

# Esame di Programmazione 2

2 febbraio 2022, turno delle 11:30 (tempo disponibile: 2 ore)

Gli identificatori di un linguaggio di programmazione sono spesso scritti in *snake-style*, in cui le parole sono separate da underscore, come ad esempio `snakes_Are_slow_food` (il maiuscolo o minuscolo è a libera scelta del programmatore). Si immagini un altro modo di scrivere gli identificatori, in cui le parole vengono concatenate, le vocali si scrivono maiuscole e le lettere minuscole. Per esempio `snAkEsArEs10wf00d`. Questo modo alternativo di scrivere gli identificatori lo chiameremo *vowel-style*. Entrambi sono esempio di identificatori *multiword*, cioè composti a partire da più parole.

L'interfaccia che definisce un *identificatore* ha un unico metodo che permette di tradurre un identificare in stringa:

```
public interface Identifier {  
    String toString();  
}
```

## Esercizio 1 (13 punti) (si consegna MultiWordIdentifier.java)

Si completi la classe astratta `MultiWordIdentifier` che rappresenta un identificatore multiword. **È possibile aggiungere campi o metodi ma solamente se privati**. Si noti la presenza di due metodi `toString`: il primo richiamerà il secondo su ciascuna parola componente dell'identificatore e concatenerà il risultato in un'unica stringa.

```
@Override public final String toString() {  
    // TODO: si richiami il metodo ausiliario toString(pos, word) sulle componenti  
    // e si concateni il risultato in un'unica stringa  
    return "";  
}
```

```
// restituisce la stringa con cui si stampa la componente pos-esima dell'identificatore  
protected abstract String toString(int pos, String word);
```

## Esercizio 2 (9 punti) (si consegna VowelStyleIdentifier.java)

Si completi la classe concreta `VowelStyleIdentifier`, sottoclasse di `MultiWordIdentifier`. Si noti che la classe ha tre costruttori: il primo costruisce l'identificatore a partire dalle stringhe componenti, il secondo a partire da un qualsiasi iterabile e il terzo concatenando le componenti di altri identificatori:

```
public VowelStyleIdentifier(String... words) {  
    // TODO  
}  
  
public VowelStyleIdentifier(Iterable<String> words) {  
    // TODO  
}  
  
public VowelStyleIdentifier(MultiWordIdentifier... ids) {  
    // TODO  
}
```

### Esercizio 3 (9 punti)

(si consegna `SnakeStyleIdentifier.java`)

Si completi la classe concreta `SnakeStyleIdentifier`, sottoclasse di `MultiWordIdentifier`, analogamente a quanto fatto per `VowelStyleIdentifier`.

\* \* \*

Per esempio, un'esecuzione del file `Main.java` fornito è:

```
Inserisci una parola alla volta e termina con END
ciao
Amico
comE
va
END
id1 = cIAOAmIcOcOmEvA
id2 = ciao_Amico_comE_va
id3 = cIAOAmIcOcOmEvAcIAOAmIcOcOmEvA
id3 snake style = ciao_Amico_comE_va_ciao_Amico_comE_va
```