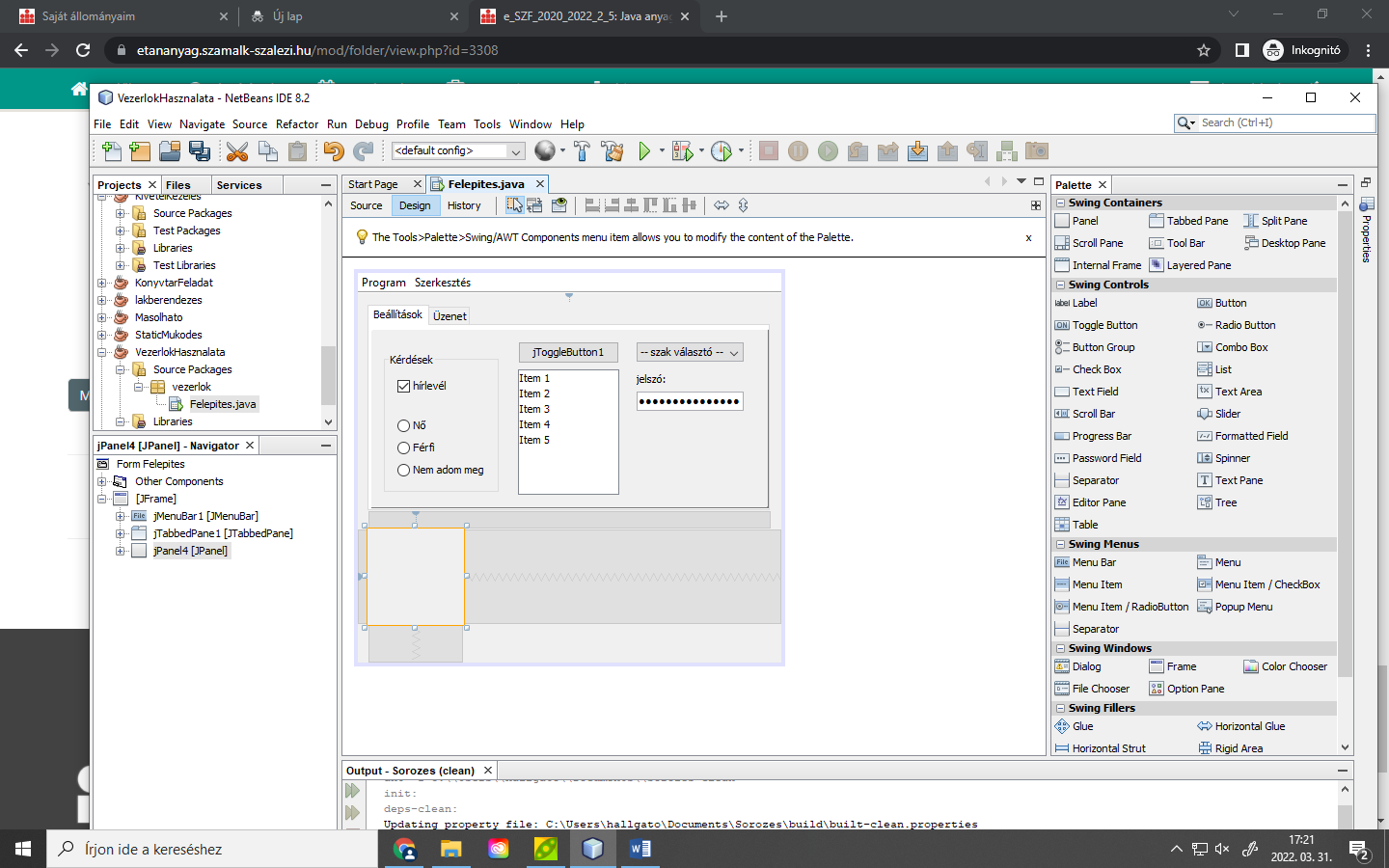
**VezerlokHasznalata**

jelszó mutatása

…

panel 🡪 border 🡪 Dinamikus tart.



**btnFelveszActionPerformed**:

String szak = txtujSzak.getText();

jComboBox1.addItem(szak);

**btnListbeActionPerformed**:

ListModel 🡪 nem írható elem, csak lekérdezésre tudjuk haszn.

DefaultListModel 🡪 interface 🡪 ehhez hozzá tudunk adni az addElement() metódus felhasználásával

ListModel 🡪ezt a DefaultListModel is megvalósítja!

if (jComboBox1.getSelectedIndex()>0{

String szak = jComboBox1.getItemAt(0);

ListModel<String> listModel = jList1.getModel();

DefaultListModel dlm = new DefaultListModel();

dlm.addElement(szak);

dlm.add(0,”elso”);

jList1.setModel(dlm);

} else {

JOptionPane.showMessageDialog(rootPane,”Szakot kell választani!”); //felugró ablak

}

**btnListbeActionPerformed:**

Alt + Enter 🡪 hosszabb rész kommentezése, becsukhatóvá tétele

int ind = jComboBox1.getSelectedIndex();

if(ind > 0){

String szak = jComboBox1.getItemAt(ind);

DefaultListModel dlm = new DefaultListModel();

if(cbHozzafuz.isSelected()){

ListModel<String> listModel = jList.getModel();

for(int i=0; i<listModel.getSize(); i++){

dlm.addElement(listModel.getElementAt(i));

}

if(rdb.Eleje.isSelected()){

dlm.add(0,szak);

}else{

dlm.addElement (0,szak);

}

}else {

dlm.addElement(szak);

}

jList.setModel(dlm);

} else {

JOptionPane.showMessageDialog(rootPane,”Szakot kell választani!”); //felugró ablak

}

}

Új buttongroup léterhozása: *buttonGroup3* 🡪 elejére/végére rádiogombok ehhez legyenek kötve

Fejlécet enged: toggleButton

ha benyomom, írja ki: Fejlécet enged, egyébként: Fejlécet NEM

**jToggleButton1ActionPerformed**

JToggleButton g = jToggleButton1;

g.setText(„Fejlécet NEM”);

if(g.isSelected()){

g.setText(„Fejlécet enged”);

}

**btnListbeActionPerformed** módosítása:

int ind = jComboBox1.getSelectedIndex();

if(ind > 0){

String szak = jComboBox1.getItemAt(ind);

DefaultListModel dlm = new DefaultListModel();

if(jToggleButton1.isSelected() || !jToggleButton1.isSelected() && ind>0){

ListModel<String> listModel = jList.getModel();

for(int i=0; i<listModel.getSize(); i++){

dlm.addElement(listModel.getElementAt(i));

}

if(rdb.Eleje.isSelected()){

dlm.add(0,szak);

}else{

dlm.addElement (0,szak);

}

}else {

dlm.addElement(szak);

}

jList.setModel(dlm);

} else {

JOptionPane.showMessageDialog(rootPane,”Szakot kell választani!”); //felugró ablak

}

}

Események

Events fül alatt találhatóak.

stateChanged 🡪 akkor hívódik meg, ha megv. az obj. állapota

pl. ComboBox 🡪 itemStateChanged

Az eseményekre fel kell iratkozni.

evt……..

Ha a Hozzáfűz ki van jelölve, csak akkor lehessen kiválasztani az elejére/végére opciókat, egyéb esetben ne legyenek aktívak.

Az obj. Events fülén a megfelelő eseménybe belekattintok 🡪 tudok kódot írni az eseményhez / feliratkoztam rá

**cbnHozzafuzStateChanged**

if (cbnHozzafuz.isSelected()){

rdbVege.setEnabled(true);

}else{

rdbVege.setEnabled(false);

rdbEleje.setSelected(true);

}

A Comboboxból kiválasztott elem kerüljön bele a jelszó mezőbe

**jComboBox1ItemStateChanged**

if(jComboBox1.getSelectedIndex()>0){ //a 0. elemnél ne…

pswJelszo.setText(jComboBox1.getSelectedItem()+””);

}

Ha az Új szak mezőbe beírunk valamit, és Enter-t nyomunk, akkor az kerüljön bele a Comboba

Ha történt változás, majd Entert nyomunk (Entert ütött-e), akkor fel kel venni a Comboba. Minden egyes karakterről meg kell vizsgálni, h. Enter-e

Esemény, amelyre fel kell iratkozni: keyReleased

**txtUjSZakKeyReleased**

jobb klikk + method

private void UjSzakotFelvesz(){

String szak = txtUjSzak.getText();

jComboBox1.addItem(szak);

}

if(/\*evt.getKeyCode()==10\*/ evt.getKeyCode()==KeyEvent.VK\_ENTER ){

UjSzakotFelvesz();

}

Ha felvettük az új szakot, akkor rögtön töröljük ki az új tartalmat, akár a Felvesz gombbal

private void UjSzakotFelvesz(){

String szak = txtUjSzak.getText();

jComboBox1.addItem(szak);

txtUjSzak.SetText(„”);

}

if(!txtUjszak.getText().isEmpty()){

btnFelvesz.setEnabled(true);

if (evt.getKeyCode()==KeyEvent.VK\_ENTER)

UjSzakotFelvesz();

}else{

btnFelvesz.setEnabled(false);

}

Ha nem választok ki nemet, akkor írja ki, h. nincs nem választva

Kilépés rákérdezéssel

Kilépés és a piros x u.azt a függvényt hívja meg

jFrame 🡪 windowClosing

**formWindowClosing**

kilepes();

Kilépés menüpont 🡪 mnuKilep

mnuKilepActionPerformed

kilepes();

private void kilepes(){

//JOptionPane.showMessageDialog(rootPane,”Kilépés”);

int gomb=JOptionPane.showConfirmDialog(rootPane,”Biztos kilép?”,”KILÉPÉS”, JOptionPane.OK\_CANCEL\_OPTION);

if(gomb==JOPtionPane.OK\_OPTION){

System.exit(0);

//Csak OK-ra lép ki

//akkor történjen meg, ha felugrik 1 ablak

}

}

b)

//message type hozzáadása

int gomb=JOptionPane.showConfirmDialog(rootPane,”Biztos kilép?”,”KILÉPÉS”, JOptionPane.OK\_CANCEL\_OPTION,JOptionPane.WARNING\_MESSAGE);

if(gomb==JOPtionPane.OK\_OPTION){

System.exit(0);

//Csak OK-ra lép ki

//akkor történjen meg, ha felugrik 1 ablak

}

}

c)

//ikon hozzáadása

Icon icon = new ImageIcon(„fajlnev.jpg”); //Stringet visszaadó metódus

int gomb=JOptionPane.showConfirmDialog(rootPane,”Biztos kilép?”,”KILÉPÉS”, JOptionPane.OK\_CANCEL\_OPTION,JOptionPane.WARNING\_MESSAGE);

if(gomb==JOPtionPane.OK\_OPTION){

System.exit(0);

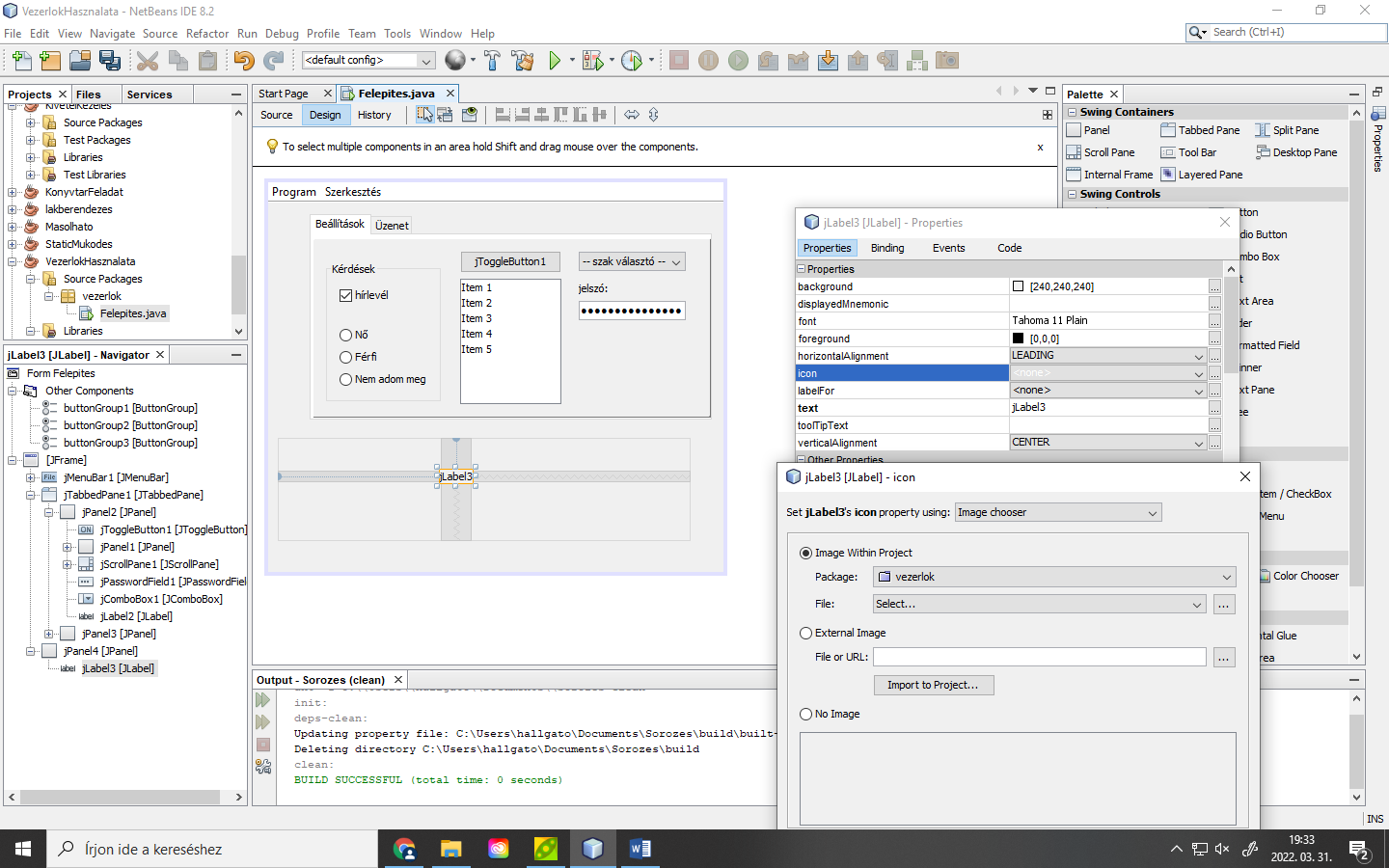
//Csak OK-ra lép ki

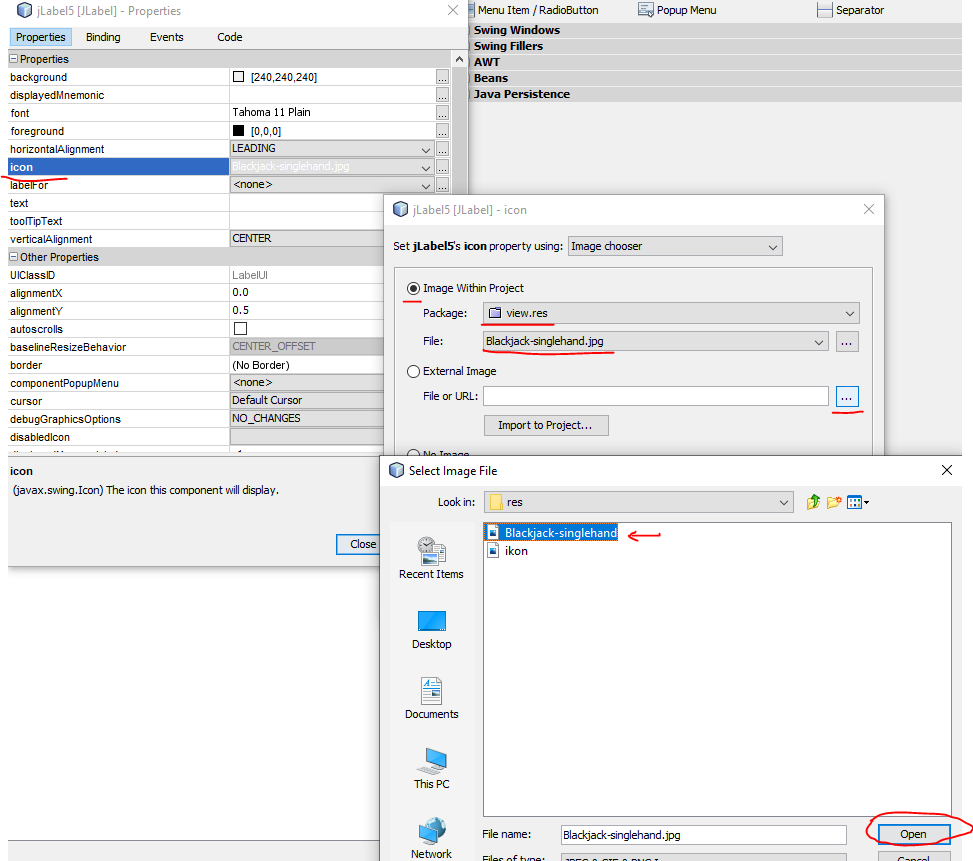
//akkor történjen meg, ha felugrik 1 ablak

}

}

Kép hozzáadása jLabel-hez: jLabel 🡪 Import 🡪 import to project





src-vel egy szintre kell pakolni a képfájlt

ha máshova, pls src-n belülre egy kepek mappába: src/kepek/fajlnev.jpg

Run 🡪 Clean and build 🡪 dist mappa:

Ebbe belekerül egy JAR fájl, de ebben nincs még ott a kép

Ha pl. a src/vezerlok alatt található a projekt --> src/vezerlok/Felepites.java, akkor a src/vezerlok alá kell egy 'kepek' mappa, abba helyezni a képfájlt, és akkor az alábbi módon elérhetővé válik

Icon icon = new ImageIcon(this.getClass().getResource(„kepek/fajlnev.jpg”)); //a jarba így belekerül