Mediating Access to Shared Resources via Java Semaphore

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 an 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](en.wikipedia.org/wiki/Palantir)
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.

  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 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.

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 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.
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 Palantirë concurrently.

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

- 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 show 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
  • When a being thread is done it gazing it releases the semaphore
  • Another being thread can then acquire it & 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

```
start() \rightarrow run()
start() \rightarrow run()
start() \rightarrow run()
```

```
mPalantiriManager : PalantiriManager
: Palantir
: BeingRunnables

PalantiriPresenter
```

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

- UML sequence diagram for this app

```
start()

: Palantiri
Presenter
```
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

![UML sequence diagram]

```
Applying a Java Semaphore to Mediate Access
```

```
start()
```

```
: Palantiri Presenter →
```

```
: BeingRunnables
```

start()
start()
start()
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

: Palantiri
Presenter

: BeingRunnables

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

: Palantiri Presenter → : BeingRunnables
mPalantiriManager : PalantiriManager

start()
start()
start()
run()
p = acquire()
p = acquire()
p = acquire()
run()
run()
run()
```
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

.start() : Palantiri Presenter

.start() : BeingRunnables

p : Palantir

mPalantiriManager : PalantiriManager

run()

p.gaze()

p = acquire()

run()

p.gaze()

p = acquire()

run()

p = acquire()
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

- Applying a Java Semaphore to Mediate Access

- Palantiri Presenter

- BeingRunnables

- Palantir

- PalantiriManager

- BeingRunnables

- PalantiriPresenter

- PalantiriManager

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

- UML sequence diagram for this app

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

: Palantiri Presenter
: BeingRunnables
p : Palantir
mPalantiriManager : PalantiriManager

Palantiri
Palantiri
Palantiri
Presenter
PalantiriManager
Palantiri
Presenter
PalantiriManager
Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```
start()  -->  start()  -->  start()  -->  start()
         |                      |                      |
         v                      v                      v
         run()  -->  run()  -->  run()  -->  run()
             |                      |                      |
             v                      v                      v
             p.gaze()  -->  p.gaze()  -->  p.gaze()  -->  p.gaze()
                 |                      |                      |
                 v                      v                      v
                 release(p)  -->  release(p)  -->  release(p)  -->
```

- **Applying a Java Semaphore to Mediate Access**
  - **mPalantiriManager**
  - **PalantiriManager**
  - **Palantiri**
  - **PalantiriPresenter**
  - **BeingRunnables**

- **UML sequence diagram for this app**
  - ```
    start()  -->  start()  -->  start()  -->  start()
    run()  -->  run()  -->  run()  -->  run()
    p = acquire()  -->  p = acquire()  -->  p = acquire()
    p.gaze()  -->  p.gaze()  -->  p.gaze()  -->  p.gaze()
    release(p)  -->  release(p)  -->  release(p)  -->
  ```
Applying a Java Semaphore to Mediate Access

UML sequence diagram for this app

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

- UML sequence diagram for this app

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

: Palantiri
Presenter →

: BeingRunnables →

p : Palantir

mPalantiriManager : PalantiriManager

Applying a Java Semaphore to Mediate Access

- UML sequence diagram for this app

```java
start()
run()
start()
start()
p = acquire()
p.gaze()
run()
release(p)
p = acquire()
p.gaze()
run()
release(p)
p = acquire()
p.gaze()
run()
release(p)
p = acquire()
p.gaze()
run()
release(p)
p = acquire()
p.gaze()
run()
release(p)
p = acquire()
p.gaze()
run()
release(p)
p = acquire()
p.gaze()
run()
release(p)
```
End of Mediating Access to Shared Resources via Java Semaphore