

**Fondamenti di Informatica – a.a. 2017-2018**  
**Prova pratica di programmazione del 18 Giugno 2018**

## MISURE

La coppia *tempo/valore* rappresenta una misura di una grandezza fisica, dove *tempo* è l'istante in cui avviene la misura e *valore* la grandezza misurata. Sia *tempo* una variabile intera e *valore* una variabile reale (double). La seguente classe MIS rappresenta una misura

```
public class MIS implements Comparable // -- classe Misura
{ //parte privata
    ...
    // parte pubblica
    public MIS(int t, double v);
    public int tempo() {...}
    public double valore() {...}
    public String toString() {...}
}
```

con il seguente significato dei metodi pubblici:

- MIS costruttore
  - tempo() restituisce il tempo in cui la misura è stata effettuata
  - valore() restituisce il valore della misura
  - toString() rappresentazione testuale nel formato (tempo valore); es: (1 27.5)
- L'ordine naturale della classe sia determinato dal tempo.

Un segnale è un contenitore di misure ed è codificato dalla seguente classe:

```
public class SIG implements Dizionario // -- classe Segnale
{ // parte privata
    ...
    // parte pubblica
    public SIG(String filename) {...}
    public MIS[] toSortedArray() {...}
}
```

dove

- il costruttore SIG(String filename) inizializza un segnale con le misure lette dal file il cui nome è passato come parametro; nel file ciascun campione è memorizzato nel formato (t v). Il file *temperatura.txt*, fornito dal docente, è un esempio di file contenente misure nel formato definito;
- il metodo toSortedArray restituisce un array di misure di dimensioni pari al numero di misure contenute nel segnale, ordinate secondo il loro ordine naturale.

L'interfaccia Dizionario è la seguente:

```
public interface Dizionario // -- interfaccia Dizionario
{ void makeEmpty();
  boolean isEmpty();
  int size();
  void insert(int t, double v);
  double find(int t) throws java.util.NoSuchElementException;
  void remove(int t);
}
```

I metodi dell'interfaccia svolgono le seguenti funzioni:

- makeEmpty rende vuoto il contenitore.
- isEmpty restituisce true se il contenitore è vuoto, false altrimenti;
- size restituisce il numero di elementi nel contenitore;
- insert inserisce la misura t/v nel contenitore. Se è già presente una misura di tempo t, questa viene sovrascritta;
- find restituisce il valore associato al tempo t. Se nel contenitore non c'è una misura di tempo t, lancia l'eccezione java.util.NoSuchElementException;

- remove rimuove dal contenitore la misura di tempo *t*, se questa è presente, altrimenti termina silenziosamente.

Il candidato completi il codice delle classi MIS e SIG. In sede di correzione il codice sarà provato con i seguenti comandi:

```
$ rm *.class
$ javac *.java
$ java ProvaSIG temperatura.txt
```

dove ProvaSIG è una classe fornita dal docente.

Nella realizzazione delle classi MIS e SIG:

- non è lecito aggiungere elementi pubblici;
- non si possono usare classi della libreria standard, a eccezione di quelle del pacchetto `java.io` e delle classi `java.util.Scanner` e `java.util.NoSuchElementException`.

Se il compito è svolto correttamente, l'output deve essere il seguente:

```
*** size = 5 ***

*** insert ***
(1 25.1)
(2 25.2)
(3 25.3)
(4 25.4)
(5 25.5)

*** find e remove ***
(1 25.1)
(2 25.2)
(3 25.3)
(4 25.4)
(5 25.5)

*** size = 0 ***
```

Al termine della prova il candidato dovrà lasciare nella cartella di lavoro i file modificati (MIS.java e SIG.java) e quelli forniti dal docente, che NON devono essere modificati (Dizionario.java, ProvaSIG.java e temperatura.txt).

I file prodotti dal candidato dovranno contenere un commento con nome e cognome del candidato, matricola, data, numero della postazione.

Sono valutati insufficienti i compiti che presentano errori di compilazione.

### FONDAMENTI DI INFORMATICA I - a.a. 2017-2018 - (prof. Sergio Canazza)

Nome e Cognome \_\_\_\_\_

Matricola \_\_\_\_\_ Postazione \_\_\_\_\_

☐ Consegno l'elaborato.    ☐ Non consegno l'elaborato e mi ritiro dall'esame.

Firma \_\_\_\_\_