

Esame di Programmazione II, 5 febbraio 2025

(si consegnino `AbstractTime.java`, `ItalianTime.java`, `AmericanTime.java` e `Interval.java`)

*Si crei un progetto Eclipse e il package `it.univr.time`. Si copino al suo interno le classi del compito. Non si modifichino le dichiarazioni dei metodi e delle classi. Si possono definire altri campi, metodi o costruttori non richiesti dal compito, ma devono essere **private**. Si possono definire altre classi, che in tal caso vanno consegnate. La soluzione che verrà consegnata dovrà compilare, altrimenti non verrà corretta.*

L'interfaccia `Time` rappresenta un istante di tempo (ore, minuti, secondi). Ne esistono due implementazioni, `ItalianTime` e `AmericanTime`, che si differenziano per il modo in cui vengono rappresentate dal metodo `toString`. Il codice comune fra queste due implementazioni è in `AbstractTime`. Per costruire una `ItalianTime` o una `AmericanTime` si indicano i secondi passati dall'inizio del giorno (cioè dalle 00:00:00 di quel giorno). I metodi di `Time` permettono di ottenere l'istante successivo con il metodo `next` e un intervallo di istanti successivi con il metodo `interval`. Per capire meglio, si consideri il seguente `Main.java`, già scritto e da non modificare:

```
public class Main {
    public static void main(String[] args) {
        Time t1 = new ItalianTime(12 * 3600 + 22 * 60 + 59); // le 12:22:59
        Time t2 = new AmericanTime(12 * 3600 + 22 * 60 + 59); // le 12:22:59
        Time t3 = new ItalianTime(23 * 3600 + 59 * 60 + 59); // le 23:59:59
        Time t4 = new AmericanTime(23 * 3600 + 59 * 60 + 59); // le 23:59:59
        System.out.println("t1 = " + t1); System.out.println("t2 = " + t2);
        System.out.println("t3 = " + t3); System.out.println("t4 = " + t4);
        System.out.println("t3.next() = " + t3.next());
        System.out.println("t4.next() = " + t4.next());

        System.out.println("10 secondi da t3 in poi:");
        for (Time t: t3.interval(10))
            System.out.println(t);

        System.out.println("10 secondi da t4 in poi:");
        for (Time t: t4.interval(10))
            System.out.println(t);
    }
}
```

che dovrebbe stampare:

```
t1 = 12:22:59
t2 = 12:22:59pm
t3 = 23:59:59
t4 = 11:59:59pm
t3.next() = 00:00:00
t4.next() = 12:00:00am
10 secondi da t3 in poi:
23:59:59
00:00:00
00:00:01
00:00:02
00:00:03
```

```
00:00:04
00:00:05
00:00:06
00:00:07
00:00:08
10 secondi da t4 in poi:
11:59:59pm
12:00:00am
12:00:01am
12:00:02am
12:00:03am
12:00:04am
12:00:05am
12:00:06am
12:00:07am
12:00:08am
```

Si noti la differenza di stampa fra le ore italiane e quelle americane, che usano am per la mattina e pm per il pomeriggio:

italiana	americana
00	12am
01	01am
02	02am
11	11am
12	12pm
13	01pm
14	02pm
15	03pm
23	11pm

Esercizio 1 (7 punti). Si completi la classe `AbstractTime.java`, che contiene il codice comune alle due rappresentazioni di un istante. Si faccia riferimento ai commenti dei metodi da completare.

Esercizio 2 (11 punti). Si completino le sottoclassi `ItalianTime.java` e `AmericanTime.java`, che si differenziano solo per i metodi `next` e `toString`. **Suggerimento:** nella conversione da ora italiana (00–23) a ora americana (am/pm) vi conviene gestire a parte i casi 00 e 12; inoltre può esservi utile il formato di stampa `%02d` di `String.format`, che rappresenta un `int` su due cifre, riempiendo con 0 la cifra a sinistra se l'intero avesse solo una cifra.

Esercizio 3 (13 punti). Si completi la classe `Interval.java` che rappresenta un intervallo di istanti consecutivi. Si noti che si tratta di un `Iterable<Time>`, quindi il suo metodo `iterator()` deve restituire un iteratore che ad ogni chiamata di `next` restituisce il prossimo `Time` dell'intervallo. Si veda l'esempio di utilizzo nel `main`, con i due cicli `foreach`.