

Esame di Programmazione II, 20 febbraio 2025

(si consegna `AbstractNote.java`, `AbstractSong.java`, `BasicSong.java`, `EnglishNote.java`, `IllegalNoteException.java`, `ItalianNote.java` e `Scale.java`)

*Si crei un progetto Eclipse e il package `it.univr.notes`. 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 `Note`, già completa, rappresenta una nota musicale tramite il suo semitono: note basse hanno semitono basso e note acute hanno semitono alto. Le note italiane e inglesi si differenziano per come il loro `toString` le rappresenta, secondo la seguente tabella:

ItalianNote	semitono	EnglishNote
1.do	0	1.C
1.do#	1	1.C#
1.re	2	1.D
1.re#	3	1.D#
1.mi	4	1.E
1.fa	5	1.F
1.fa#	6	1.F#
1.sol	7	1.G
1.sol#	8	1.G#
1.la	9	1.A
1.la#	10	1.A#
1.si	11	1.B
2.do	12	2.C
2.do#	13	2.C#
2.re	14	2.D
2.re#	15	2.D#
2.mi	16	2.E
2.fa	17	2.F
2.fa#	18	2.F#
2.sol	19	2.G
2.sol#	20	2.G#
2.la	21	2.A
2.la#	22	2.A#
2.si	23	2.B
3.do	24	3.C
3.do#	25	3.C#
eccetera	eccetera	eccetera

Le note hanno un metodo `shift` che permette di ottenere una nota a partire da un'altra nota, aggiungendo o togliendo semitoni. Per esempio, `(1.la).shift(3)` restituisce `2.do`, mentre `(1.G).shift(8)` restituisce `2.D#` e `(2.do).shift(-1)` restituisce `1.si`.

Esercizio 1 (2 punti). Si definisca l'eccezione controllata `IllegalNoteException` che viene usata per indicare che si sta cercando di creare una nota fuori dai limiti `0...Note.MAX_SEMITONE`.

Esercizio 2 (3 punti). Si completi la classe `AbstractNote.java`, che contiene il codice comune alle implementazioni concrete delle note.

Esercizio 3 (5 punti). Si completino le sottoclassi `ItalianNote.java` e `EnglishNote.java`, che si differenziano per `toString` e `shift`. Quest'ultimo metodo se applicato a una `ItalianNote` restituisce una `ItalianNote` e se applicato a una `EnglishNote` restituisce una `EnglishNote`.

Esercizio 4 (5 punti). L'interfaccia `Song` (già completa, da non modificare) rappresenta una canzone, cioè una sequenza di note. Iterando su un `Song` si devono ottenere le sue note. Si completi la classe `AbstractSong.java` con il codice comune alle implementazioni concrete delle canzoni, cioè il `toString`, che concatena il `toString` delle note della canzone con uno spazio come separatore.

Esercizio 5 (7 punti). Si completi la classe `BasicSong` che implementa una canzone le cui note sono fornite esplicitamente al costruttore e salvate dentro una lista di note.

Esercizio 6 (9 punti). Si completi la classe `Scale` che implementa una canzone le cui note sono ottenute come una scala che parte da una nota `start` fornita al costruttore e sale di semitono in semitono, per un totale di 12 note. Si chiede di risolvere questo punto senza aggiungere campi alla classe `Scale`, oltre a quelli che si trovano lì già definiti.

Per capire meglio, si consideri il seguente `Main.java`, già scritto e da non modificare:

```
public class Main {

    public static void main(String[] args) throws IllegalNoteException {
        // una canzone fatta da 4 note
        Song s1 = new BasicSong
            (new ItalianNote(5), new ItalianNote(25), new EnglishNote(13), new EnglishNote(38));
        // una scala dalla nota inglese di semitono 20 in su, fino alla nota inglese di semitono 31
        Song s2 = new Scale(new EnglishNote(20));
        System.out.println("s1 = " + s1); // stampa s1
        System.out.println("s2 = " + s2); // stampa s2
        System.out.println("s1.shift(-3) = " + s1.shift(-3)); // stampa s1 meno tre semitoni
        System.out.println("s2.shift(4) = " + s2.shift(4)); // stampa s2 piu' quattro semitoni
        System.out.println("s1 = " + s1); // ristampa s1 (non sara' cambiato)
        System.out.println("s2 = " + s2); // ristampa s2 (non sara' cambiato)
        s2.shift(29); // eccezione: spostando l'ultima nota di s2 si esce dai limiti 0...MAX_SEMITONE
    }
}
```

La sua esecuzione dovrebbe stampare:

```
1 = 1.fa 3.do# 2.C# 4.D
s2 = 2.G# 2.A 2.A# 2.B 3.C 3.C# 3.D 3.D# 3.E 3.F 3.F# 3.G
s1.shift(-3) = 1.re 2.la# 1.A# 3.B
s2.shift(4) = 3.C 3.C# 3.D 3.D# 3.E 3.F 3.F# 3.G 3.G# 3.A 3.A# 3.B
s1 = 1.fa 3.do# 2.C# 4.D
s2 = 2.G# 2.A 2.A# 2.B 3.C 3.C# 3.D 3.D# 3.E 3.F 3.F# 3.G
Exception in thread "main" it.univr.notes.IllegalNoteException
```