Navy

Please do not use indigo to run your concurrent code. Instead please use navy.cse.yorku.ca.

A Counter

```
Counter : monitor
begin
  value : int;
  procedure increment(result number : int)
  begin
    value := value + 1;
    number := value;
  end
  procedure decrement(result number : int)
  begin
    value := value - 1;
    number := value;
  end
  value := 0;
end
```

Monitors in Java

monitor	class
variable	attribute
procedure	synchronized method
initialization	constructor

Counter Class

Problem

Implement the class `Counter` with

- attribute `value`,
- initialized to zero, and
- the methods `increment` and `decrement`.

Synchronized Methods

A lock is associated with every object. For threads to execute a synchronized method on such the object, first its lock needs to be acquired.

A Resource

```
Resource : monitor
begin
  available : boolean;
  free : condition;
  procedure acquire()
  begin
    if (not available) free.wait;
    available := false;
  end
  procedure release()
  begin
    available = true;
    free.signal;
  end
  available := true;
end
```

Resource Class

Problem

Implement the class `Resource` with

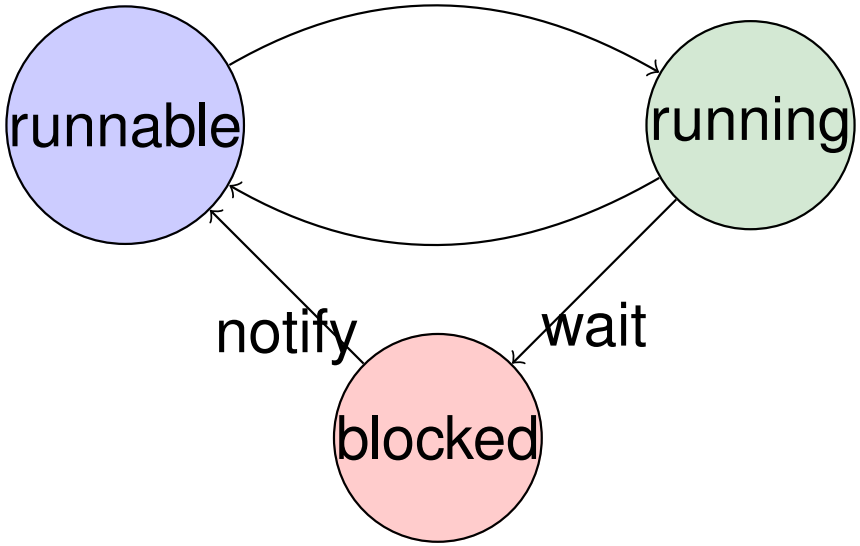
- attribute `available`,
- initialized to true, and
- the methods `acquire` and `release`.

Wait and Notify

The Object class contains the following three methods:

- `wait`: causes the current thread to wait until another thread wakes it up.
- `notify`: wakes up a single thread waiting on this object's lock; if there is more than one waiting, an arbitrary one is chosen; if there are none, nothing is done.
- `notifyAll`: wakes up all threads waiting on this objects lock.

Since every class extends the class Object, these methods are available to every object.


runnable
running
notify
wait
blocked