

Es. 10.1 – Cinema

Un `Cinema` è identificato da una stringa di lunghezza minore o uguale a 30 caratteri, e mantiene informazioni sulle prenotazioni dei posti della sala. La sala del cinema è composta da un numero di file minore o uguale a 26, indicate da lettere maiuscole dell'alfabeto (A è la prima fila, B è la seconda fila e così via). Ogni fila ha un numero di posti minore o uguale a 9, numerati con numeri interi a partire da 1.

Implementare il tipo di dato astratto `Cinema`, su cui possono essere effettuate le seguenti operazioni:

PRIMA PARTE

- **`Cinema c(n, p, s)`**: Costruttore che inizializza un `Cinema c` con nome indicato dalla stringa `s`, `n` file e `p` posti per fila. Inizialmente tutti i posti sono liberi.
- **`c.prenota(l, j)`**: Operazione che prenota il posto `j` della fila `l`, dove `j` è un intero ed `l` è un carattere. Se il posto è già occupato, la funzione lascia il cinema inalterato e restituisce `false`. Se l'operazione ha successo la funzione restituisce `true`.
- **`c.cancella(l, j)`**: Operazione che cancella la prenotazione del posto `j` della fila `l`. Se il posto è libero, la funzione lascia il cinema inalterato e restituisce `false`. Se l'operazione ha successo la funzione restituisce `true`.
- **`c.stampa()`**: Operazione che mostra a video il nome del cinema e, per ogni fila e ogni posto, mostra se il posto è libero o prenotato. L'output deve avere il seguente formato:

```
Nome del cinema: Moderno
 1234567
A    *
B
C*****
```

In questo esempio, il nome del cinema è "Moderno", la sala ha 3 file e 7 posti per fila. Nella prima fila tutti i posti sono liberi eccetto il quarto; nella seconda fila tutti i posti sono liberi; nella terza fila tutti i posti sono prenotati.

SECONDA PARTE

- **`c.quantiliberi()`**: Operazione che restituisce il numero di posti liberi nella sala.
- **`c.prenotaAdiacenti(k)`**: Operazione che prenota i primi `k` posti adiacenti liberi nella stessa fila, a partire dalla prima fila e dal primo posto. Se l'operazione fallisce (ovvero, non ci sono `k` posti liberi adiacenti), la funzione restituisce `false`. Se l'operazione ha successo la funzione restituisce `true`.

Definire opportunamente i file `compito.h` e `compito.cpp`, contenenti rispettivamente la dichiarazione e l'implementazione del tipo di dato astratto `Cinema`, definito dalle precedenti specifiche. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

Utilizzare il seguente file *main.cpp* di prova:

```
#include <iostream>
#include "compito.h"
using namespace std;

int main(){

    // PRIMA PARTE
    Cinema c(3,7,"Moderno");
    c.prenota('A',4);
    c.prenota('C',1);
    c.prenota('C',2);
    c.prenota('C',3);
    c.prenota('C',4);
    c.prenota('C',5);
    c.prenota('C',6);
    c.prenota('C',7);
    c.prenota('B',4);
    c.stampa();
    c.cancella('B',4);
    c.stampa();

    // SECONDA PARTE
    cout << "Il numero di posti liberi in c e': " << c.quantiliberi() << endl;
    c.prenotaAdiacenti(4);
    c.stampa();

    return 0;
}
```

L'output desiderato è il seguente:

```
Nome del cinema: Moderno
 1234567
A    *
B    *
C*****
Nome del cinema: Moderno
 1234567
A    *
B
C*****
Il numero di posti liberi in c e': 13
Nome del cinema: Moderno
 1234567
A    *
B****
C*****
```