AlgaT

Progetto di Algoritmi e Strutture dati
A.A. 2018-2019
Mengoli Chiara
Raimondi Bianca
Amato Alessio

Indice

AlgaT

Che cos'è

AlgaT(ALGA as Tutorial) è un'applicazione stand- alone basata su JavaFX. L'applicazione mostra un tutorial su un argomento a scelta del corso di Algoritmi e Strutture Dati. Questo progetto si presta a mostrare gli alberi bilanciati di ricerca tramite la struttura dati degli alberi RedBlack.

Interfaccia utente

L'applicazione all'apertura mostra una pagina iniziale contenente un'introduzione sull'argomento scelto e la possibilità di accedere, tramite appositi pulsanti, ai tutorial.
I tutorial simulano in modo interattivo il funzionamento della struttura dati degli alberi alberi bilanciati di ricerca. L'applicazione offre due tutorial: il primo tutorial mostra la ricerca, dato in input il valore di una chiave, di un nodo all'interno dell'albero, il secondo mostra le operazioni di inserimento e cancellazione di un nodo, dato in input il valore di una chiave. Il funzionamento di tale struttura dati viene mostrato graficamente e accompagnato da una spiegazione testuale, il tutto avviene tramite un'esecuzione step-by-step gestita dall'utente tramite i relativi bottoni.
Ogni tutorial inoltre permette di accedere ad una fase di domande di autoapprendimento sempre inerenti all'argomento scelto; una volta terminate tutte le domande con esito positivo, l'utente viene riportato alla pagina iniziale.

Progetto

Struttura

```
AlgaT_mod
    |_______ src
              |______algat_mod
                          |_____ AlgaT_mod.java
                          |_____ DomandeController.java
                          |_____ GraficaAlbero.java
                          |_____ PaginaInizialeController.java
                          |_____ TutorialController.java
                          |_____ domande
                          |              |_____ DomandeTutorial1
                          |              |_____ DomandeTutorial2
                          |
                          |_____ fxml
                          |              |_____ Domande.fxml
                          |              |_____ PaginaIniziale.fxml
                          |              |_____ Tutorial.fxml
                          |
                          |______ img
                          |              |_____ RedBlackTree.png
                          |
                          |______ tree
                                         |_____ Nodo.java
                                         |_____ RedBlackTree.java
```

Scelte implementative

Domande di autoapprendimento
Le domande per la fase di autoapprendimento sono memorizzate su file e vengono caricate tramite l'istanziazione di un oggetto FileReader() a cui viene passato come paramentro il percorso del file da caricare, e un oggetto BufferedReader() che ci permette tramite il metodo readLine() di leggere riga per riga il file di testo.
Questa scelta rende più facile l'inserimento e/o la modifica delle domande.
Il file di testo che viene caricato è strutturato in modo che la prima riga contenga la domanda e la riga successiva contenga la risposta relativa,e cosi via.

Interfaccia grafica

L'interfaccia grafica è stata implementata con l'utilizzo di JavaFX Scene Builder, generando un file con estensione .fxml. E' stata creata un'interfaccia con un design responsive, capace di adattarsi alle diverse dimensione degli schermi, a tal proposito, le dimensioni dell'applicazione, in fase di apertura, vengono calcolate in proporzione allo schermo del dispositivo su cui viene lanciata.

Ogni pagina fa riferimento ad un controller, una classe java, che gestisce l'inizializzazione della pagina stessa e gli eventi possibili sulle varie componenti della pagina (es. evento di click su un bottone).

Il caricamento delle pagine fxml avviene tramite l'istanziazione di un oggetto FXMLLoader a cui viene passato il percorso del file con estenzione .fxml, che poi verrà caricato tramite il metodo load().

Esecuzione step-by-step

Per l'esecuzione step-by-step utilizziamo i Thread, essi sono avviati con il comando nomeThread.start() all'interno della funzione insert o delete nella classe RedBlackTree.

All'interno dei Thread sono presenti un'istruzione Thread.sleep() ,che useremo come timer in modo da poter permettere al programma di disegnare la struttura dell'albero prima di bilanciarlo, e le funzioni insertFixUp() o deleteFixUp() per poter bilanciare l'albero.

Successivamente premendo il tasto Forward verrà disegnato l'albero dopo il bilanciamento.

Nel caso si voglia annullare l'ultimo inserimento o l'ultima cancellazione basta premere il tasto Back che eseguira l'opposto dell'ultima azione eseguita(tutte le azioni sono registrate in una matrice contenente numero e tipo di azione).

Durante l'esecuzione step by step non sarà possibile inserire, cancellare o annullare l'ultima azione.

RedBlackTree

Caratteristiche

La grafica degli alberi è stata creata utilizzando tre classi. Le prime due (Nodo e RedBlackTree) servono per implementare la struttura al di sotto della grafica, ovvero l'albero in sé. Nodo viene utilizzata per implementare le caratteristiche di un nodo dell'albero redblack (chiave, figlio sinistro e destro, padre e colore). La seconda classe viene utilizzata per le funzioni di un albero, settando la radice come Nodo con i rispettivi figli. Questa classe viene utilizzata per le funzioni ricerca inserimento e cancellazione, per le quali sono state create le rispettive funzioni di ribilanciamento (inserfixup e deletefixup) che implementano il cambio colore dei nodi e le rispettive rotazioni, in modo tale che l'albero una volta inserito/cancellato un nodo si ribilanci da sè.
L'ultima classe, GraficaAlbero, viene invece utilizzata per la visualizzazione grafica della struttura dell'albero. Qui sono state implementate varie funzioni per disegnare i nodi del redblacktree.
La funzione che disegna l'albero (assieme alle funzioni che vengono chiamate da essa) viene chiamata ogni volta che la struttura viene modificata da un inserimento o da una cancellazione, in modo tale da visualizzare l'albero e i suoi rispettivi colori.