

Esercitazione #13

Raccomandazioni: Ove non altrimenti indicato, rispondete alle domande prima ragionando su carta e poi provando a editare, compilare ed eseguire. Gli esercizi etichettati con l'asterisco (*) sono più difficili: affrontateli dopo aver risolto gli altri.

Esercizio 1

(difficoltà media percepita: 89.13% [23 voti])

Si prenda in considerazione il seguente codice che implementa il tipo di dato astratto CodaCircolare. Una coda circolare è una struttura dati per la memorizzazione di oggetti (in questo caso numeri interi) che segue la logica FIFO (first input first output). In una coda circolare gli oggetti vengono inseriti in coda (tail) ed estratti dalla testa (head).

```
class CodaCircolare {
    private int head;
    private int tail;
    private int[] coda;
    private static final int MAXELEMENTI = 10;

    public CodaCircolare() {
        coda = new int[MAXELEMENTI];
        head = 0;
        tail=tail;
    }

    public static boolean isEmpty(CodaCircolare c) {
        return ...;
    }

    public static boolean isFull(CodaCircolare c) {
        return ...;
    }

    public static void inserisci(CodaCircolare c, int elemento) {
        if (!isFull(c)) {
            c.coda [c.tail] = elemento;
            c.tail = (c.tail+1) % c.MAXELEMENTI;
            System.out.println("elemento inserito");
        } else {
            System.out.println("coda piena");
        }
    }

    public static int estrai(CodaCircolare c) {
        if (!isEmpty(c)) {
            int elemento = c.coda[c.head];
            c.head = (c.head+1) % c.MAXELEMENTI;
            return elemento;
        } else {
            System.out.println("coda vuota");
            return 0;
        }
    }
}
```

- Individuare le condizioni per determinare se la coda è piena (**isFull**) e se la coda è vuota (**isEmpty**), andando a scrivere il codice adeguato nel rispettivo metodo.

Esercizio 2

(difficoltà media percepita: 84.21% [19 voti])

- Scrivere un programma **UsaCodaCircolare** che permetta di inserire interi nella coda finché non appare il segnale di coda piena
- Qual'è la caratteristica di questa implementazione? Si può fare diversamente? (sugg. guardare il numero di elementi presenti nella coda quando è piena). Cercate solo le risposte alle domande, non implementate variazioni, questo esercizio vi verrà riproposto più avanti.

Esercizio 3

(difficoltà media percepita: 81.67% [15 voti])

- Scrivere un programma **UsaCodaCircolare2** che permetta, senza modificare il codice del TDA, di svolgere le seguenti operazioni:
 1. creare una nuova coda vuota **c**;
 2. inserire nella coda **c** interi letti da tastiera, finché la coda non è piena;
 3. creare una nuova coda vuota **d**;
 4. estrarre i primi tre elementi da **c** ed inserirli in **d**

Esercizio 4

(difficoltà media percepita: 85.71% [14 voti])

- Scrivere un metodo **stampaCodaCircolare(CodaCircolare c)** che permetta, senza modificare il codice del TDA, di stampare tutti gli elementi della coda in questo formato: **[1,3,...,4]** dove, ad esempio, **1** è il primo elemento inserito e **4** è l'ultimo. Modificare il programma **UsaCodaCircolare2** in modo da stampare alla fine i contenuti delle coda **c** e **d**, usando il metodo **stampaCodaCircolare**.

Esercizio 5

(difficoltà media percepita: 68.75% [12 voti])

Modificate il codice di **CodaCircolare** aggiungendo:

- un costruttore che permetta di definire il numero di elementi massimo della coda (il numero viene passato come parametro);

Esercizio 6

(difficoltà media percepita: 82.5% [10 voti])

Modificate il codice di **CodaCircolare** aggiungendo:

- un costruttore che permetta di definire il numero di elementi massimo della coda e permetta di inizializzare la coda inserendo i valori passati attraverso un vettore;

Esercizio 7

(difficoltà media percepita: 80.56% [9 voti])

Modificate **UsaCodaCircolare2** in modo da:

- 1) creare la coda **c** con il numero di elementi massimo letto da input;
- 2) inserire interi letti da input finché la coda **c** non è piena;
- 3) creare la coda **d** con un numero di elementi massimo pari al doppio di quelli della coda **c** e in modo che sia inizializzata con un vettore contenente la prima metà degli elementi della coda **c**;
- 4) stampare le due code;

Esercizio 8

(difficoltà media percepita: 83.33% [9 voti])

- Modificare il tipo di dato astratto **CodaCircolare** inserendo i metodi **getTail(CodaCircolare c)** e **getHead(CodaCircolare c)** che permettono, rispettivamente, di ritornare l'elemento in coda e l'elemento in testa, senza apportare alcuna modifica alla coda stessa.
- Modificare il tipo di dato astratto **CodaCircolare** inserendo i metodi **getIndexTail(CodaCircolare c)** e **getIndexHead(CodaCircolare c)** che permettono, rispettivamente, di ritornare l'indice dell'array dell'elemento in coda e dell'elemento in testa, senza apportare alcuna modifica alla coda stessa.
- Modificare il tipo di dato astratto **CodaCircolare** inserendo il metodo **getSize(CodaCircolare c)** che ritorna il numero di elementi presenti nella coda.

Esercizio 9

(difficoltà media percepita: 83.33% [9 voti])

Modificare **UsaCodaCircolare2** in modo da avere, sia per la coda **c** che per la coda **d** il seguente output:

```
index tail: valore tail: valore  
[valore1, valore2, ..., valoreN]
```

Esercizio 10

(difficoltà media percepita: 80.56% [9 voti])

Modificare il tipo di dato astratto **CodaCircolare** inserendo il metodo **getVettore(CodaCircolare c)** che ritorna un array contenente tutti e soli gli elementi della coda **c**, senza modificare gli elementi presenti nella coda.

Esercizio 11

(difficoltà media percepita: 86.11% [9 voti])

(*) Modificare il tipo di dato astratto **CodaCircolare** inserendo un metodo **somma(CodaCircolare c, CodaCircolare d)** che date due code circolari in input restituisce una nuova **CodaCircolare** che è l'unione delle due in ingresso (sfruttare il metodo precedente). Creare un programma che crea due code circolari, indicando il numero di elementi massimi ed il vettore degli elementi, richiami il metodo **somma** e stampi le due code create e quella restituita dal metodo.

Esercizio 12

(difficoltà media percepita: 0.0% [0 voti])

(*)Modificare l'implementazione della **CodaCircolare** per evitare l'anomalia accennata all'ultimo punto dell'esercizio 1.