

Metodi Avanzati di Programmazione, Corso A, 21 Giugno 2023

- 1) Fornire le specifiche algebriche minimali e complete (semantiche e di restrizione) per il tipo astratto **Tubo** di cui si forniscono le seguenti specifiche sintattiche:

Tipi: SequenzaIndicizzata, Elemento, Intero

Operatori:

crea() → SequenzaIndicizzata

aggiungi(SequenzaIndicizzata, Elemento) → SequenzaIndicizzata // aggiunge l'Elemento in testa alla SequenzaIndicizzata

leggi(SequenzaIndicizzata, Intero) → Elemento // legge Elemento che è memorizzato nella posizione specificata dall' Intero passato come secondo argomento nella SequenzaIndicizzata passata come primo argomento; solleva errore se la posizione non è valida

scambia(SequenzaIndicizzata, Intero, Intero) → SequenzaIndicizzata // scambia gli elementi della SequenzaIndicizzata (primo argomento) che occupano le posizioni definite dagli interi specificati come secondo e terzo argomento, genera errore se gli interi non sono posizioni valide

conta(SequenzaIndicizzata) → Intero // restituisce il numero di Elementi memorizzati in SequenzaIndicizzata

combina(SequenzaIndicizzata, SequenzaIndicizzata) → SequenzaIndicizzata // concatena gli Elementi in posizione dispari della prima SequenzaIndicizzata con gli elementi in posizione dispari della seconda SequenzaIndicizzata

N.B. Qualora si decida di svolgere l'esercizio in forma tabellare, scrivere almeno due equazioni in maniera esplicita. Dichiarare tutte le assunzioni fatte per svolgere l'esercizio.

(11 punti)

- 2) Descrivere in maniera esaustiva l'astrazione generica in ADA e in Java evidenziando le differenze. Scrivere il tipo astratto **SequenzaIndicizzata** in Ada (con riferimento ai metodi **crea**, **aggiungi**, **leggi** definiti nell'esercizio 1) assumendo che Elemento sia generico. Mostrare l'uso del tipo astratto generico scritto in ADA per usare una **SequenzaIndicizzata** popolata con gli interi: 1 e 10, ed una SequenzaIndicizzata popolata con i caratteri: 'a', 'b' e 'c'. Commentare le decisioni prese nello scrivere il codice e il comportamento di questo.

(11 punti)

- 3) Definire in maniera **esaustiva e con opportuni esempi** il meccanismo della gestione delle eccezioni in Java. Realizzare in Java la classe **Elemento** (che modella un nome di tipo stringa e un valore di tipo reale) e la classe **SequenzaIndicizzata** con riferimento alle specifiche di **crea()**, **aggiungi()**, **leggi()** e **scambia()** definite nell'esercizio 1. Realizzare una versione del metodo **scambia** usando le eccezioni controllate e una versione usando le eccezioni non controllate. Scrivere un main per testare il codice scritto in entrambi i casi e chiarire le differenze tra le due versioni del metodo **scambia** scritte.

In aggiunta alla domanda precedente:

Gli iscritti che avrebbero dovuto frequentare l'esame di MAP negli AA 2017-2018, 2018-2019, 2019-2020, 2020-2021, 2021-2022 o 2022-23 (perché allora iscritti al secondo anno) DEVONO rispondere alla domanda

Usando la programmazione funzionale scrivere il metodo della classe **SequenzaIndicizzata** che raggruppa le istanze di **Elemento** in essa archiviate per nome e stampa a video la media dei valori nei gruppi costruiti.

Gli iscritti che avrebbero dovuto frequentare l'esame di MAP negli AA precedenti al 2017-2018 DEVONO rispondere alla domanda sulla programmazione funzionale o in alternativa alla seguente domanda

Descrivere in maniera esaustiva l'uso della parola chiave **final** in JAVA fornendo opportuni esempi.

(11 punti)