



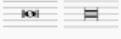



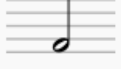


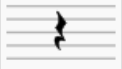


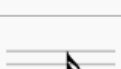


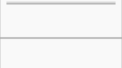
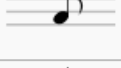
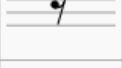
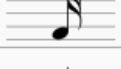
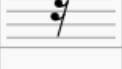

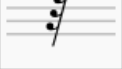


**CODIFICA
DELLA DURATA
DI NOTE
E PAUSE**

LA DURATA DELLE NOTE

- La durata delle note può essere codificata e rappresentata in diversi modi:
 - Simboli grafici in partitura
 - Etichette testuali
 - Corrispondenza con valori numerici frazionari, con riferimento alla durata teorica delle note (e delle pause)
 - Corrispondenza con valori numerici razionali, con riferimento alla durata assoluta delle note (e delle pause)
- L'adozione di stringhe e caratteri presenta vantaggi e svantaggi simili a quanto visto per le altezze

Nota	Pausa	Nome italiano	Nome britannico	Valore
		Massima (<i>in disuso</i>)	Maxima	$\frac{32}{4}$
		Lunga (<i>in disuso</i>)	Longa	$\frac{16}{4}$
		Breve (<i>rara</i>)	Breve	$\frac{8}{4}$
		Semibreve	Semibreve	$\frac{4}{4}$
		Minima	Minim	$\frac{2}{4}$
		Semiminima	Crotchet	$\frac{1}{4}$
		Croma	Quaver	$\frac{1}{8}$
		Semicroma	Semiquaver	$\frac{1}{16}$
		Biscroma	Demisemiquaver	$\frac{1}{32}$
		Semibiscroma	Hemidemisemiquaver	$\frac{1}{64}$
		Fusa (<i>in disuso</i>)	Quasihemidemisemiquaver	$\frac{1}{128}$
		Semifusa (<i>in disuso</i>)	Demisemihemidemisemiquaver	$\frac{1}{256}$

RAPPRESENTAZIONI NUMERICHE DELLA DURATA

- Nella teoria musicale moderna, i valori delle note (figurazioni ritmiche) si indicano tramite frazioni, il cui numeratore è 1 e il denominatore è una potenza di 2
 - Si tratta di un **sistema relativo** per assegnare la durata alle note: una nota da un quarto dura il doppio di una nota da un ottavo, ecc.
 - Un modo testuale per assegnare il nome alle durate delle note tiene conto di questo approccio: «nota da un quarto», «ottavi e sedicesimi», ecc.
- Esistono poi rappresentazioni in **termini assoluti**, ad esempio il numero di secondi di durata (o multipli o sottomultipli dell'unità di tempo)

LEGAME TRA TEMPO RELATIVO E ASSOLUTO: BPM

- Il passaggio da una codifica relativa a quella assoluta avviene grazie al valore di metronomo, anche detto **BPM** (*beats per minute*)
- Fissato il valore ritmico della pulsazione - il *beat* - il valore di BPM stabilisce quante pulsazioni hanno luogo in un minuto

$$\text{BPM} : 60[\text{s}] = 1 : d_{\text{pulsazione}}[\text{s}]$$

$$d_{\text{pulsazione}} = 60 / \text{BPM}$$

- BPM = 60 pulsazioni al min. $\rightarrow d_{\text{pulsazione}} = 1\text{s}$
(se la pulsazione dura un quarto, $d_{\text{quarto}} = 1\text{s}$, $d_{\text{ottavo}} = 0.5\text{s}$, ecc.)
- Attenzione: il beat non corrisponde sempre al quarto, anche se per molte suddivisioni è così; ad es.: in 6/8 spesso il beat è di 3/8
 - BPM = 100 $\rightarrow d_{\text{pulsazione}} = 0.6\text{s}$
(se la pulsazione dura 3/8, $d_{\text{ottavo}} = 0.2\text{s}$, ecc.)

PAUSE

- D'ora in poi sarà possibile arricchire l'insieme di informazioni che descrivono una nota con la codifica del valore, espresso ad esempio tramite frazione
- Ha senso considerare la pausa come una nota “degenera”, ossia una nota cui manca informazione sull'altezza
- Dal punto di vista pratico, esistono molti modi alternativi per farlo: esprimere la durata delle pause con valori negativi, riservare al pitch della pausa un valore speciale di altezza (ad es. null, '?', '-', stringa vuota, valore numerico negativo, ecc.)
 - Attenzione: si sta progettando la rappresentazione interna delle note in un proprio progetto. Nello studio di formati e linguaggi già esistenti, la possibilità di scegliere le convenzioni non c'è (vedi ad esempio MIDI o Csound)

RAPPRESENTAZIONE NUMERICA RELATIVA: METODO 1

- Si pone in corrispondenza un singolo intero con il denominatore della nota, ipotizzando il numeratore sempre uguale a 1, e usando i numeri negativi per le pause
- Si tratta del reciproco della durata frazionaria → **Reciprocal Duration Code** (vedi slide successiva)
 - Esempio: intero = $1/1 \rightarrow 1$; metà = $1/2 \rightarrow 2$; quarto = $1/4 \rightarrow 4$; ecc.
- Vantaggi:
 - compattezza e semplicità
- Svantaggi:
 - si creano incongruenze nella gestione aritmetica (ad esempio, se si moltiplica un valore per 2 si passa alla figura più breve)
 - sono consentiti valori inesistenti (ad es. 5 o 13)
 - non sono consentiti alcuni valori esistenti (ad es. le note con punti o legature di valore)

RAPPRESENTAZIONE NUMERICA RELATIVA: METODO 2

- Si codifica il risultato della frazione.
 - Esempio: intero \rightarrow 1; metà = $\frac{1}{2} \rightarrow$ 0.5; quarto = $\frac{1}{4} \rightarrow$ 0.25; ecc.
- Vantaggi:
 - compattezza;
 - possibilità di rappresentare tutti i valori ottenibili con punti e legature di valore (ma non tutti i gruppi irregolari)
- Svantaggi:
 - si complica la ricostruzione algoritmica della scrittura originaria. Ad esempio, si pensi alla corrispondenza $0.875 \rightarrow$ metà seguita da due punti di valore
 - si consente di esprimere valori non esistenti (ad es. 0.173)

RAPPRESENTAZIONE NUMERICA RELATIVA: METODO 3

- Si mantiene l'intera frazione, più eventuali informazioni aggiuntive (numero di punti di valore, ecc.) tramite un'opportuna struttura dati
- Vantaggi:
 - completezza
 - aderenza alla notazione ritmica originaria
 - possibilità di utilizzare l'aritmetica sulle frazioni (somme, moltiplicazioni, divisioni, semplificazioni, ecc.)
- Svantaggi:
 - necessità di creare una classe apposita, con relativi metodi e attributi. In Java non esiste un tipo o una classe predefinita per la gestione delle frazioni (se ne proporrà una più avanti)

POTENZIALI PROBLEMI

- Gruppi irregolari



- Punti di valore: codificare il valore ritmico al netto o al lordo?
 - Ad esempio, per denotare una metà con il punto si usa la frazione $\frac{3}{4}$ oppure la frazione $\frac{1}{2}$ + presenza del punto?
- Legature di valore: codificare il valore ritmico aggregato o segnalare nella struttura dati la presenza di una legatura di valore?

ESERCIZIO

(RDC1.java, RDC2.java)

- Il software legge in ingresso una sequenza di almeno 2 elementi di valori interi, ove:
 - il primo valore rappresenta i BPM, con riferimento alla pulsazione da un quarto
 - i successivi valori – dotati di segno – sono in codifica Reciprocal Duration Code
- Il programma mostra in output la sequenza di valori ritmici in forma testuale, seguiti dalla loro durata assoluta in secondi
- Input: 120 2 8 6 -4
- Output:
 - nota da una metà, 1s
 - nota da un ottavo, 0.25s
 - valore non valido
 - pausa da un quarto, 0.5s

ESERCIZIO

(AbsoluteDurations.java)

- Il software riceve in ingresso un valore di BPM compreso tra 40 e 208 (corrispondente al quarto) e una durata espressa in secondi, eventualmente con virgola, e scrive tale durata come somma di figure ritmiche
- Input:
120 2
- Output:
Il valore ritmico e' costituito da 1 nota/e da 1/1
- Input:
60 2.125
- Output:
Il valore ritmico e' costituito da 1 nota/e da 1/2, 1 nota/e da 1/32