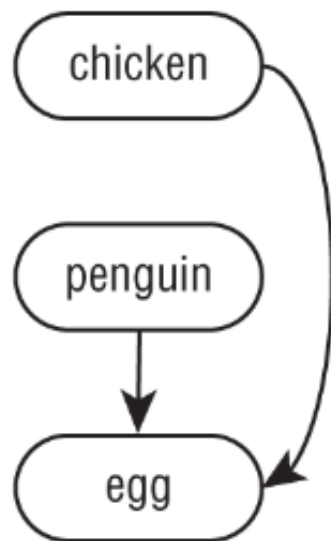
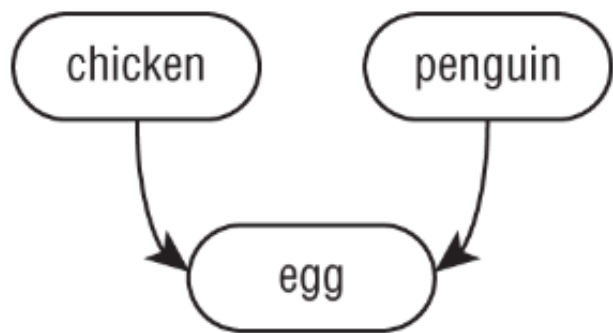
Module name	What it contains	Coverage in book
java.base	Collections, Math, IO, NIO.2, Concurrency, etc.	La mayoría se cubre
java.desktop	Abstract Windows Toolkit (AWT) and Swing	No es parte del examen
java.logging	Logging	No es parte del examen
java.sql	JDBC	Chapter 21, "JDBC"
java.xml	Extensible Markup Language (XML)	No es parte del examen

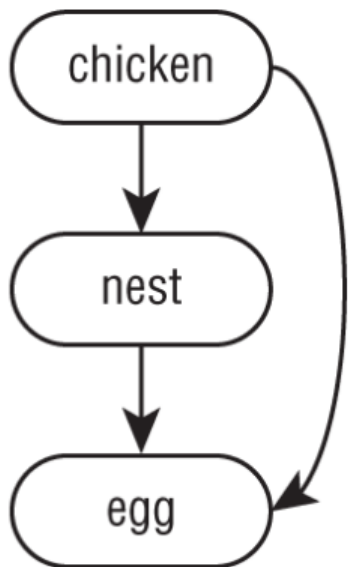
java.base	java.naming	java.smartcardio
java.compiler	java.net .http	java.sql
java.datatransfer	java.prefs	java.sql.rowset
java.desktop	java.rmi	java.transaction.xa
java.instrument	java.scripting	java.xml
java.logging	java.se	java.xml.crypto
java.management	java.security.jgss	
java.management.rmi		
java.security.sasl		



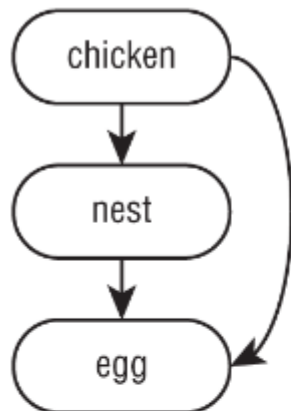
jdk.accessiblity	jdk.jconsole	jdk.naming.dns
jdk.attach	jdk.jdeps	jdk.naming.rmi
jdk.charsets	jdk.jdi	jdk.net
jdk.compiler	jdk.jdwp.agent	jdk.pack
jdk.crypto.cryptoki	jdk.jfr	jdk.rmic
jdk.crypto.ec	jdk.jlink	jdk.scripting.nashorn
jdk.dynalink	jdk.jshell	jdk.sctp
jdk.editpad	jdk.jsobject	jdk.security.auth
jdk.hotspot.agent	jdk.jstatd	jdk.security.jgss
jdk.httpserver	jdk.localdata	jdk.xml.dom
jdk.jartool	jdk.management	jdk.zipfs
jdk.javadoc	jdk.management.agent	
jdk.jcmd	jdk.management.jfr	







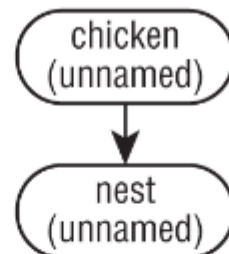
① classpath



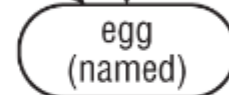
module path

②

classpath



module path

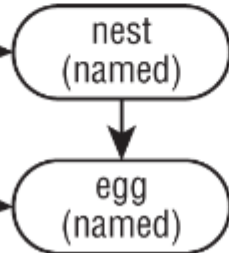


③

classpath



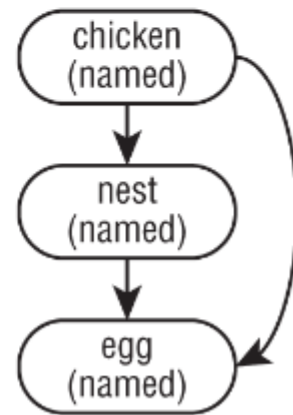
module path

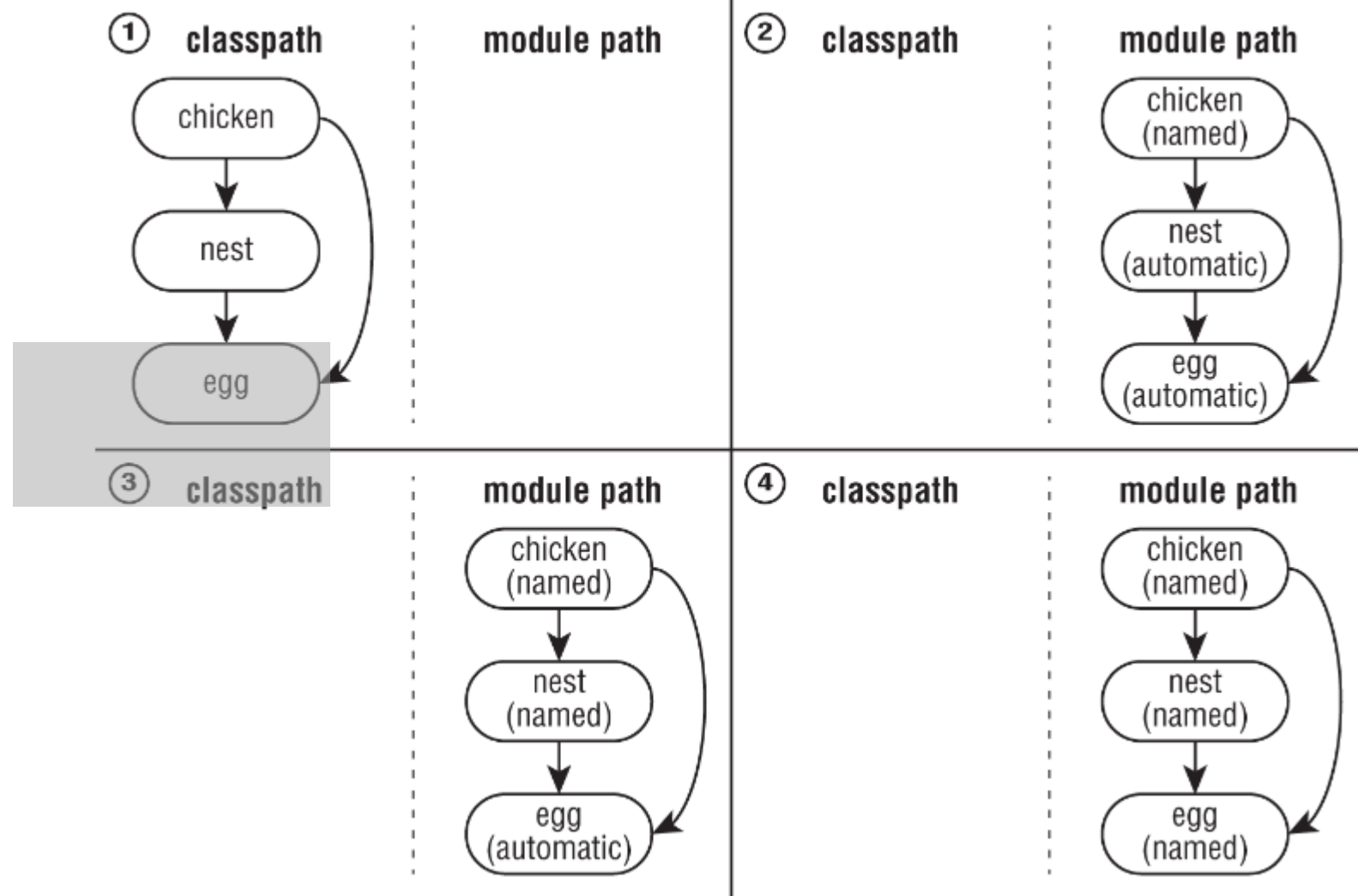
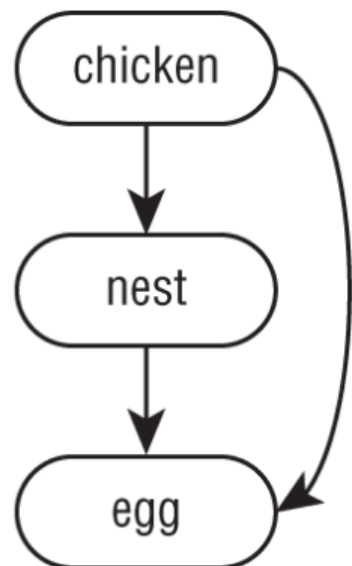


④

classpath

module path

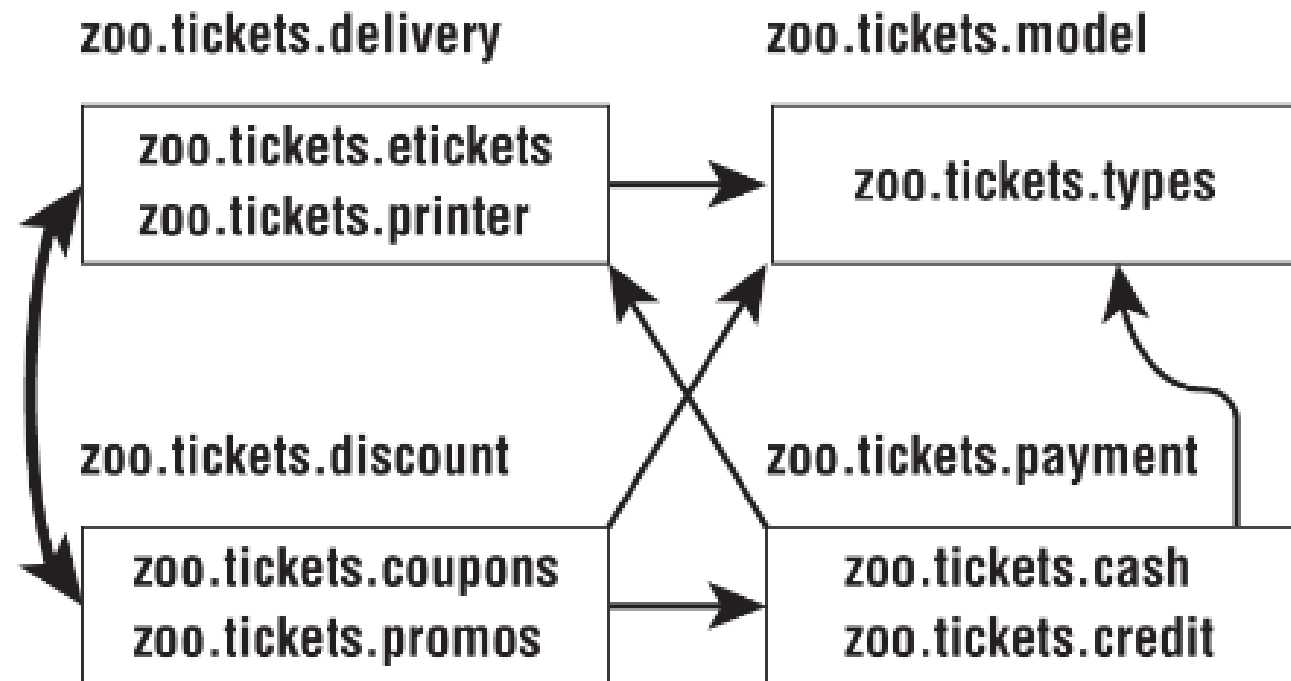




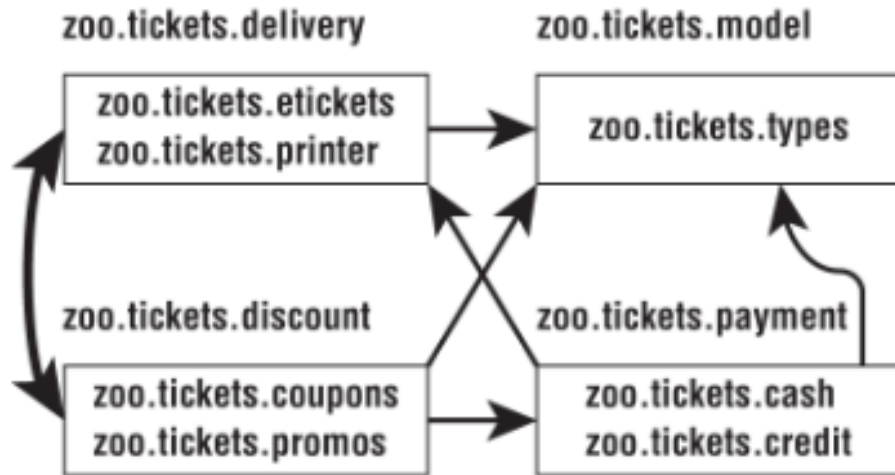
Before

zoo.tickets.cash
zoo.tickets.coupons
zoo.tickets.credit
zoo.tickets.etickets
zoo.tickets.promos
zoo.tickets.printer
zoo.tickets.types

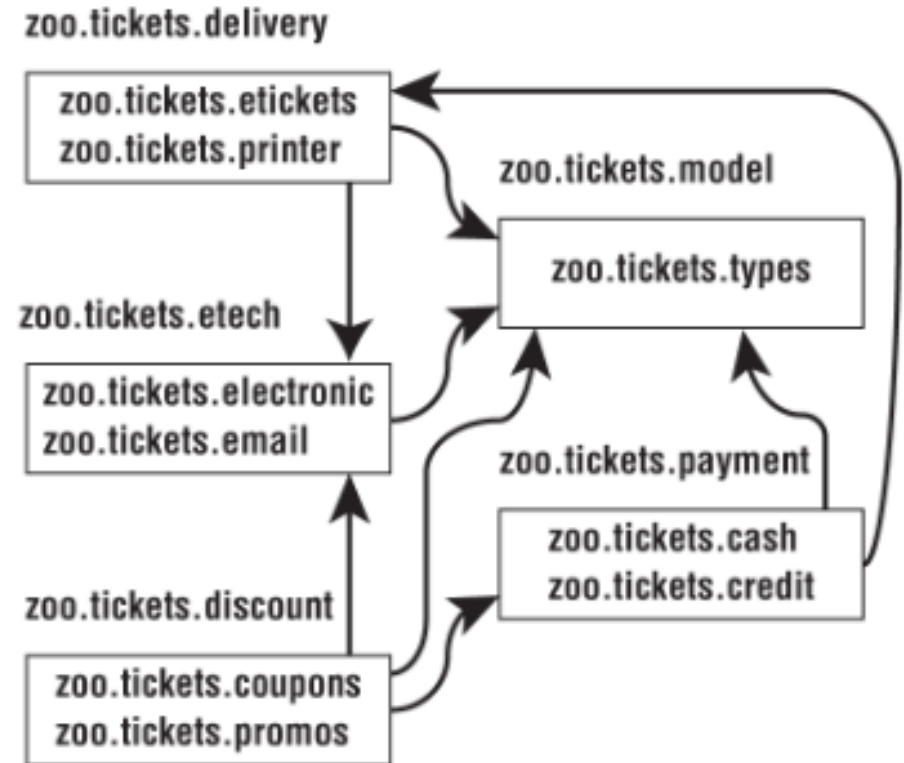
After



Before



After



Modulo: zoo.tours.api

Notificador



```
module zoo.tours.api {  
  exports zoo.tours.api;  
}
```

Modulo: tours.reservations

Locator

```
module zoo.tours.reservations {  
  exports zoo.tours.reservations;  
  requires zoo.tours.api;  
  uses zoo.tours.api.Tour;  
}
```

Modulo: zoo.visitor

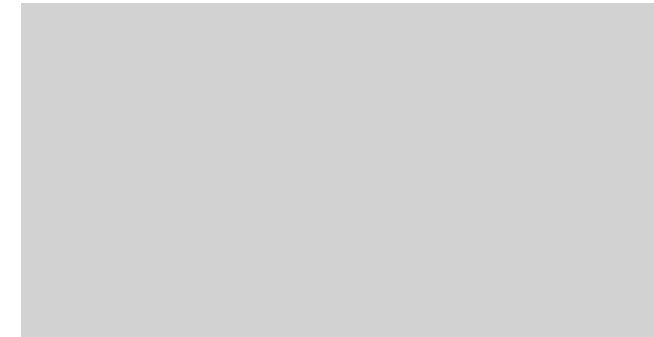
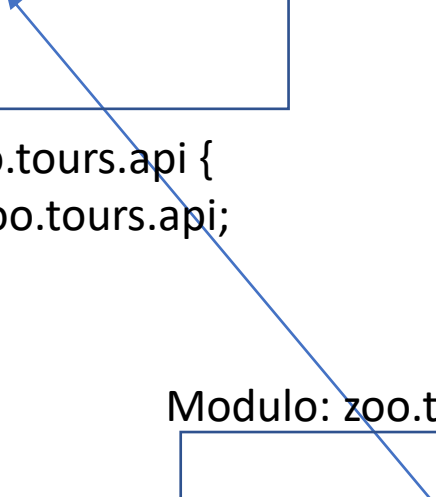
Consumer

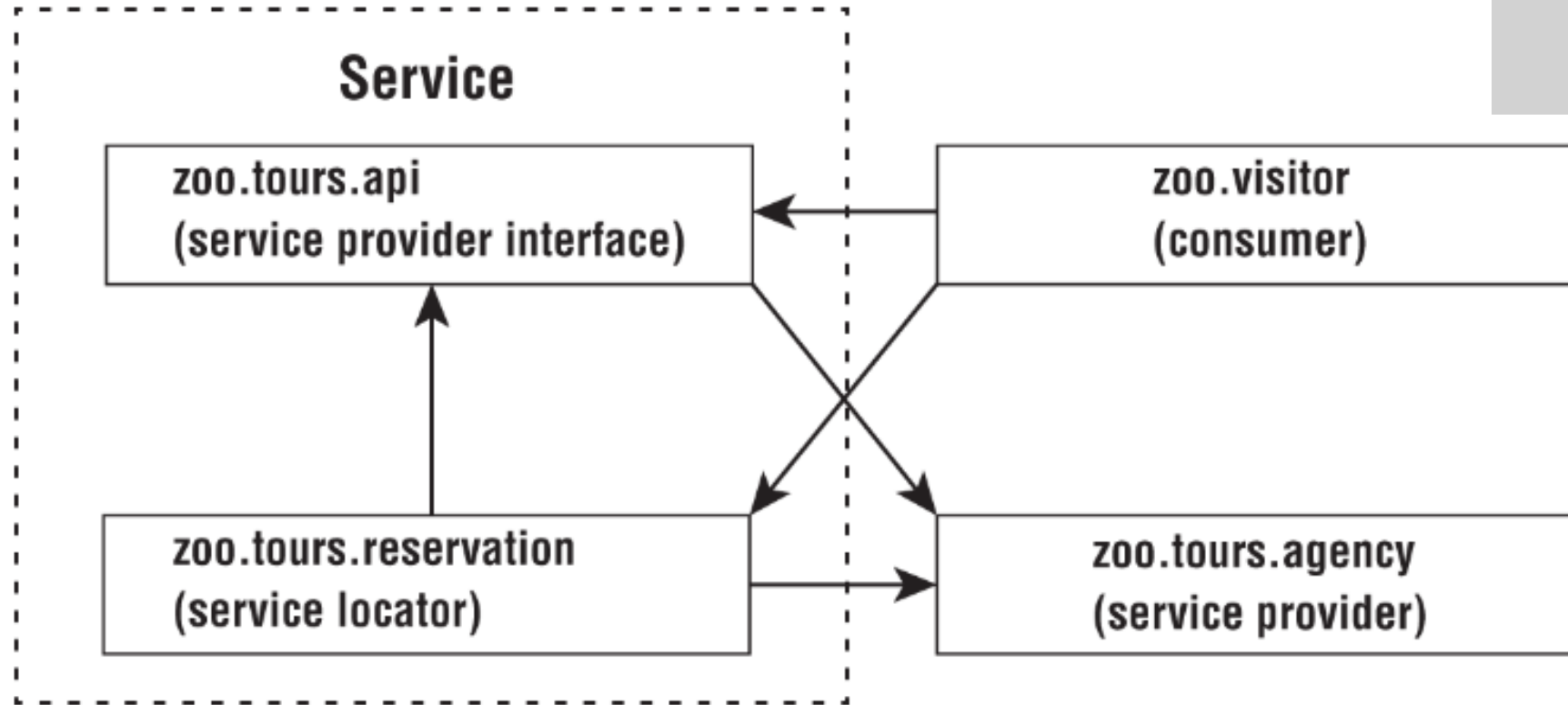
```
module zoo.visitor {  
  requires zoo.tours.api;  
  requires zoo.tours.reservations;  
}
```

Modulo: zoo.tours.agency

NotificadorEmail

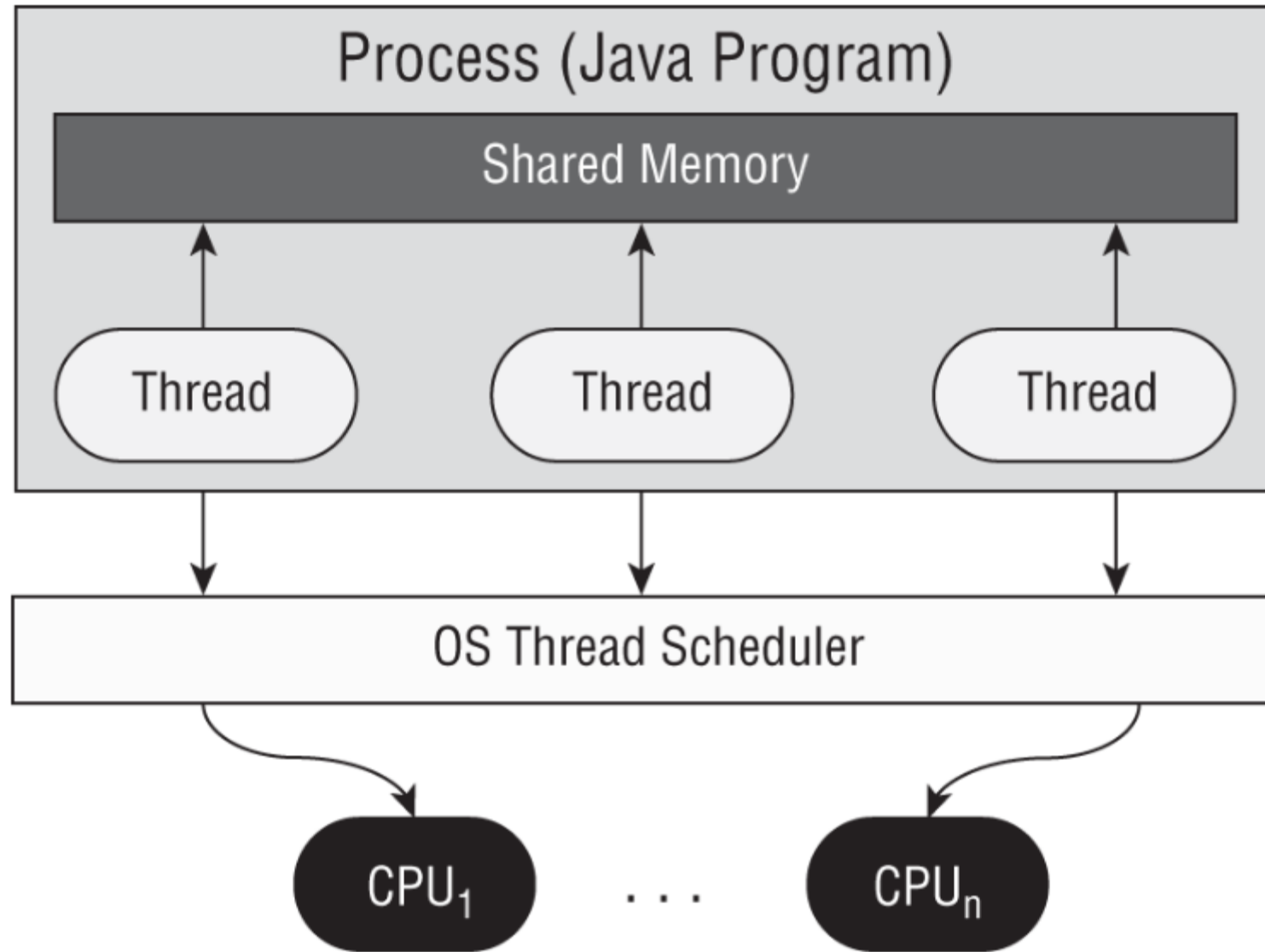
```
module zoo.tours.agency {  
  requires zoo.tours.api;  
  provides zoo.tours.api.Tour with zoo.tours.agency.TourImpl;  
}
```



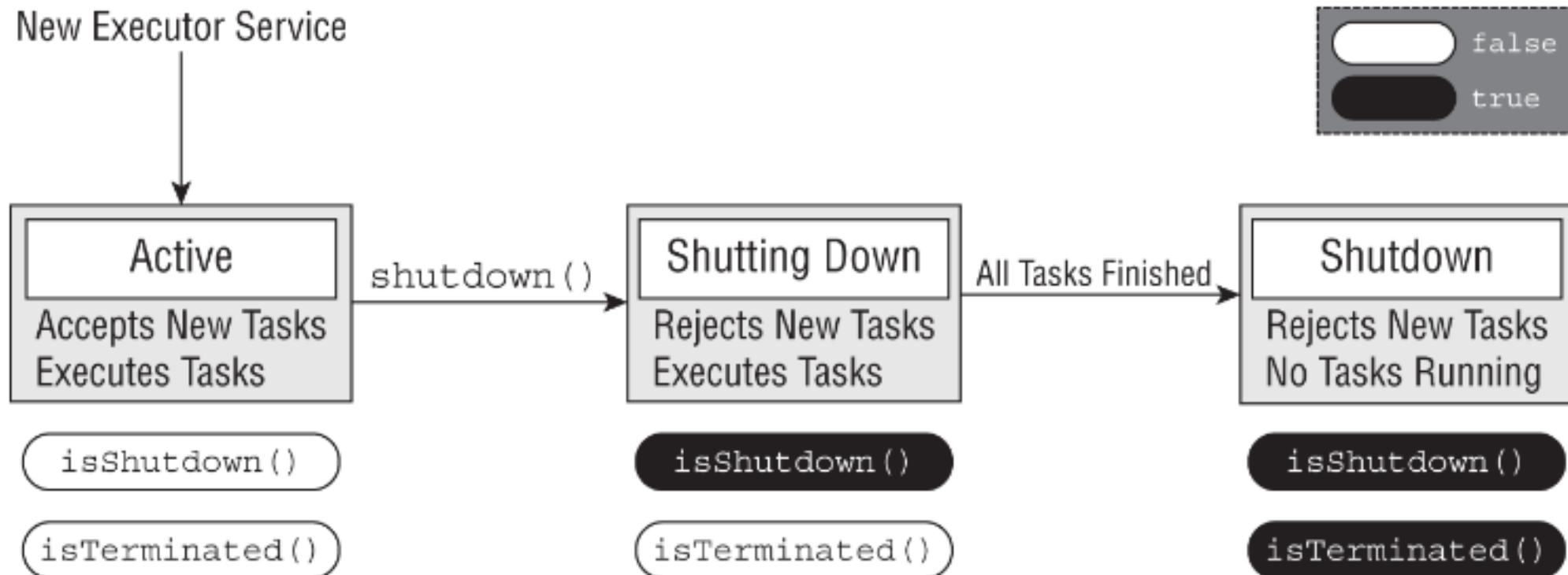


Artifact	Part of the service	Directives required in <code>module-info.java</code>
Service provider interface	Yes	<code>exports</code>
Service provider	No	<code>requires</code> <code>provides</code>
Service locator	Yes	<code>exports</code> <code>requires</code> <code>uses</code>
Consumer	No	<code>requires</code>

Ch18 - Concurrency



New Executor Service



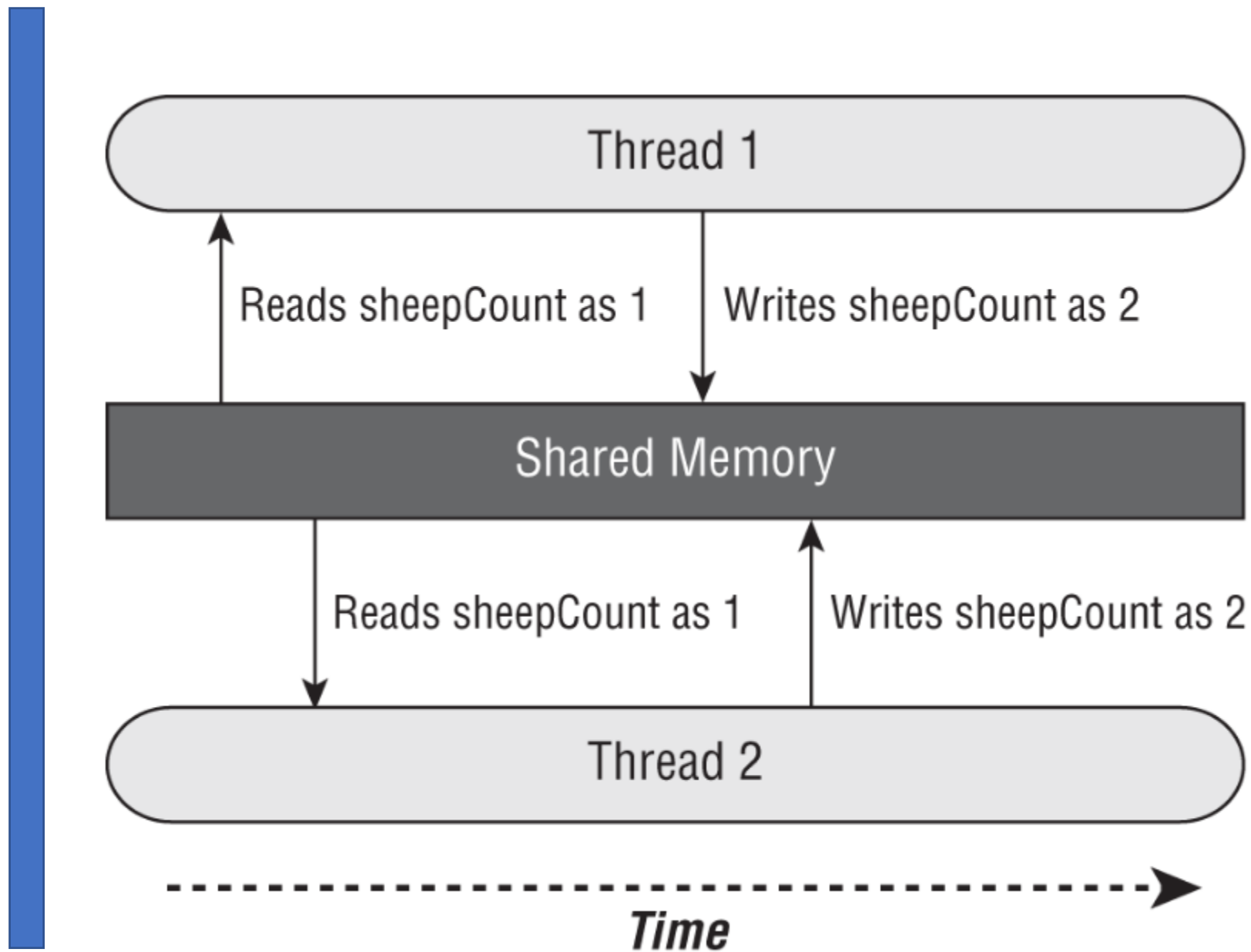
Method name	Description
void execute(Runnable command)	Executes a Runnable task at some point in the future
Future<?> submit(Runnable task)	Executes a Runnable task at some point in the future and returns a Future
<T> Future<T> submit(Callable<T> task)	Executes a Callable task at some point in the future and returns a Future representing the pending results of the task
<T> List<Future<T>> invokeAll(Collection<? extends Callable<T>> tasks) throws InterruptedException	Executes the given tasks and waits for all tasks to complete. Returns a List of Future instances, in the same order they were in the original collection
<T> T invokeAny(Collection<? extends Callable<T>> tasks) throws InterruptedException, ExecutionException	Executes the given tasks and waits for at least one to complete. Returns a Future instance for a complete task and cancels any unfinished tasks

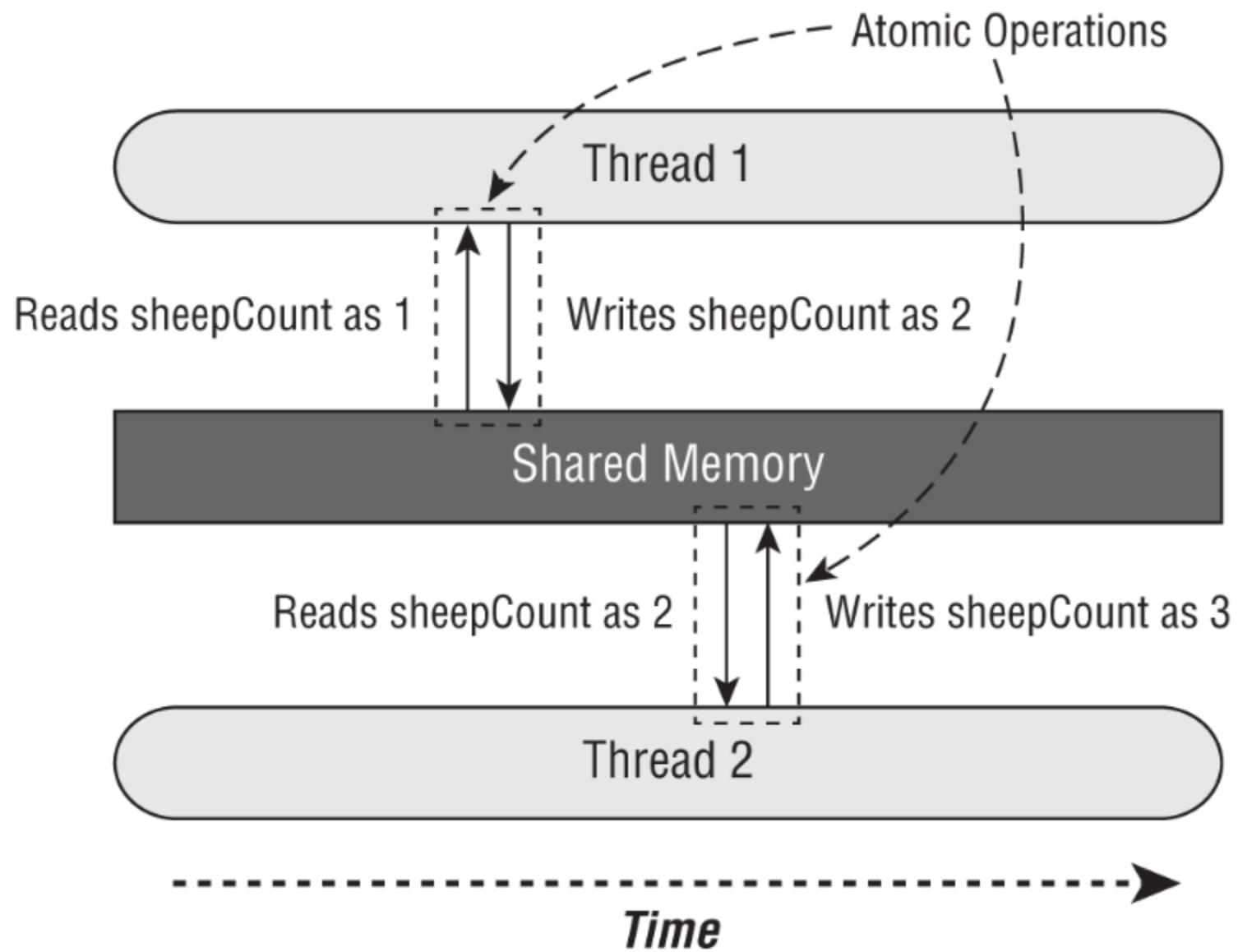
Method name	Description
<code>boolean isDone()</code>	Returns true if the task was completed, threw an exception, or was cancelled
<code>boolean isCancelled()</code>	Returns true if the task was cancelled before it completed normally
<code>boolean cancel(boolean mayInterruptIfRunning)</code>	Attempts to cancel execution of the task and returns true if it was successfully cancelled or false if it could not be cancelled or is complete
<code>V get()</code>	Retrieves the result of a task, waiting endlessly if it is not yet available
<code>V get(long timeout, TimeUnit unit)</code>	Retrieves the result of a task, waiting the specified amount of time. If the result is not ready by the time the timeout is reached, a checked <code>TimeoutException</code> will be thrown.

Enum name	Description
TimeUnit.NANOSECONDS	Time in one-billionth of a second (1/1,000,000,000)
TimeUnit.MICROSECONDS	Time in one-millionth of a second (1/1,000,000)
TimeUnit.MILLISECONDS	Time in one-thousandth of a second (1/1,000)
TimeUnit.SECONDS	Time in seconds
TimeUnit.MINUTES	Time in minutes
TimeUnit.HOURS	Time in hours
TimeUnit.DAYS	Time in days

Method Name	Description
<code>schedule(Callable<V> callable, long delay, TimeUnit unit)</code>	Creates and executes a Callable task after the given delay
<code>schedule(Runnable command, long delay, TimeUnit unit)</code>	Creates and executes a Runnable task after the given delay
<code>scheduleAtFixedRate(Runnable command, long initialDelay, long period, TimeUnit unit)</code>	Creates and executes a Runnable task after the given initial delay, creating a new task every period value that passes
<code>scheduleWithFixedDelay(Runnable command, long initialDelay, long delay, TimeUnit unit)</code>	Creates and executes a Runnable task after the given initial delay and subsequently with the given delay between the termination of one execution and the commencement of the next

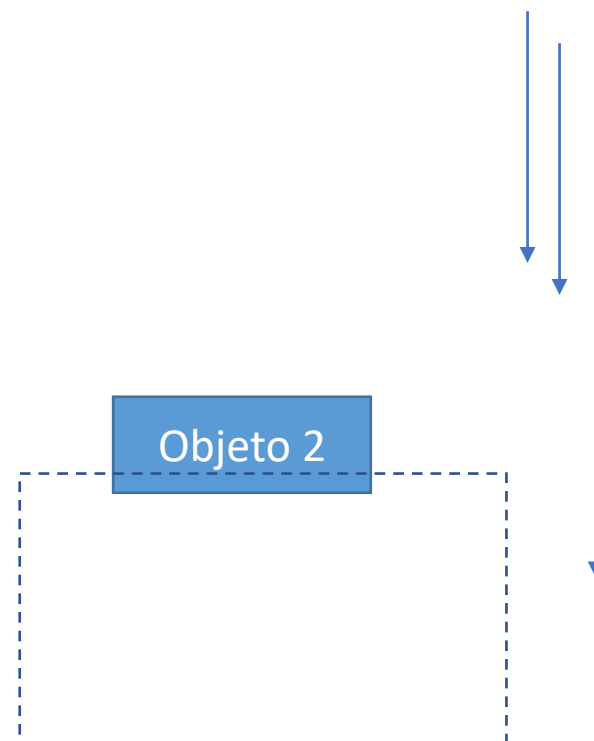
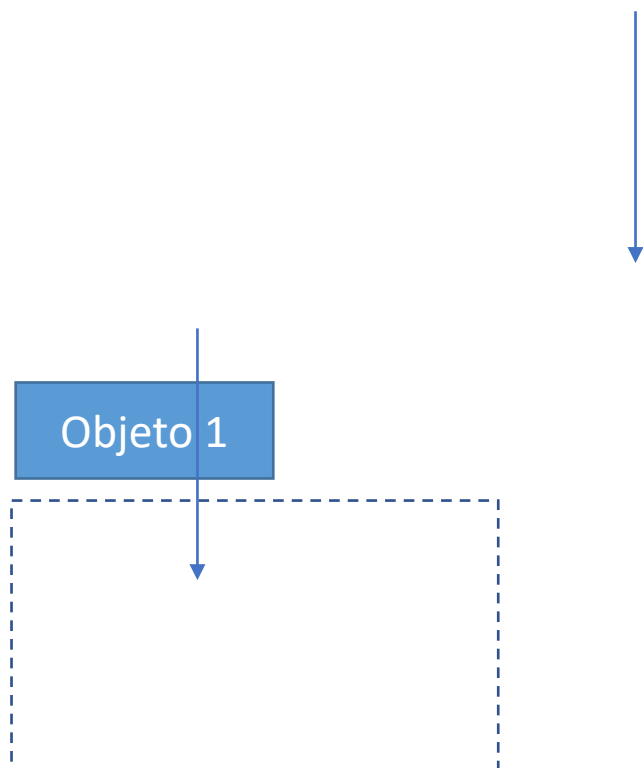
Method	Description
ExecutorService newSingleThreadExecutor()	Creates a single-threaded executor that uses a single worker thread operating off an unbounded queue. Results are processed sequentially in the order in which they are submitted.
ScheduledExecutorService newSingleThreadScheduledExecutor()	Creates a single-threaded executor that can schedule commands to run after a given delay or to execute periodically
ExecutorService newCachedThreadPool()	Creates a thread pool that creates new threads as needed but will reuse previously constructed threads when they are available
ExecutorService newFixedThreadPool(int)	Creates a thread pool that reuses a fixed number of threads operating off a shared unbounded queue
ScheduledExecutorService newScheduledThreadPool(int)	Creates a thread pool that can schedule commands to run after a given delay or to execute periodically





Class Name	Description
AtomicBoolean	A boolean value that may be updated atomically
AtomicInteger	An int value that may be updated atomically
AtomicLong	A long value that may be updated atomically

Method name	Description
<code>get()</code>	Retrieves the current value
<code>set()</code>	Sets the given value, equivalent to the assignment = operator
<code>getAndSet()</code>	Atomically sets the new value and returns the old value
<code>incrementAndGet()</code>	For numeric classes, atomic pre-increment operation equivalent to ++value
<code>getAndIncrement()</code>	For numeric classes, atomic post-increment operation equivalent to value++
<code>decrementAndGet()</code>	For numeric classes, atomic pre-decrement operation equivalent to --value
<code>getAndDecrement()</code>	For numeric classes, atomic post-decrement operation equivalent to value--



Class name	Java Collections Framework interfaces	Elements ordered?	Sorted?	Blocking?
ConcurrentHashMap	ConcurrentMap	No	No	No
ConcurrentLinkedQueue	Queue	Yes	No	No
ConcurrentSkipListMap	ConcurrentMap SortedMap NavigableMap	Yes	Yes	No
ConcurrentSkipListSet	SortedSet NavigableSet	Yes	Yes	No
CopyOnWriteArrayList	List	Yes	No	No
CopyOnWriteArraySet	Set	No	No	No
LinkedBlockingQueue	BlockingQueue	Yes	No	Yes

