Java Semaphore: Mediating Access to Shared Resources

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Module

• Understand the concept of semaphores
• Be aware of the two types of semaphores
• Note a human known use of semaphores
• Recognize the structure & functionality of Java Semaphore
• Know the key methods defined by the Java Semaphore class
• Learn how Java semaphores enable multiple threads to
  • Mediate access to a limited # of shared resources
Applying a Java Semaphore to Mediate Access
Applying a Java Semaphore to Mediate Access

- This Android app shows how a Java semaphore can be used to limit the number of Middle-Earth beings who can gaze into Palantiri concurrently.

Each being is implemented to run in a separate thread.

See en.wikipedia.org/wiki/Palantir
Applying a Java Semaphore to Mediate Access

- This Android app shows how an Java semaphore can be used to limit the # of Middle-Earth beings who can gaze into Palantiri concurrently
- The app can be configured to restrict the # of being threads that concurrently gaze into palantiri

*e.g., limit to two palantiri on a quad-core device to ensure system responsiveness*
Applying a Java Semaphore to Mediate Access

• This Android app shows how an Java semaphore can be used to limit the # of Middle-Earth beings who can gaze into Palantiri concurrently

• The app can be configured to restrict the # of being threads that concurrently gaze into palantiri

• A permit must be acquired from a semaphore before a being can gaze

Acquiring a permit atomically decrements the permit count
Applying a Java Semaphore to Mediate Access

- This Android app shows how a Java semaphore can be used to limit the number of Middle-Earth beings who can gaze into Palantiri concurrently.
- The app can be configured to restrict the number of being threads that concurrently gaze into Palantiri.
- A permit must be acquired from a semaphore before a being can gaze.

All available permits are now in use.
Applying a Java Semaphore to Mediate Access

• This Android app shows how an Java semaphore can be used to limit the # of Middle-Earth beings who can gaze into Palantiri concurrently

• The app can be configured to restrict the # of being threads that concurrently gaze into palantiri

• A permit must be acquired from a semaphore before a being can gaze

• Other being threads must block until a permit is available
Applying a Java Semaphore to Mediate Access

• This Android app shows how a Java semaphore can be used to limit the number of Middle-Earth beings who can gaze into Palantiri concurrently.
  
  • The app can be configured to restrict the number of being threads that concurrently gaze into palantiri.
  
  • A permit must be acquired from a semaphore before a being can gaze.
  
  • Other being threads must block until a permit is available.
  
  • When a being thread is done gazing it releases the semaphore.
Applying a Java Semaphore to Mediate Access

- This Android app shows how a Java semaphore can be used to limit the number of Middle-Earth beings who can gaze into Palantiri concurrently.
  - The app can be configured to restrict the number of being threads that concurrently gaze into Palantiri.
  - A permit must be acquired from a semaphore before a being can gaze.
  - Other being threads must block until a permit is available.
    - When a being thread is done gazing, it releases the semaphore.
    - Another being thread can then acquire it and proceed to gaze.
Applying a Java Semaphore to Mediate Access

- This Android app shows how a Java semaphore can be used to limit the number of Middle-Earth beings who can gaze into Palantiri concurrently.

- The app can be configured to restrict the number of being threads that concurrently gaze into palantiri.

- A permit must be acquired from a semaphore before a being can gaze.

- Other being threads must block until a permit is available.

  - When a being thread is done gazing, it releases the semaphore.
  - Another being thread can then acquire it and proceed to gaze.

This example “fully brackets” the acquiring & releasing of permits, i.e., the thread that acquires a semaphore is the same as the one that releases it.
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```
: Palantiri
    Presenter

: BeingRunnables

: Palantir

: mPalantiriManager
    : PalantiriManager

start()
    start()
    start()

run()
    r.gaze()
    release(r)
    r = acquire()

run()
    r.gaze()
    release(r)
    r = acquire()

run()
    r.gaze()
    release(r)
    r = acquire()

run()
    r.gaze()
    release(r)
    r = acquire()
```
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```
: Palantiri
Presenter

start()
```
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

1. : Palantiri
   Presenter

2. : BeingRunnables

```
start()
start()
start()
```

```
start()
start()
start()
```
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

![UML Sequence Diagram]

- Palantiri Presenter
- BeingRunnables
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```
start()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
run()
p = acquire()
```

```
Applying a Java Semaphore to Mediate Access

• UML sequence diagram for this app

start()  \rightarrow \text{Palantiri Presenter}  \rightarrow \text{BeingRunnable}

mPalantiriManager : PalantiriManager

start()  \rightarrow \text{Palantiri Presenter}  \rightarrow \text{BeingRunnable}

start()  \rightarrow \text{Palantiri Presenter}  \rightarrow \text{BeingRunnable}

run()  \rightarrow \text{Palantiri Presenter}  \rightarrow \text{BeingRunnable}

p = acquire()  \rightarrow \text{Palantiri Presenter}  \rightarrow \text{BeingRunnable}

p = acquire()  \rightarrow \text{Palantiri Presenter}  \rightarrow \text{BeingRunnable}

p = acquire()  \rightarrow \text{Palantiri Presenter}  \rightarrow \text{BeingRunnable}
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```
start()  start()  start()  start()

run()    p = acquire()    p = acquire()    p = acquire()

p.gaze() p.gaze() p.gaze()
```

```
: PalantiriPresenter
: BeingRunnables
p : Palantir
mPalantiriManager : PalantiriManager
```
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```
start()
start()
start()

run()
p = acquire()
p.gaze()
release(p)
p = acquire()
p.gaze()
release(p)
p = acquire()
p.gaze()
release(p)
```
Applying a Java Semaphore to Mediate Access

• UML sequence diagram for this app

```
start()
start()
start()
run()
p = acquire()
p.gaze()
run()
p = acquire()
p.gaze()
run()
p = acquire()
p.gaze()
run()
p = acquire()
p.gaze()
run()
p = acquire()
p.gaze()
release(p)
release(p)
release(p)
release(p)
```

: Palantiri Presenter

: BeingRunnables

p : Palantir

mPalantiriManager : PalantiriManager
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```
start()
run()
start()  
run()  
p.gaze()
start()
run()  
p.gaze()
start()  
run()  
p.gaze()
start()  
run()  
p.gaze()
start()  
run()  
p.gaze()
start()  
run()  
p.gaze()
start()  
run()  
p.gaze()
```

: Palantiri Presenter ➔ :

: BeingRunnables ➔ ➔ ➔ ➔

p : Palantir

mPalantiriManager : PalantiriManager

Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```
start()
run()
start()  
run()  
p.gaze()
start()
run()  
p.gaze()
start()  
run()  
p.gaze()
start()  
run()  
p.gaze()
start()  
run()  
p.gaze()
start()  
run()  
p.gaze()
```
End of Java Semaphore: Mediating Access to Shared Resources