

Progettazione di software sicuro on line

Esame del 17 maggio 2013 – (Prova di laboratorio)

1 Queue [pt. 2]

Scrivere una classe *Queue* che rappresenti un coda FIFO contenente stringhe. La classe rappresenta la coda usando un array di stringhe: le celle libere dell'array contengono la stringa vuota "".

Il costruttore della classe deve inizializzare l'array usando come dimensione il valore che viene passato come parametro al costruttore. All'inizio la coda è vuota.

Metodi add e pop La classe deve disporre di un metodo booleano *add(String str)* che permetta di aggiungere un elemento in fondo alla coda; il metodo è mostrato nel Codice 1. Se l'inserimento è possibile, il metodo aggiunge la stringa e ritorna *true*, altrimenti ritorna *false*.

La classe deve inoltre disporre di un metodo booleano *pop()* che permetta di rimuovere un elemento dalla cima della coda; il metodo è mostrato nel Codice 2. Se la coda non è vuota ritorna la stringa in cima, altrimenti ritorna *null*.

```
public boolean add(String str) {
    if(size < capacity) {
        arr[(top + size)%capacity] = str;
        size++;
        return true;
    }
    return false;
}
```

Codice 1: Metodo *add*

```
public String pop() {
    if(size > 0) {
        String str = arr[top];
        arr[top] = "";
        top = (top + 1)%capacity;
        size--;
        return str;
    }
    else {
        return null;
    }
}
```

Codice 2: Metodo *pop*

Metodo numberOfDistinctWords La classe deve disporre di un metodo *numberOfDistinctWords* che ritorni il numero di parole distinte presenti nella coda (quindi non

bisogna considerare le celle vuote (stringa "") dell'array). Tabella 1 mostra alcuni esempi.

array	# parole distinte
"pippo", "", "pippo", "sdsada"	2
"pippo", "", "pluto", ""	2
"", "", "", ""	0
"pippo", "dsad", "pluto", "wwds"	4

Table 1: Conteggio parole distinte

2 JUnit

In JUnit, scrivere:

- ✓ • un caso di test che mostri che, se si prova ad prelevare un elemento dalla coda vuota, viene ritornato *null*; [pt. 1]
- ✓ • un caso di test che mostri che, se si prova ad aggiungere una stringa alla coda piena, la stringa non viene aggiunta (basandosi sul valore ritornato dal metodo); [pt. 1]
- ✓ • un caso di test che mostri che il funzionamento della coda è effettivamente FIFO; [pt. 1]
 - un caso di test che testi il funzionamento del metodo *numberOfDistinctWords*. [pt. 1]

3 JML

Scrivere in JML la seguente preconditione al costruttore:

- { • la dimensione massima della coda deve essere compresa tra 2 e 100. [pt. 0.25]

Scrivere in JML la seguente postcondizione al costruttore:

- la coda non contiene elementi. [pt. 0.5]

Scrivere in JML la seguente preconditione al metodo *add(String str)*:

- ✓ • la lunghezza della stringa che si vuole aggiungere è compresa tra 2 e 10 caratteri. [pt. 0.25]

Scrivere in JML le seguenti postcondizioni al metodo *add(String str)*:

- ✓ • il metodo ha ritornato *true* se e solo se, prima dell'esecuzione del metodo, la dimensione della coda era minore della sua capacità massima; [pt. 1.5]
- la media delle lunghezze delle stringhe contenute nell'array (tutte le stringhe, anche quelle vuote) è minore od uguale a 10. [pt. 1.5]

Totale punti = 2 + 4 + 4 = 10