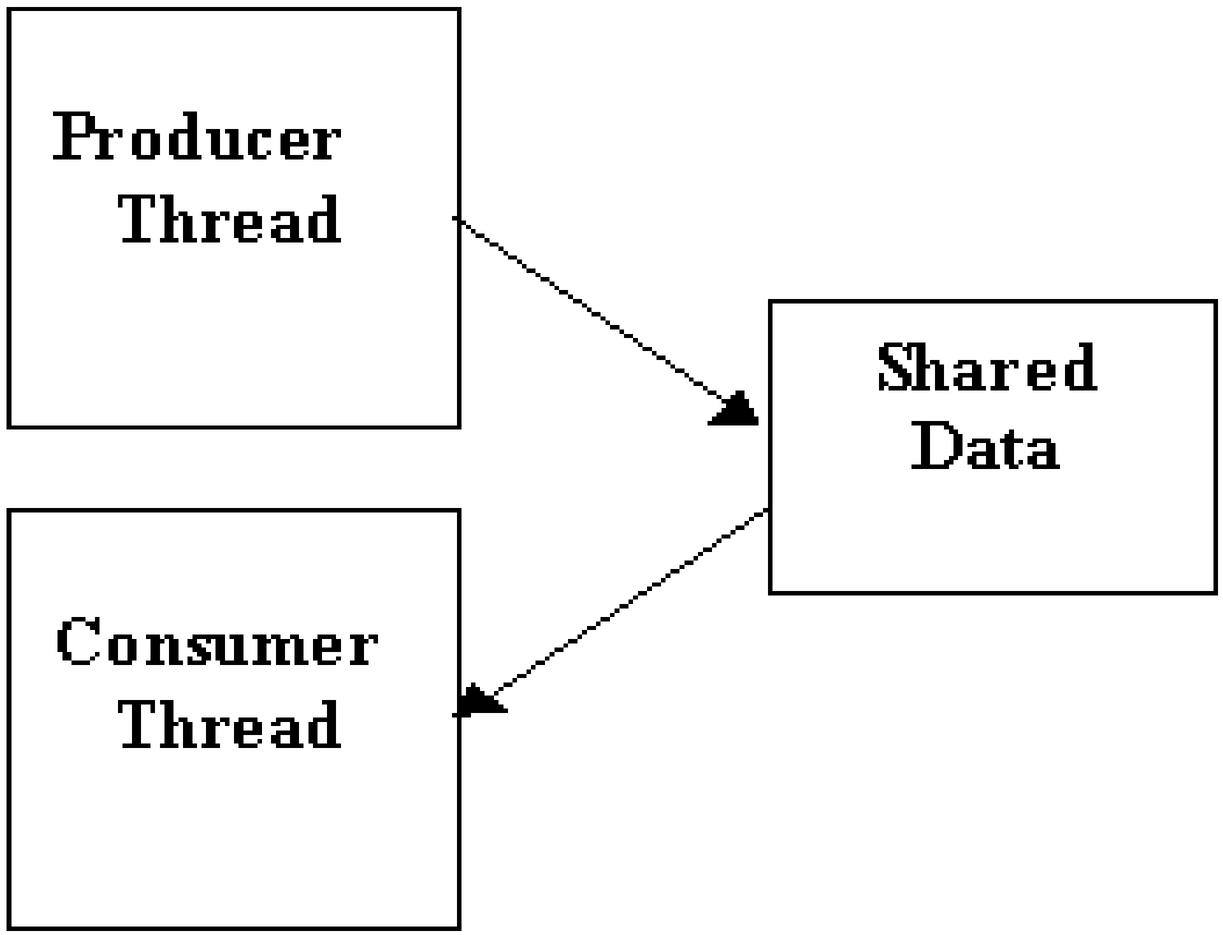
6. Java Threads

Thread Synchronization



Inter Thread Communication

- Java includes an elegant inter-process communication mechanism via the wait(), notify(), and notifyAll() methods. Also they can share data between each other for communication purpose

Thread Control Methods

- wait() tells the calling thread to give up the monitor and go to sleep until some other thread enters the same monitor and calls notify().
- notify() wakes up the one thread that called wait() on the same object (if there are more than one thread in wait state then it is not known that which thread will wake up.
- notifyAll() wakes up all the threads that called wait() on the same object.

Thread Control Methods

- Defined in the Object class
- Should compulsorily be defined within a synchronized block
- Replace **notify()** with **notifyAll()** to notify all the waiting threads on this monitor

wait() and notify()

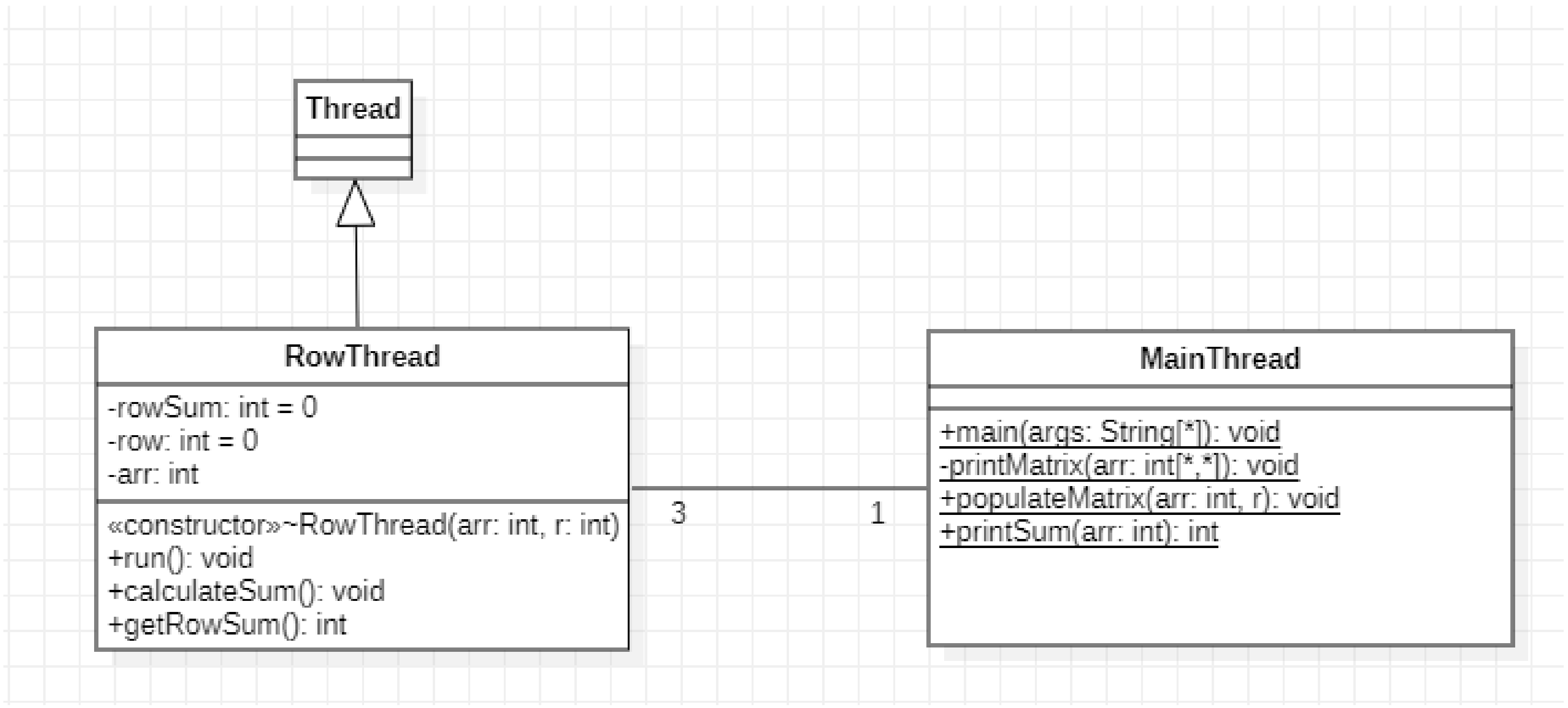
```
public synchronized String retrieveMessage() {
    while(request = = false) {
            try{
                        wait();
            } catch(Interrupted Exception e){}
    }
    request = false;
    notify()
    return message;
}
```

wait() tells the calling thread to give up the monitor and go to wait state until some other thread enters the same monitor and calls notify().

Lab - Assignment

- Assume an integer matrix having 3 rows and 4 columns. Populate the matrix with a sequence of random numbers ranging from 1 to 100. Create three threads where each thread calculates the sum of elements of one distinct row. The main thread should calculate the total sum of the matrix elements by using the partial sums from these threads. Ensure that the main thread finishes last. Use a UML class diagram to depict the classes and their relationships used in this program.

Lab - Assignment


Thread
RowThread
-rowSum: int = 0
-row: int = 0
-arr: int
«constructor»~RowThread(arr: int, r: int)
+run(): void
+calculateSum(): void
+getRowSum(): int
3
1
MainThread
+main(args: String[*]): void
-printMatrix(arr: int[*,*]): void
+populateMatrix(arr: int, r): void
+printSum(arr: int): int