

Università degli Studi di Napoli Federico II

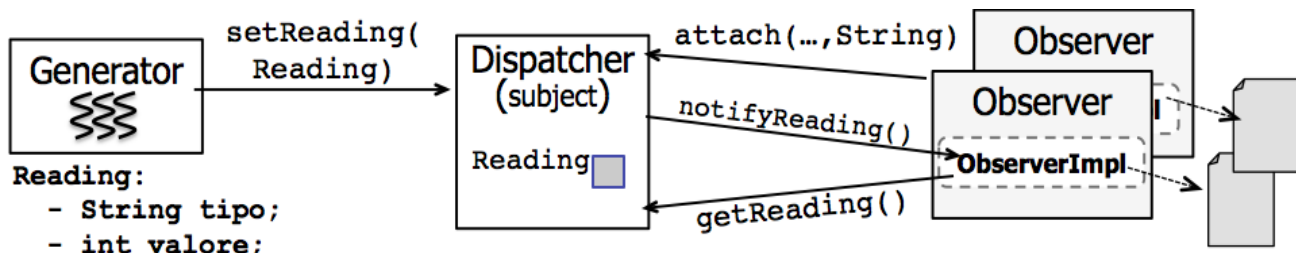
Esame di Programmazione II

Prova pratica del giorno 22/10/2019
Durata della prova: 120 minuti

Cognome..... Nome..... Matr.....

Lo studente legga attentamente il testo e produca il programma ed i casi di test necessari per dimostrarne il funzionamento. La mancata compilazione dell'elaborato, la compilazione con errori o l'esecuzione errata dell'algoritmo daranno luogo alla valutazione come **prova non superata**.

Ricordarsi di indicare Cognome, Nome e matricola su questo stesso foglio, che dovrà essere in ogni caso consegnato alla Commissione. Al termine della prova lo studente dovrà far verificare il funzionamento del programma ad un membro della Commissione.



Il candidato realizzi un sistema per la notifica distribuita di *letture di sensori* (istanze di **Reading**) basato sul pattern **Observer** e **JavaRMI**. Il sistema è composto dalle seguenti entità.

Generator. Avvia 3 thread. Ciascun thread esegue 3 invocazioni Java RMI di *setReading* sul Dispatcher, passando un'istanza di **Reading**; il thread attende 5 secondi tra due invocazioni *setReading* successive. Le istanze di *Reading* sono composte da i) campo tipo (String), scelto in maniera casuale tra “temperatura” o “pressione”, e ii) campo valore, generato in maniera casuale tra 0 e 50.

Observer si sottoscrive al Dispatcher (metodo Java RMI *attach*) per ricevere notifica delle letture associate ad un certo tipo. A tal fine, il metodo *attach* –oltre al *tipo* di lettura di interesse (ossia una stringa indicante “temperatura” o “pressione”)– prevede che sia passato al Dispatcher il riferimento remoto di un oggetto *callback* (istanza di *ObserverImpl*, come mostrato in figura) destinato a ricevere le notifiche provenienti dal Dispatcher, come specificato nel seguito.

Dispatcher (ruolo di subject). Lo “stato” del Dispatcher (subject) è rappresentato da un'istanza di *Reading*. Ad ogni *setReading*, il Dispatcher controlla il campo *tipo* dell'istanza di *Reading* ricevuta: nel caso in cui esistano *Observer* sottoscritti per letture di quel *tipo*, essi dovranno essere notificati dal Dispatcher. A tal fine, il Dispatcher invocherà il metodo di *callback* *notifyReading* fornito lato *Observer*. La *setReading* offerta da Dispatcher dura un tempo scelto a caso tra 1 e 5 secondi; il Dispatcher gestisce in mutua esclusione invocazioni concorrenti di *setReading*.

All'invocazione di *notifyReading*, *ObserverImpl* i) recupera l'istanza corrente di *Reading* invocando il metodo *getReading* su Dispatcher, e ii) salva su file il valore associato alla lettura ottenuta.

Il sistema sarà testato da prompt con un numero arbitrario di **Observer**; *tipo* e nome del file sono specificati da prompt, per es.:

```
java nomePackage.Observer temperatura lettureTemperatura.txt
```

Il candidato utilizzi **Java RMI**, produca le interfacce di Dispatcher ed Observer, ed implementi i meccanismi necessari al trasferimento delle istanze di *Reading*.