Esercitazione 8
Gruppo AK

Introduzione ai thread java
mutua esclusione

Agenda

Argomenti:

- I thread in java: creazione e attivazione di threads concorrenti.
- Mutua esclusione in java: metodi synchronized

Esercizio 1 – da svolgere

- Concorrenza in Java: sincronizzazione di thread concorrenti tramite syncronized

Esercizio 2 – da svolgere

- Estensione dell'esercizio 1 con simulazione di un thread che "attende"

Esercizio 1 (1/3)

Si realizzi un programma java che simuli gli ingressi dei visitatori ad un museo.

Il museo è composto da due zone:

- Area espositiva dedicata alla **mostra** vera e propria, accessibile da parte di visitatori dietro pagamento di un biglietto ad importo fisso B
- **shopping area** (collocata al termine del percorso suggerito ai visitatori), dove è possibile acquistare gadget a vario prezzo.

Per motivi di sicurezza, non è consentito l'acceso al museo (mostra+shopping area) a più di T visitatori alla volta. Inoltre, l'accesso alla mostra è consentito solo a massimo M visitatori alla volta; con T>=M.

Esercizio 1 (2/3)

Comportamento dei visitatori:

Si supponga che ogni visitatore tenti prima di entrare alla mostra e poi, una volta concluso il percorso espositivo, possa decidere arbitrariamente di entrare nella shopping area ed eventualmente fare acquisti.

Qualora l'ingresso alla mostra non sia possibile, il visitatore dovrà stampare a video un messaggio del tipo: «Accesso alla mostra non riuscito»

Esercizio 1 (3/3)

Progettare un'applicazione java che regoli gli accessi al **museo**, nella quale:

- **i visitatori** siano rappresentati da **thread** concorrenti,
- il **museo** sia rappresentato da un **oggetto condiviso** da tutti i thread.

Il **thread iniziale** (**main**), una volta terminati tutti gli altri thread, stampa:

- **Il totale dei visitatori entrati alla mostra e il totale degli entrati alla shopping area.**
- **Il totale degli incassi della giornata (incassi dei biglietti, più incassi dalla shopping area)**
- Successivamente il main termina.

Impostazione

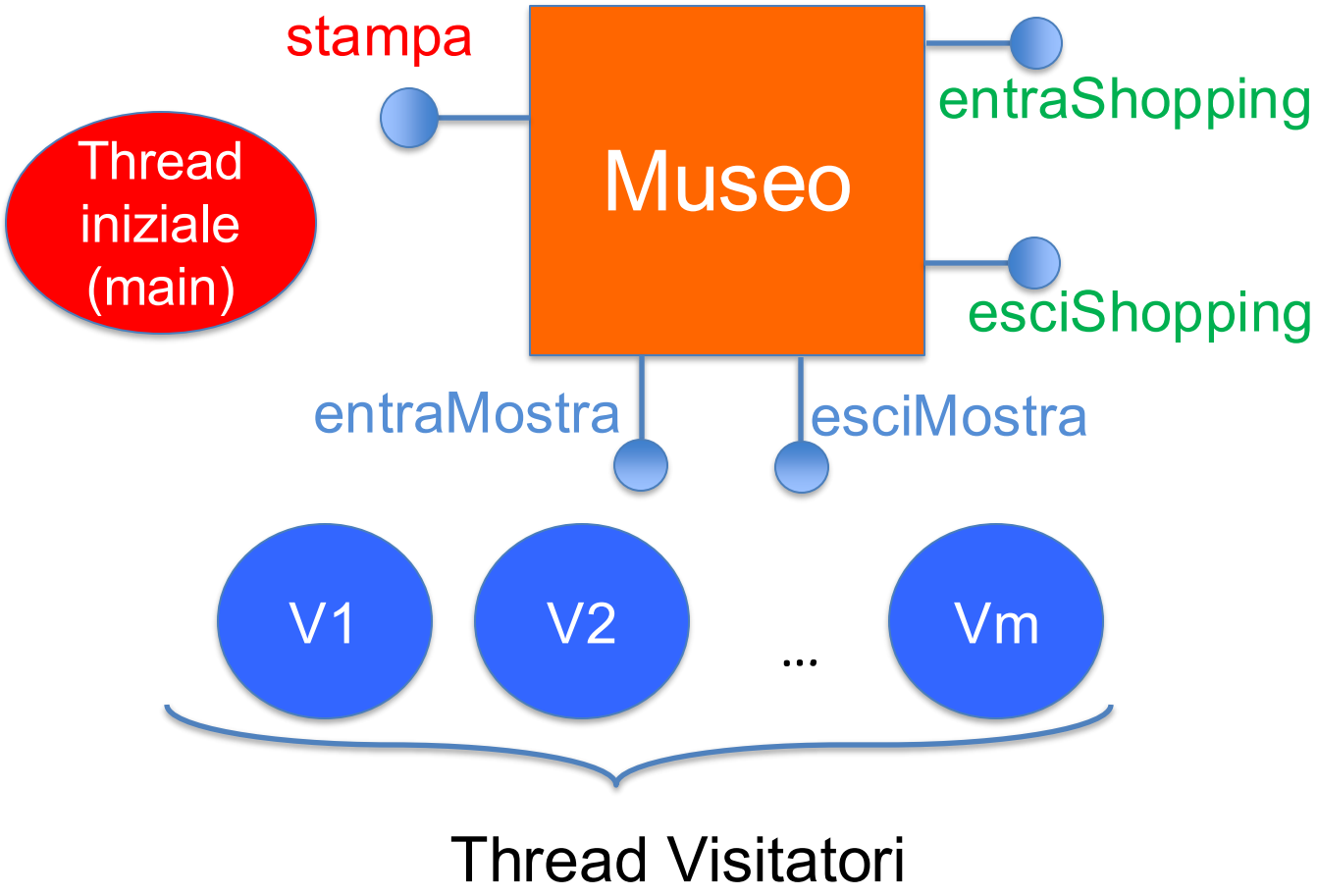
Classi da definire:

- **Museo**: è una risorsa condivisa acceduta in modo concorrente dai thread visitatori
  - Quali variabili locali?
  - Quali metodi (necessità di sincronizzazione!)?

- **Visitatore**: thread che entra prima alla mostra e poi (eventualmente) nella shopping area per fare acquisti.

- **Main:** definisce il metodo main

Impostazione



Impostazione

Classi da definire:

- **Visitatore**: il generico thread concorrente che accede al museo. Il suo comportamento è definito dal metodo run:

```
public class Visitatore <extends/implements …> {
  Museo m;
  <costruttore, etc.>
  public void run() {
     boolean entrato = m.entraMostra();
     if (entrato){
        <visita la mostra>
        m.esceMostra();
        <decide se entrare nella shopping area>
        if (ingressoShopping){
           m.entraShopping();
           <visita shopping area ed eventualmente acquista>
           m.esciShopping(spesa);}
    }else
       <stampa "Accesso alla mostra non riuscito">
}
```

Museo: è una risorsa condivisa da thread concorrenti.

-> Usiamo i metodi **Museo**.

```
public class Museo {
  // var.locali: visMostra, visTot, incassi;
  public synchronized boolean entraMostra(..){
    <verifica disponibilità posto + eventuale ingresso>
  }
  public synchronized void esciMostra(..){}
  public synchronized void entraShopping(..){}
  public synchronized void esciShopping(spesa){
    <acquisto e uscita>
  }

  public syncronized void stampa (){
    <stampa incassi e tot. ingressi>
  }
```

valore restituito:

- vero se il thread è entrato;
- falso se non è stato possibile

Classe Main: contiene il metodo main

```
import java.util.Random;
public class Main{
   <definizioni costanti ecc.>

   public static void main(String[] args) {
      <creazione Museo m>
      <creazione Visitatori>
      <attivazione thread>
      <attesa della terminazione di tutti i thread>
      m.stampa(); //stampa dei valori finali
   } }
```

Esercizio 2

- Definire una variante dell'esercizio 1, in cui ogni visitatore, nel caso in cui non riesca ad accedere alla mostra, «attende».

- Come realizzare l'attesa? → con gli strumenti finora visti l'unica soluzione possibile è **l'attesa attiva...**