

Note sullo svolgimento della prova

- Non è consentito l'uso di vecchi testi d'esame, libri, o appunti.
- Utilizzare NetBeans per editare, compilare ed eseguire i programmi.
- Creare un file **Cognome_matricola.zip** contenente i soli file *.java* da consegnare.
- Al termine della prova, dopo il caricamento degli elaborati, il docente mostrerà una possibile soluzione. Dopo aver visto la soluzione, gli studenti avranno la possibilità di riprendere il proprio elaborato.

Esercizio

La memoria di un calcolatore ha una dimensione di m byte e viene gestita a blocchi di n byte. m è un multiplo di n . Ogni volta che un processo ha bisogno di memoria la chiede al sistema operativo; quando il processo non ne ha più bisogno la restituisce al sistema operativo. La memoria assegnata a un processo non può essere assegnata a un altro processo. I processi sono divisi in due categorie: quelli con priorità e quelli senza. Nel caso in cui non ci sia memoria libera, le eventuali nuove richieste di processi con priorità vengono soddisfatte prima di quelle dei processi senza priorità. È possibile installare nuova memoria.

Realizzare una classe *GestoreMemoria* dotata almeno dei seguenti costruttori e metodi:

- *GestoreMemoria(int m, int n)*: crea un gestore che gestisce una memoria dalle caratteristiche specificate. Se i parametri sono non validi lancia *IllegalArgumentException*.
- *Blocco acquisisci(boolean prio)*: invocato dai client per ottenere un blocco di memoria; l'argomento indica se il processo è prioritario o meno; il metodo è bloccante nel caso in cui al momento non siano disponibili blocchi di memoria.
- *void free(Blocco b)*: invocato dai processi per liberare un blocco.
- *void installa(int k)*: vengono installati k byte di memoria, con k multiplo di n . Lancia *IllegalArgumentException* se i parametri non sono validi.

Realizzare una classe *Blocco* che soddisfi gli scopi indicati in precedenza. Realizzare una classe *Processo* che

- prende un blocco dal gestore;
- dorme per una quantità di tempo casuale;
- libera la memoria presa in precedenza.

La classe *Processo* permette di specificare se il processo è prioritario o meno.