APD - Lab4

Tuesday, October 23, 2007
12:18 PM

synchronized == regiune critica
fiecare obiect are asociat un lock (zavor)

```
class X
{
	void f()
	{
		synchronized
		{

		}
	}
}
```

primul thread care intra si incearca sa execute codul din "synchronized" pune lock-ul
al thread nu poate executa threadul pana cand lock-ul nu a fost scos

```
class X
{
	public synchronized void f()
	{

	}

	public synchronized void g()
	{

	}
}
```

declararea unei functii sincronizate

daca avem

X x1,x2;

x1.f()	x2.f()
x1.f()	x1.g()

(situatii blocante)

zavoarele sunt unice pentru obiecte

diferenta dintre functii sincronizate si regiuni de sincronizare este momentul cand se seteaza lock-ul

Clasa Vector are metode de set...() si get()... sincronizate pentru a permite sa lucreze cu obiectele unui singur thread la un moment dat.

public static synchronized void f() == sincronizare globala

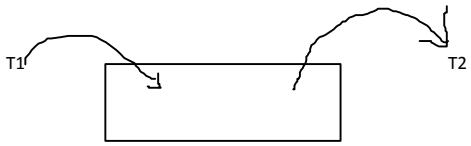
```
synchronized(o)
{
	//metode de accesare?
}
```

setarea lock-ului pentru un anumit obiect

Wait si Notify

busy waiting (astepta un eveniment verificand la anumite intervale de timp)

T2, cand asteapta: o.wait()
T1, cand anunta pe T1: o.notify()



Thread1

```
synchronized(o)
{
	o.wait();
}
```

Thread2

```
synchronized(o)
{
	o.notify();
}
```

la apelare wait() threadul se adauga la multimea waitset (?)
la wait() lock-ul este scos

celelalte threaduri concureaza pentru preluarea zavorului

o.notifyAll() == "trezeste" toate threadurile

stari threaduri:
- creat
- gata de executie
- executie
- blocat
- terminat

```
while(buffer gol)
{
	o.wait();
}
```

daca bufferul e gol, threadul a fost "trezit" din greseala,
deci asteapta in continuare

ReentrantLock	l =	new ReentrantLock()

l.lock()

l.lock()	l.unlock()	l.trylock()

```
try{
        //cod regiune critica
}

finally{
        l.unlock(); (poate sa returneste IllegalMonitorStateException daca threadul nu detine lock-ul)
}
```

clasa: ReentrantReadWriteLock cu lock-uri pentru citire si scriere
permite mai multe threaduri care citesc in acelasi timp dar nu mai multe care scriu in acelasi timp