

Assignment 2 (5 points)
Due 21.07.2010
Exercise 1: Telegraph (3 points)

There are two threads, sender and receiver, that communicate using telegrams. Implement the `Telegraph` class (use synchronization with `wait/notify/notifyAll`) to ensure that the receiver blocks until a telegram has been sent and the sender blocks until the previous telegram has been received.

```
public interface ITelegraph
{
    String take();
    void put(String telegram);
}
```

Implement a sender thread, which generates a series of telegrams and telegraphs them using the `put` method. The string "DONE" indicates that all telegrams have been sent. To simulate the unpredictable nature of real-world applications, the sender thread pauses for random intervals between telegrams. Implement a receiver thread, which simply receives the telegrams using the `take` method and prints them out until it receives the "DONE" string. This thread also pauses for random intervals.

Exercise 2: Reentrant Lock (2 points)

Implement a class `ReentrantLock` providing a reentrant lock with the same semantics as built-in Java synchronized locks: once a thread has a lock, it can re-obtain it any number of times without blocking. The lock is made available to other threads when as many releases as acquires have occurred. Consider the following interface:

```
public interface IReentrantLock
{
    void acquire() throws InterruptedException;
    void release() throws IllegalMonitorStateException;
}
```

The `acquire()` method acquires the lock and throws `InterruptedException` if the thread was interrupted. The `release()` method releases the lock and throws an `IllegalMonitorStateException` if the current thread is not the holder of the lock.

Submission instructions

The assignment must be submitted to me before Wednesday 21st, 10.00 by e-mail at

`vaide.narvaez@gmail.com` and `vaide_n@payap.ac.th`. You have to submit one zip file that contains all your java source files. Late submissions are penalized 5% of the final grade each late day!

This is an individual assignment, plagiarism is not tolerated.