NON SPEGNERE IL PC A FINE ESAME

Corso di Sistemi Operativi e Reti

Prova scritta di LUGLIO 2018

ISTRUZIONI

1. **Rinomina** la cartella chiamata "CognomeNomeMatricola" che hai trovato sul Desktop e in cui hai trovato questa traccia, sostituendo "Cognome" "Nome" e "Matricola" con i tuoi dati personali;
2. **Carica** tutto il materiale didattico che vorrai usare sul Desktop; puoi farlo solo nei primi 5 minuti della prova;
3. **Svolgi** il compito; lascia tutto il sorgente che hai prodotto nella cartella di cui al punto 1;
4. Quando hai finito lascia la postazione facendo logout,

senza spegnere il PC.

SALVA SPESSO il tuo lavoro

NON SPEGNERE IL PC A FINE ESAME

ESERCIZIO 1 (Programmazione multithread. Punti: 0-20)

Si progetti una struttura dati thread-safe chiamata `MappaDelMalandrino`. Tale struttura dati registra e aggiorna la posizione di un certo numero di oggetti di tipo `Punto` su una mappa di NxN posizioni. Un `Punto` può essere aggiunto o cancellato dalla mappa, e può essere monitorato secondo le specifiche di cui sotto. Un punto ha come attributi una label, e le coordinate x e y.

I metodi pubblici disponibili dovranno essere i seguenti:

`void addPoint(Point P)` . Aggiunge il punto `P` alla mappa.

`void delPoint(Point P)` . Rimuove il punto `P` dalla mappa.

`void update(P,int DX, int DY)` . Sposta il punto `P` di `DX` posizioni a destra e di `DY` posizioni in basso sulla mappa. `DX` e `DY` possono essere eventualmente negativi, in tal caso i movimenti sono da intendersi rispettivamente a sinistra e in alto; non è possibile specificare valori di `DX` e `DY` che portino fuori dalla mappa: in tal caso deve essere generata una eccezione.

`Condition addWatch(Point P1, Point P2)`. Crea e restituisce una nuova `Condition`. Tale nuova condition deve essere segnalata allorquando `P1` e `P2` dovessero trovarsi nello stesso punto sulla mappa in seguito a una modifica della loro posizione. Un thread che volesse monitorare la sovrapposizione di P1 e P2 può dunque invocare addWatch per poi aspettare sulla condition restituita.

`void delWatch(Point P1, Point P2)`. Cessa di segnalare una eventuale condition impostata per segnalare la sovrapposizione della coppia di punti `P1` e `P2`.

`void waitFor(Point P1, Point P2)`. Si pone in attesa bloccante fino a che non avviene che i due punti `P1` e `P2` si sovrappongono sulla mappa. `waitFor` può essere implementata utilizzando `addWatch` per poi aspettare sulla condition generata.

`void waitCond(Condition c)` Si pone in attesa bloccante fino a che la condition `c`, ottenuta tramite una chiamata del metodo `addWatch`, non viene notificata.

`ArrayList<String> printMap()`. Restituisce la mappa del malandrino in forma di array di stringhe. Si assuma che ogni cella vuota sia rappresentata da un '.' e ogni cella occupata da un numero compreso tra 0,1 e 9 (1 = un punto presente, 2= due punti presenti, ..., 0= più di 9 punti presenti).

NON SPEGNERE IL PC A FINE ESAME

I metodi richiesti devono essere implementati garantendo la necessaria thread safety; non sono ammesse situazioni di deadlock; è opportuno migliorare l'accessibilità concorrente alla struttura dati ed evitare situazioni di starvation.

La validazione dei valori ammissibili di input dei metodi è a carico del candidato.

NON SPEGNERE IL PC A FINE ESAME

CI SONO DEI PUNTI AMBIGUI NELLA TRACCIA? COMPLETA TU

È parte integrante di questo esercizio completare le specifiche date nei punti non esplicitamente definiti, introducendo o estendendo tutte le strutture dati laddove si ritenga necessario, e risolvendo eventuali ambiguità.

POSSO CAMBIARE IL PROTOTIPO DEI METODI RICHIESTI? NO

Non è consentito modificare il prototipo dei metodi se questo è stato fornito. Potete aggiungere qualsivoglia campo e metodo di servizio, e qualsivoglia classe ausiliaria, ma NON variare l'interfaccia dei metodi pubblici già specificati.

CHE LINGUAGGIO DEVO USARE? JAVA 7 O SUCCESSIVO

Il linguaggio da utilizzare per l'implementazione è Java. *È* consentito usare qualsiasi funzione di libreria di Java 7 o successivi.

MA IL MAIN() LO DEVO SCRIVERE? E I THREAD DI PROVA? SOLO PER FARE IL TUO DEBUG

Non è esplicitamente richiesto di scrivere un `main()` o di implementare esplicitamente del codice di prova, anche se lo si suggerisce per testare il proprio codice prima della consegna.

ESERCIZIO 2 (Linguaggi di scripting. Punti 0-10)

Si scriva uno script in `Perl` dal nome `compare.pl` che effettui semplici operazioni di ricerca e comparazione tra files.

```
./compare.pl [options] [file1 file2]
```

Lo script può ricevere diversi argomenti/parametri al momento della sua esecuzione ed in particolare dovrà poi svolgere le seguenti operazioni:

A. Nel caso in cui l'utente inserisca l'opzione `-d`, lo script dovrà comparare riga per riga il contenuto di questi ultimi.

   Esempio: Supponendo che i file siano così formati:

   `FILE1:`
   Buona fortuna
   Accidenti

   `FILE2:`
   Buona vacanza
   Accidenti
   Divertimento

   Lo script dovrà mostrare il seguente output:

```
Differenze in riga 0
Buona fortuna
----------------
Buona vacanza

Differenze in riga 2
 <MISSING>
----------------
Divertimento
```

   Infatti alla riga `0` il `file1` e il `file2` hanno contenuti completamente differenti mentre la riga 2 non esiste nel `file1` (`<MISSING>`) ma è presente nel `file2`.

B. Nel caso in cui l'utente inserisca l'opzione `-s`, lo script dovrà contare all'interno del `file2` le occorrenze di ogni singola parola contenuta in `file1`. Infine si dovrà stampare in output, **in ordine decrescente,** la parola trovata seguita dal numero delle sue occorrente nel `file2`. **A parità di occorrenze è necessario stampare anche il termine in ordine lessicografico.**

NON SPEGNERE IL PC A FINE ESAME

Esempio: Supponendo che i file siano così formati:

FILE1:

Buona fortuna

Accidenti

FILE2:

Buona vacanza

Accidenti

Divertimento

Lo script mostrerà il seguente output:

```
Accidenti --> 1
Buona --> 1
fortuna --> 0
```

Infatti le parole contenute all'interno del `file1` sono:

`Buona, fortuna, Accidenti`

La parola fortuna non compare mai in file2, mentre le altre compaiono 1 sola volta, di conseguenza si stamperà in ordine decrescente (di valore) e lessicografico (di parola).

- **TUTTE LE ALTRE OPZIONI NON SONO AMMESSE**