

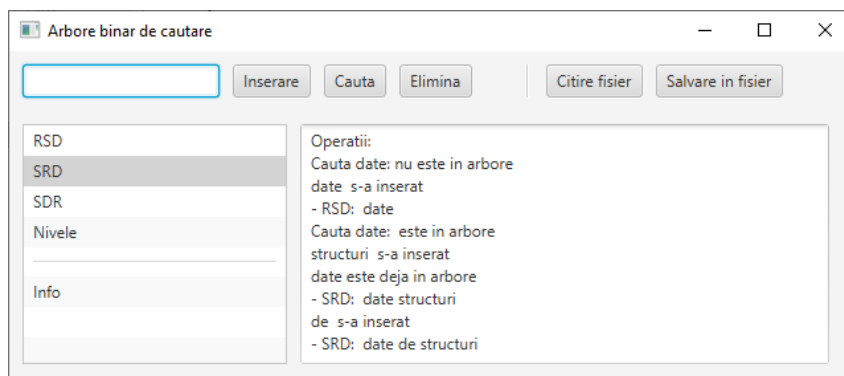
Laborator 9

Vocabular - Arbore binar de căutare

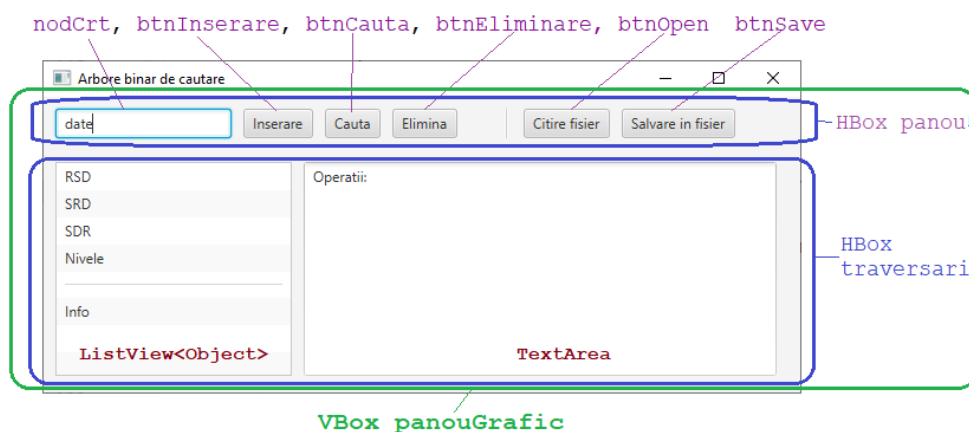
În acest laborator aveți de scris o aplicație grafică care să permită efectuarea unor prelucrări asupra unui *arbore binar de căutare* care memorează cuvinte (siruri de caractere).

Veti scrie clasa **PanouGrafic** care afișează un panou grafic ca cel din figura următoare și care va realiza următoarele acțiuni comandate prin acționarea unor butoane sau opțiuni din ListView:

- **Inserare** – se inserează în arbore sirul din câmpul de editare; se afișează în TextArea sirul din câmpul de editare urmat de unul din textele următoare: “s-a inserat” sau “este deja în arbore”.
- **Cauta** – se caută în arbore sirul din câmpul de editare; se afișează în TextArea “Cauta ” urmat de sirul din câmpul de editare, după care se afișează rezultatul căutării “: este în arbore” sau “: nu este în arbore”.
- **Elimina** – se elimină din arbore sirul preluat din câmpul de editare; se afișează în TextArea sirul din câmpul de editare urmat de rezultatul eliminării “s-a eliminat din arbore” sau “nu este în arbore”
- **Citire fisier** – se va citi un fișier care conține cuvinte pe care le veți insera în arbore; aveți la dispoziție [fișierul vocabular fara diacritice](#) care conține 7273 cuvinte; după ce s-a citit un fișier se va afișa în TextArea numărul de cuvinte citite
- **Salvare fisier** – scrie într-un fișier text cheile din arbore, câte una pe fiecare linie; se va afișa numărul de chei salvate
- **RSD, SRD, SDR, Nivele** – (opțiuni din ListView) se afișează în TextArea item-ul selectat
- **Info** – (opțiune din ListView) se afișează în TextArea “Program ... autor”



O sugestie pentru realizarea panoului grafic: scena are ca root un **VBox panouGrafic** care conține două **HBox**-uri, ca în figura următoare:



Puteti folosi codul prezentat la curs sau fragmentele de cod de mai jos pe care sa le completati dv.

```
public class PanouGrafic extends Application {

    private Stage mainStage;
    private String sirIntrodus="";

    private TextField nodCrt = new TextField();
    private Button btnInserare = new Button("Inserare");
    // . . .

    private HBox operatiiArbore() {
        HBox panou = new HBox(10, nodCrt, btnInserare, btnCauta, btnEliminare,
                               btnOpen, btnSave);

        btnInserare.setOnAction(a -> {
            sirIntrodus = nodCrt.getText().trim();
            zonaTextArea.appendText("\n" + sirIntrodus);
            if( sirIntrodus.length() != 0) {
                zonaTextArea.appendText(arb.add(sirIntrodus) ? " s-a inserat"
                                                : " este deja in arbore");
                nodCrt.setText("");
            }
        });

        btnCauta.setOnAction(a -> {
            // . . .
        });

        btnEliminare.setOnAction(a -> {
            // . . .
        });

        btnOpen.setOnAction(e -> {
            File fin = deschideFisier(true);
            if(fin == null)
                return;
            try {
                // citire fisier . . .
                zonaTextArea.setText("\nS-au citit "+ nr+ " cuvinte.");
            } catch (FileNotFoundException fileNotFoundException) {
                fileNotFoundException.printStackTrace();
            }
        });

        btnSave.setOnAction(e -> {
            // . . .
        });

        return panou;
    }

    private HBox traversari() {
        zonaTextArea = getTextArea("Operatii:");
        return new HBox(10, getListaSelectie(), zonaTextArea);
    }

    @Override
    public void start(Stage primaryStage) {
        mainStage = primaryStage;
        VBox panouGrafic = new VBox(20, operatiiArbore(), traversari());
        panouGrafic.setPadding(new Insets(10));

        Scene scena = new Scene(panouGrafic, 640, 250);
```

```

        primaryStage.setScene(scena);
        primaryStage.setTitle("Arbore binar de cautare");
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
} // sfarsit clasa PanouGrafic

```

Funcția de deschidere fișier utilizabilă și pentru citire și scriere.

```

private File deschideFisier(boolean citire) {
    FileChooser fileChooser = new FileChooser();
    fileChooser.setTitle("Deschide fișier cu cuvinte");
    fileChooser.getExtensionFilters().addAll(
        new ExtensionFilter("Text Files", "*.txt"),
        new ExtensionFilter("All Files", "*.*"));
    File selectedFile = citire ? fileChooser.showOpenDialog(mainStage)
        : fileChooser.showSaveDialog(mainStage);
    if (selectedFile != null) {
        zonaTextArea.appendText("\n" + selectedFile);
    }
    return selectedFile;
}

```

Clasa ArboreDeCautare o găsiți în curs.

Indicații:

1. Consultați ultimul [cursul în care se prezintă controalele grafice](#).
2. Pentru separatorul orizontal din **ListView** puteți proceda astfel:
 - Includeti **new Separator()** în lista observabilă de itemi; veți observa o eroare deoarece **Separator** nu este **<String>**; corectați eroarea utilizând un tip generic corespunzător
 - Inserați cod în *Listener* pentru ca atunci când este selectat **separatorul** acesta să nu producă nici o acțiune

Punctaj

- Controalele grafice din **HBox panou** – 5p (1p pt. funcționarea corectă a fiecărui buton)
- **TextArea** – 1p
- Aranjarea grafică conform figurii anterioare (1p)
- Funcționare **ListView** (afisare minim RSD și SRD în **TextArea**) - 2p

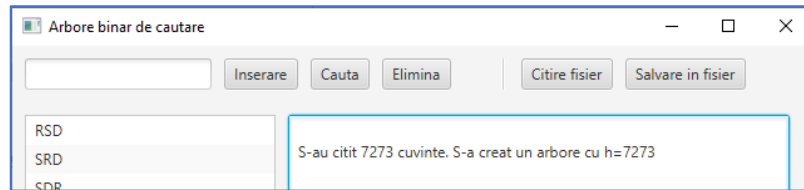
TEMA ACASA

1. Deoarece în vocabularul pus la dispoziție cuvintele sunt sortate alfabetic, inserarea lor în această ordine conduce la un arbore de cautare cu performanțe **$O(n)$** . Acest lucru poate fi observat dacă veți afișa înălțimea arborelui de cautare rezultat. Puteți folosi formula recursivă care dă înălțimea unui nod:

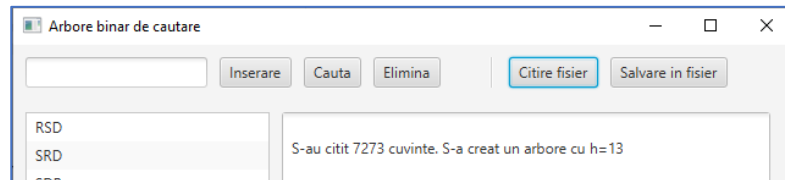
$$h(\text{nod}) = 1 + \max \{ h(\text{nod.stg}), h(\text{nod.dr}) \}$$

Un arbore vid are înălțimea 0.

Veti observa la citirea [fisierului vocabular fara diacritice](#) urmatoarele:



Pornind de la lista cuvintelor citite din fisier scrieti codul necesar ca sa se obtina un arbore de cautare in care performantele sa fie cat mai apropiate de $O(\log_2 n)$. Rezultatul la citirea [fisierului vocabular fara diacritice](#) trebuie sa fie urmatorul:



2. Dezvoltati operatia de cautare. Daca in `TextField` se introduc doua siruri separate prin semnul :, exemplu `as:bal`, programul va afisa toate sirurile din arbore care lexicografic sunt \geq cu „as” si strict mai mici decat „bal”.
3. Programati jocul **Fazan**. Veti insera un buton cu aceasta eticheta la actionarea caruia programul va alege in mod aleatoriu o litera pe care o va afisa in `TextArea`. Jucatorul va introduce in `TextField` un cuvnt care incepe cu acea litera, iar programul va continua cu un cuvnt care incepe cu ultimele 2 litere ale acelui cuvnt s.a.m.d. Bineinteles ca se va juca doar cu cuvintele memorate in nodurile arborelui de cautare.

Referinte

Oracle Tutorial JavaFx - <https://docs.oracle.com/javafx/2/>

Oracle - Getting Started with JavaFX – https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm

Oracle – Layouts - https://docs.oracle.com/javafx/2/layout/builtin_layouts.htm#CHDGHCDG

Tutorialspoint - Learn JavaFX Absolute Beginners -

<https://www.tutorialspoint.com/javafx/index.htm>

Controale grafice:

Oracle - https://docs.oracle.com/javafx/2/ui_controls/jfxpub-ui_controls.htm

Tutorialspoint – https://www.tutorialspoint.com/javafx/javafx_ui_controls.htm