

## A Programom:

Egy Szakácskönyvre gondoltam amiben (jól gondoltad) recepteket tárolhatunk :). A program indításakor a főablakban kiválaszthatjuk majd a "Menü" gombra kattintva hogy milyen ételről lenne szó pl:(előétel, főétel,saláta,desszert) vagy amit a felhasználó megadott. Kiválasztáskor ha már tartalmaz receptet az adatbázis akkor betölti és ki írja egy listában a kiválasztott csoporton belül az ételek neveit abc sorrendben és mellé a felvételi dátumot. Ha nincs az adott csoporton belül étel akkor egy üres listát látunk majd és a mellette lévő gombok segítségével(Elkészítés,Hozzáadás,Módosítás,Törlés) kiválaszthatjuk mit szeretnénk csinálni.

Új recept felvételekor eltárolnánk 1 adaghoz az alapanyagokat,anyagmennyiséget, a beszerzési költségeit,elkészítését.Ha ezt el szeretnénk készíteni akkor az elkészítés gombra rákattintva, megkérdezi majd hány adagot szeretnénk belőle, majd az alapanyagok beszerzési árából és a receptben található mennyiségekből kiszámítja az adott adagra jutó mennyiség költségeit és a hozzávalók mennyiségét és ki írja az elkészítési módját.

## A Programom működése:

Az adatbázisban(MySQL) létrehozunk egy sémát "receptek" néven, ennek lesz 2 táblája:

- **recept** (ez tartalmaz 5 oszlopot):

- **ID** (receptek felvételi számát tartalmazza), Típusa: INT, (PK,NN,AI kipipálva)
- **Nev** (Recept neve), Típusa: varchar(45), (NN kipipálva)
- **Fajta** (recept fajtája), Típusa: varchar(45), (NN kipipálva)
- **Elkesztese** (recept elkészítése), Típusa: LongText(NN kipipálva)
- **Datum** (recept felvételi dátuma), Típusa: DATE, (NN kipipálva)

- **hozzavalok** (ez tartalmaz 8 oszlopot):

- **Sor** (hozzávalók sorszámát tartalmazza) Típusa: INT (PK,NN,AI kipipálva)
- **Darab** (hozzávalók darabszámát tartalmazza az adott Recepthez)
- **ReceptID**(A recept táblán látható receptek ID-jével megegyezik) Típusa: INT (NN kipipálva)
- **ReceptNev**(A recept táblán látható receptek nevével megegyezik) Típusa: varchar(45), (NN kipipálva)
- **Mennyiseg**(A hozzávaló mennyiségét tartalmazza) Típusa: DOUBLE, (NN kipipálva)
- **Mertekegyseg** Típusa: varchar(45), (NN kipipálva)
- **Alapanyag** Típusa: varchar(45), (NN kipipálva)
- **Ara** Típusa: INT, (Default/Expression : '0')

Az adatbázis létrehozása után az osztályok a NetBeansben: Foablak.java, MenuDialog.java, ModositoDialog.java, HozzavalDialog.java, ElkeszitesDialog.java, Adatbazis.java, AdatbazisKezelo.java, Recept.java, Hozzavalok.java

## SzakacsKonyvDAO osztály:

Ez egy interface osztály.

Metódusai:

```
public List<Recept> beOlvas(String fajta)
public void kiir(Recept recept)
public List<Hozzavalo> hozzavaloBeolvasasa(Recept recept)
public void hozzavaloKiir(Hozzavalo hozzavalo, Recept recept)
public void receptModosit(Recept recept)
public void hozzavaloModositasa(Hozzavalo hozzavalo)
public void receptTorlese(Recept recept)
public void hozzavaloTorlese(Hozzavalo hozzavalo)
public List<Recept> kereses(String nev, String fajta)
public List<String> etelFajtak()
public void bezar()
```

## SzakacsKonyvDAOJDBCImpl osztály:

Ez az osztály implementálja az "SzakacsKonyvDAO" osztály-t.

10 adattagja van:

```
private Connection conn;
private PreparedStatement receptBeolvas
private PreparedStatement receptKiir
private PreparedStatement recepTorlese
private PreparedStatement receptModosit
private PreparedStatement getID
private PreparedStatement hozzavalokBeolvas
private PreparedStatement hozzavalokKiir
private PreparedStatement hozzavaloTorlese
private PreparedStatement hozzavaloModositas
```

**2 konstruktora** van, egy sima konstruktora, és és egy "SzakacsKonyvDAOJDBCImpl(Connection conn)" ami paraméterként megkapja a "Connection" a main osztályból. Itt beállítja az összes adattagot, és SQLExceptiont dob.

- **receptBeolvas** : kiválaszt mindent a recept táblából ahol a recept fajtája majd megegyezik a kapott fajtával.
- **receptKiir** : hozzáadja a recept táblához a recept(nevét,fajtáját,elkészítését,dátumát)
- **recepTorlese** : kitörli a recept táblából azt a receptet ahol a recept Id-je emegyezik a kapott ID-vel.
- **receptModosit** : felülírja a recept táblában a recept nevét és elkészítését ahol a recept ID-je megegyezik a kapott ID-vel.
- **getID** : kiválasztja azt az ID-t a recept táblából ahol a recept neve és fajtája megegyezik a kapott értékekkel.

- **hozzavalokBeolvas** : kiválaszt mindent a recept és a hozzávalók tábla összekapcsolásából Id alapján ahol Fajta és a recept ID-je megegyezik a kapott értékekkel.
- **hozzavalokKiir** : ki írja a hozzávalók táblába a hozzávalók (darab számát, receptID-jét, recept nevét, mennyiségét, mértékegységét, alapanyagát, árát) a kapott értékekkel.
- **hozzavalokTorlese** : ki törli a hozzávalók táblából azt a hozzávalót ahol a Sora megegyezik a kapott értékkel.
- **hozzavalokModositas** : felülírja a hozzávalók táblában a hozzávaló (Mennyiségét, Mértékegységét, alapanyagát, árát) ahol a Sor megegyezik a kapott értékkel.

#### Metódusai:

- **public List<Recept> beOlvas(String fajta)** - beolvassa az adatbázisból azokat a recepteket amelyek a kapott paraméterként "fajta" megegyezik az adatbázis recept táblájában található receptek fajtájával (a receptek hozzávalóit is beolvassa a hozzávalók táblából), és vissza adja egy List-ben. (SQLExceptiont dob)
- **public List<Hozzavalok> hozzavalokBeolvasasa(Recept recept)** - a paraméterként kapott recept fajtája és a recept id-je alapján be olvassa az adatbázisból az adott recept hozzávalóit és vissza ad egy Hozzavalok listát. (SQLExceptiont dob)
- **public void kiir(Recept recept)** - ki írja az adatbázisba a paraméterként kapott receptet, és a hozzávalókat is a "hozzavalók" táblába. (SQLExceptiont dob)
- **public void hozzavalokKiir(Hozzavalok hozzavalok, Recept recept)** - ki írja a paraméterként kapott hozzávalót az adatbázisba. a "ReceptID"-t és a recept nevét a paraméterként kapott receptből tudjuk meg. (SQLExceptiont dob)
- **public void receptModosit(Recept recept)** - módosítja a paraméterként kapott receptet a megegyező id-vel rendelkező receptet az adatbázisban. (SQLExceptiont dob)
- **public void hozzavalokModositas(Hozzavalok hozzavalok)** - A paraméterként kapott hozzávalóval módosítja az adatbázisban megegyező hozzávalót a "Soruk" alapján. (SQLExceptiont dob)
- **public void receptTorlese(Recept recept)** törli a paraméterként kapott recept ID-vel megegyező receptet az adatbázisból. (SQLExceptiont dob)
- **public void hozzavalokTorlese(Hozzavalok hozzavalok)** - ki törli a paraméterként kapott hozzávalót az adatbázisból. (SQLExceptiont dob)
- **public List<Recept> kereses(String nev, String fajta)** - megkeresi és vissza adja azokat a recepteket az adatbázisból aminek (teljesen vagy részlegesen) a neve és fajtája megegyezik. (SQLExceptiont dob)
- **public List<String> etelFajtak()** - kiolvassa az adatbázisból a receptek(ételek) fajtáit és vissza adja egy listában. (SQLExceptiont dob)
- **public void bezar()** - bezárja az összes adattagot. (SQLExceptiont dob)

### Recept osztály:

#### Adattagjai:

```
private int id
private String nev
private String fajta
private List<Hozzavalok> hozzavalok
private String elkeszítése
private LocalDate felveteliDatum
```

**2 konstruktora** van, egy üres, és egy

- **public Recept(String nev, String fajta, List<Hozzavalo> hozzavalok, String elkeszítése)**

ami beállítja a kapott paraméterrel az adattagokat.

**Metódusai:**

- Az összes adattaghoz **Getter-Setter**, és lesz egy

- **public String toString()** - melyben vissza adja a recept nevét és zárójelek között a felviteli dátumát.

## **Hozzavalo osztály:**

**Adattagjai:**

private int **darab**

private int **sor**

private Double **mennyiség**

private String **merkekegyseg**

private String **alapanyag**

private int **ar**

**2 konstruktora van:** egy üres, és egy

**public Hozzavalo(double mennyiség, String merkekegyseg, String alapanyag, int ar)** - ami beállítja a kapott paraméterrel az adattagokat.

**Metódusok:**

- Minden adattagnak **Gettere és Settere**

- **public String toString()** - egy (Override metódus) ami vissza adja a hozzávaló mennyiségét, mértékegységét, alapanyagát és zárójelben ( árát FT-ba). Ezen kívül megvizsgálja hogy a kapott mennyiség egész szám e, ha igen akkor int értékké alakítja és azt írja ki.

## Foablak (Main) osztály:

ez egy JFrame osztály melynek a kinézete:

Menü

Kérem válassza ki a Menüből ,hogymilyen fajta ételeket szeretne megnézni.

A keresett recept neve:  Keresés

Kiválasztva:

A recept neve és felvételi dátuma:(yyyy-HH-dd)

Recept elkészítése

Új recept hozzáadása

Recept módosítása

Recept törlése

Kilépés

## Az osztály adattagjai:

- private DefaultListModel<Recept> **listModel** = new DefaultListModel<>();
- private String **fajta**;
- private SzakacsKonyvDAO **adatbazisKezelo**;

## A foablak konstruktora :

- **public Foablak()** A konstruktorban beállítja az ablak elhelyezését középre (setLocationRelativeTo(null)). Létrehoz egy adatbáziskapcsolatot a megfelelő beállításokkal és az adattagban található "**adatbazisKezelo**"-nek beállítja ezt. Ha valami gond lépne fel akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Hiba történt az adatbáziskapcsolatban! Üzenet: " + ex.getMessage()**", az ablak neve: "**Hiba!**", Error ikonnal.

Ezen felül beállítja a "**receptLista**" modeljét a "**listModel**"-el.

## Komponensek:

- az ablak bal felső sarkába van egy gomb aminek a szövege: "**Menü**" Segoe UI 14-es Bold-ra van méretezve. A programban használt neve "btMenu" és van egy 'Action' eseménye.
- a "Menü" alatt van egy címke aminek a szövege: "**Kérem válassza ki a Menüből ,hogymilyen fajta ételeket szeretne megnézni.**" 14-esre méretezve és elhalványítva a színét [153,153,153].
- Alatta van egy másik címke aminek a szövege: "**A keresett recept neve:**" 14-esre méretezve
- a "**A keresett recept neve:**" mellett van egy szövegmező ami üres, a benne megjelenő szöveg mérete 14-es, a programban használt neve: "tfKeresettReceptNev"
- a "tfKeresettReceptNev" szövegmező mellett van egy gomb aminek a szövege: "Keresés" 14-esre méretezve, a programban használt neve: "btKereses" aminek van egy 'Action' eseménye.
- a "**A keresett recept neve:**" alatt van egy címke aminek a szövege: "**Kiválasztva:**" aminek a mérete 14-es
- a "**Kiválasztva:**" címke mellett van egy másik címke ami üres, a benne megjelenő szöveg mérete 18-as Bold, a programban használt neve: "lbKivalasztott".
- a "**Kiválasztva:**" címke alatt van még egy címke aminek a szövege: "**A recept neve és felviteli dátuma:(yyyy-HH-dd)**" 14-esre méretezve és elhalványítva a színét [153,153,153].
- a "**A recept neve és felviteli dátuma:(yyyy-HH-dd)**" alatt van egy lista aminek a programban használt neve: "receptLista" a benne megjelenő Recepteknek a mérete 14-es, a típusa: <Recept>.
- a "receptLista" mellett van 5 gomb egymás alatt melyeknek a szövege("Recept elkészítése", "Új recept hozzáadása", "Recept módosítása", "Recept törlése", "Kilépés") 14-esre méretezve, ezeknek a programban használt nevük: ("btElkeszites", "btUjReceptHozzaadas", "btReceptModositas", "btTorles", "btKilepes") és mindnek van egy 'Action' eseménye.

## Metódusok:

- **private void btMenuActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a "**Menü**" gombra kattint akkor készít egy új MenuDialog objektumot "**menu**" néven és ezáltal meghívja a "**MenuDialog(this, true, adatbazisKezelo)**" konstruktorát, amiben paraméterként megadtuk az adatbázisban megtalálható "**adatbazisKezelo**" adattagot. Ezután a MenuDialog ablak nevét beállítja "**Menü**"-re, majd megjeleníti az ablakot. Ezután meghívja a "**menu.getOke()**"-t és megnézi hogy "true"-e, ha igen akkor az adatbázisban megtalálható "**fajta**" adattagot beállítja a "**menu.getFajta()**" metódus visszatérési értékével, és beállítja az "**lbKivalasztott**" szövegét a "**fajta**" adatbázisban tárolt szöveggel. Ezután meghívjuk a "receptKiir()" metódust. Ha valami gond lenne benne akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Hiba a " + this.fajta + " beolvasása közben!**", az ablak neve: "**Hiba!**", Error ikonnal.
- **private void btKilepesActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a "Kilépés"

gombra kattint akkor meghívjuk az adatbázisban található "adatbázisKezelo" bezar() metódusát. Ha valami gond lenne benne akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Hiba az adatbázis bezárása közben!", az ablak neve: "Hiba!", Error ikonnal. Ezután kilép a programból.

- **private void btUjReceptHozzaadasActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó az "Új recept hozzáadása" gombra kattint akkor készít egy új HozzavaloDialog objektumot "hozzaAd" néven és ezáltal meghívja a "ModositoDialog(this, true, this.adatbázisKezelo)" konstruktorát, amiben paraméterként megadtuk az adatbázisban megtalálható "adatbázisKezelo" adattagot, ezután megjeleníti az ablakot. Ezután készít egy új Recept objektumot "ujRecept" néven ami megkapja a "hozzaAd.getRecept()" metódus meghívásával a visszatérési értékét.

Ha az "ujRecept"-ben van recept akkor meghívja a "receptKiir()" metódust. Ha valami gond lenne benne akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Hiba a " + lbKivalasztott.getText() + " hozzáadása közben!", az ablak neve: "Hiba!", Error ikonnal.

- **private void btReceptModositasActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a "Recept módosítása" gombra kattint akkor meghívja a "kivalasztEllenor()" metódust, ha "true"-val térsz vissza akkor megvizsgálja hogy van e kiválasztva recept, ha nincs akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Kérem válassza ki melyik receptet szeretné módosítani!", az ablak neve: "Értesítés!", Information ikonnal.

Ellenkező esetben ha ki van választva recept akkor készít egy új HozzavaloDialog objektumot "modosit" néven és ezáltal meghívja a "ModositoDialog(this, true, receptLista.getSelectedValue(), adatbázisKezelo)" konstruktorát, amiben paraméterként megadtuk az adatbázisban megtalálható "adatbázisKezelo"-t és a kiválasztott receptet a listából. Ezután megjeleníti az ablakot.

Ezután meghívja a "receptKiir()" metódust. Ha valami gond lenne benne akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Hiba a " + modosit.getRecept() + " recept módosítása közben!", az ablak neve: "Hiba!", Error ikonnal.

- **private void btElkeszertesActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó az "Recept elkészítés" gombra kattint akkor meghívja a "kivalasztEllenor()" metódust, ha "true"-val térsz vissza akkor megvizsgálja hogy van e kiválasztva recept, ha nincs akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Kérem válassza ki melyik receptet szeretné módosítani!", az ablak neve: "Értesítés!", Information ikonnal.

Ellenkező esetben ha ki van választva recept akkor egy String típusú "szam" változóba elmentjük a felhasználó által bevitt értéket egy 'showInputDialog' segítségével, a szöveg ami megjelenik: "Kérem írja be számmal hány adagot szeretne elkészíteni!", az ablak neve: "Kérdés", Question ikonnal.

Ezután egy int változóba "adag" névvel integerré alakítja a "szam" értékét. Ha valami gond lenne vele akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Kérem egész számmal adja meg az adagot!", az ablak neve: "Hiba!", Error ikonnal.

Ezután készít egy új ElkeszertesDialog objektumot "elkeszit" néven és ezáltal meghívja a "ElkeszertesDialog(this, true, receptLista.getSelectedValue(), adag)" konstruktorát, amiben paraméterként megadtuk a kiválasztott receptet és az adagot, majd előhossa az ElkeszertesDialog ablakot.

- **private void btReceptTorlesActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a "Recept törlése" gombra kattint akkor meghívja a "kivalasztEllenor()" metódust, ha "true"-val térsz vissza akkor megvizsgálja hogy van e kiválasztva recept, ha nincs akkor egy 'showMessageDialog' jelenik meg

ezzel a szöveggel: "Kérem válassza ki melyik receptet szeretné törölni!", az ablak neve: "Értesítés!", Information ikonnal.

Ellenkező esetben pedig egy 'showConfirmDialog' jelenik meg ezzel a szöveggel: "Biztos törölni szeretnéd " + `receptLista.getSelectedValue()` + " receptet?", az ablak neve: "Megerősítés!", Question ikonnal.

ezután megvizsgálja hogy az (int típusú) eredmény változó "Yes" vagy "No", ha Yes akkor készít egy új Recept objektumot "recept" névvel ami megkapja a kiválasztott receptet, majd meghívja a "`adatbazisKezelo.receptTorlese(recept)`" metódust amiben paraméterben megadja a "recept" objektumot. Ha valami gond lenne vele akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Hiba történt a " + `recept.getNev()` + " törlése közben!", az ablak neve: "Hiba!", Error ikonnal.

Ezután meghívja a "`receptKiir()`" metódust.

- **private void btKeresesActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a "Keresés" gombra kattint akkor meghívja a "`kivalasztEllenor()`" metódust, ha "true"-val térsz vissza akkor egy String típusú "nev" változóba elmenti a keresett nevet, majd készít egy új listát receptekkel "List<Recept> **receptek**" néven ami megkapja a "`adatbazisKezelo.kereses(nev, fajta)`" visszatérési értékét. Paraméterben megkapta a "nev" változót és az adatban megtalálható "fajta" adattagot. Ha valami gond lenne vele akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Hiba történt a " + nev + " keresése közben!", az ablak neve: "Hiba!", Error ikonnal.

Ezután megvizsgálja hogy a "**receptek**" lista üres-e, ha igen akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Nem találtam " + nev + " nevű receptet a " + `lbKivalasztott.getText()` + " között!", az ablak neve: "Értesítés!", Information ikonnal, majd meghívja a "`receptKiir()`" metódust.

Ha nem üres a "**receptek**" lista akkor meghívja a "`listModel.clear()`" metódust és ki írja a "**receptLista**"-ba a "receptek"-ben található recepteket.

- **private Boolean kivalasztEllenor()** Ez a metódus ellenőrzi hogy a "`lbKivalasztott`" címke üres e, ha igen akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Kérem válassza ki a Menüben hogy milyen fajta étellel szeretné elvégezni a műveletet.", az ablak neve: "Értesítés!", Information ikonnal, majd "false"-t ad vissza.

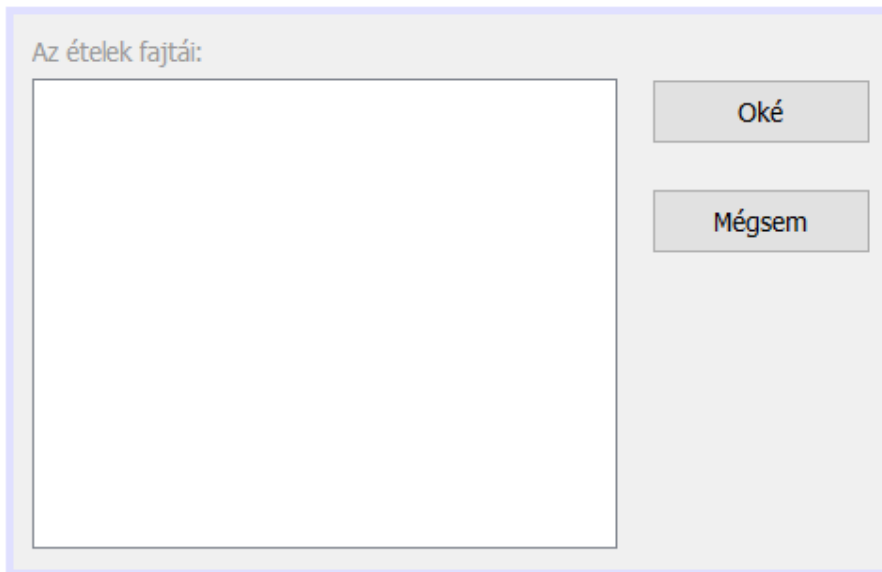
Ellenkező esetben ha nem üres akkor "true"-t ad vissza.

- **private void receptKiir()** Ez a metódus meghívja a "`listModel.clear()`" metódust, majd készít egy új listát "List<Recept> **receptek**" néven és értéként megkapja az "`adatbazisKezelo.beOlvas(fajta)`" metódusának a visszatérési értékét. Paraméterként megadja az adatban megtalálható "fajta" adattagot. Ezután ki írja a "**receptLista**"-ba az összes receptet ami a "**receptek**" listába van.



## MenuDialog osztály:

ez egy JDialog osztály melynek kinézete:



### Az osztály adattagjai:

```
private DefaultListModel<String> listModel = new DefaultListModel<>();  
private static SzakacsKonyvDAO adatbazisKezelo;  
private boolean oke;  
private String fajta;
```

#### 1 konstruktora van:

**public MenuDialog(java.awt.Frame parent, boolean modal, SzakacsKonyvDAO adatbazisKezelo)** - ami beállítja az adattagokat a kapott paraméterekkel, az ablak helyzetét (setLocationRelativeTo(null)). Beállítja a lista modeljét az adattagban megtalálható "listModel"-el, a programban használt neve a listának: "etelekFajtai", és meghívja az osztály "beOlvas()" -át. Ha valami probléma lépne fel az adatbázis beolvasása közben akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Hiba az ételek fajtája beolvasása közben!", az ablak neve: "Hiba!", Error ikonnal. Plusz még tartalmazza a magától legenerált metódusokat.

### Komponensek:

- az ablak bal felső részén van egy címke aminek a szövege: "Az ételek fajtái:" ami Tahoma 14-esre van méretezve és elhalványítva a színét [153,153,153].

- az "Az ételek fajtái" címke alatt van egy lista, a benne megjelenő szöveg mérete Tahoma 14-es, a programban használt neve: "etelekFajtai", a típus paramétere pedig String.

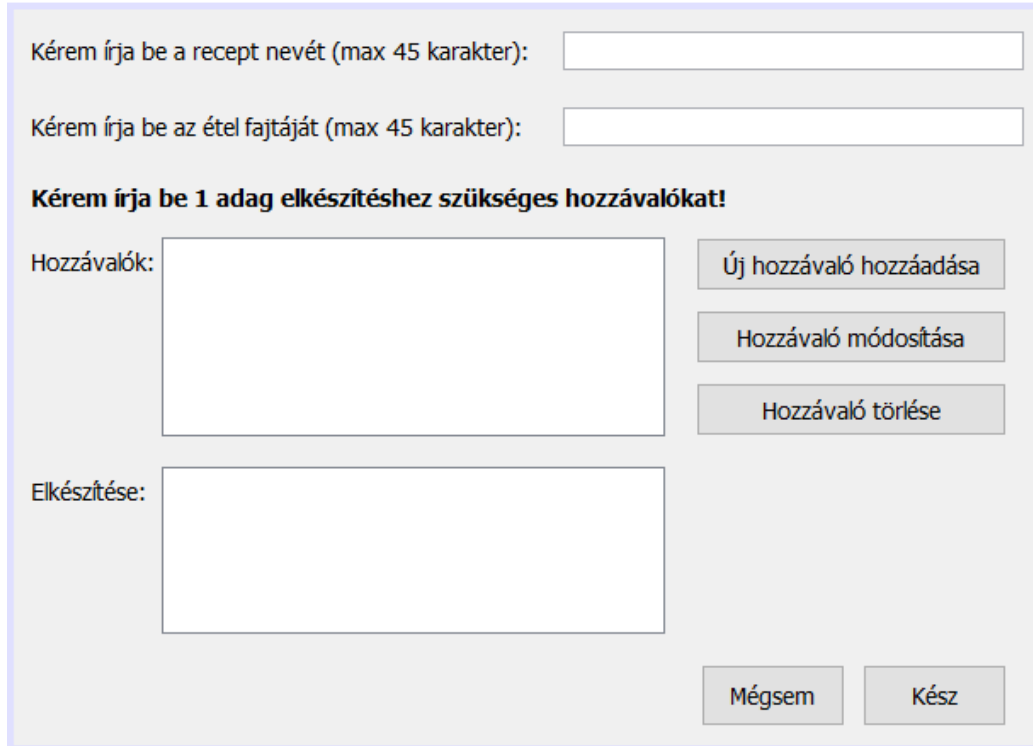
- a lista mellett jobbra 2 gomb van, az első gombnak a neve: "Oké", a programban használt neve: "btOke". Alatta a második gomb: "Mégsem", a programban használt neve: "btMegsem", mindkettő Tahoma 14-esre méretezve, és mindkettőnek van egy 'Action' eseménye.

## Metódusok:

- **private void btOkeActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó az "Oké" gombra kattint akkor először meghívja az osztály "**kivalasztEllenor()**"-át. Ha az "**oke**" adattag értéke 'true' akkor az adattagban megtalálható "**fajta**" értékét beállítja az adattagként megtalálható "**etelekFajtai**" listából kiválasztott értékkel, majd bezárja az ablakot.
- **private void btMegsemActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a "Mégsem" gombra kattint akkor bezárul az ablak.
- **private void beOlvas()** Ez a metódus létrehoz egy String típusú listát "**lista**" néven ami megkapja az adattagként megtalálható "**adatbazisKezelo**" etelFajtak()-ának visszatérési értékét. Megvizsgálja hogy a "**lista**" üres-e, ha igen akkor az "**etelekFajtai**" listában megjelenő szöveg színét beállítja (153, 153, 153)-ra és beállítja az 'elérését' false-ra, majd ki írja a listába a következő szöveget: "**Egyenlőre nincs az adatbázisban recept, adjon hozzá új receptet!**".  
Ha a "**lista**" nem üres akkor ki írja a "**lista**"-ban szereplő étel fajtaikat az "**etelekFajtai**" listába. (SQLException-t dob).
- **private void kivalasztEllenor()** Ez a metódus készít egy String "**fajta**" változót ami az adattagként megtalálható "**etelekFajtai**" listából kiválasztott értékét kapja meg. Megvizsgálja hogy a "**fajta**" változóba van e valami, ha nincs akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Kérem válassza ki a Menüben hogy milyen fajta ételeket szeretne megnézni vagy kattintson a 'Mégsem' gombra!**", az ablak neve: "**Értesítés!**", Information ikonnal, emellett még beállítja az adattagként megtalálható boolean "**oke**" adattagot false-ra.  
Ha van akkor az adattagként megtalálható boolean "**oke**" adattagot true-ra állítja.
- **public String getFajta()** Ez a metódus vissza adja az adattagként megtalálható String "**fajta**" adattagot.
- **public boolean getOke()** Ez a metódus vissza adja az adattagként megtalálható boolean "**oke**" adattagot.

## ModositoDialog osztály:

ez egy JDialog osztály melynek kinézete:



### Az osztály adattagjai:

```
private static String fajta;  
private static SzakacsKonyvDAO adatbazisKezelo  
private Recept recept  
private List<Hozzavalo> hozzavalok = new ArrayList<>()  
private int darab = 0;  
private DefaultListModel<Hozzavalo> listModel = new DefaultListModel<>()
```

### 2 konstruktora van:

- **public ModositoDialog(java.awt.Frame parent, boolean modal, SzakacsKonyvDAO adatbazisKezelo)** ami beállítja a megfelelő adattagot a kapott paraméterrel, az ablak helyzetét (setLocationRelativeTo(null)). Ezután a ModositoDialog ablak nevét beállítja "Új recept hozzáadása"-ra. Beállítja a lista modeljét az adattagban megtalálható "listModel"-el, a programban használt neve a listának: "hozzavalokLista". plusz még tartalmazza a magától legenerált metódusokat.

- **public ModositoDialog(java.awt.Frame parent, boolean modal, Recept recept, SzakacsKonyvDAO**

**adatbazisKezelo**) tartalmazza a magától legenerált metódusokat, az ablak helyzetét beállítja középre (setLocationRelativeTo(null)). Ezután a ModositoDialog ablak nevét beállítja **"Recept módosítása"**-ra, beállítja a lista modeljét az adattagban megtalálható **"listModel"**-el, a programban használt neve a listának: **"hozzavalokLista"**, beállítja az adattagokat a kapott paraméterekkel.

A **"tfReceptNev"** szövegmező szövegét beállítja a paraméterben kapott recept nevével.

A **"tfEtelFajta"** szövegmező szövegét beállítja a paraméterben kapott recept fajtájával.

Az adattagban megtalálható **"hozzavalok"** listát beállítja a paraméterben kapott recept hozzávalók listájával.

Beállítja az adattagban megtalálható **"int darab"** adattagot a hozzávalókból megkapható **"getDarab()"** metódust használva a megkapott számmal.

Meghívja a **"hozzavalokKiir()"** metódust.

Beállítja a **"taElkeszítése"** szövegmező szövegét a receptből megkapható elkészítése szöveggel.

## Komponensek:

- az ablak bal felső részén van egy címke a következő szöveggel: **"Kérem írja be a recept nevét (max 45 karakter):"**, a benne megjelenő szöveg 14-esre van méretezve.

- A **"Kérem írja be a recept nevét (max 45 karakter):"** címke mellett van egy szövegmező aminek a programban használt neve: **"tfReceptNev"** a benne megjelenő szöveg mérete 14-es, a felhasználó ide írhatja az új/modosított recept nevét.

- a **"Kérem írja be a recept nevét (max 45 karakter):"** címke alatt van egy másik címke aminek a szövege: **"Kérem írja be az étel fajtáját (max 45 karakter):"** 14-es Bold-ra méretezve.

- az **"Kérem írja be az étel fajtáját (max 45 karakter):"** címke mellett van egy szövegmező aminek a programban használt neve: **"tfEtelFajta"** a benne megjelenő szöveg mérete 14-es, a felhasználó ide írhatja az új/modosított étel fajtájának a nevét.

- az **"Kérem írja be az étel fajtáját (max 45 karakter):"** címke alatt van egy másik címke aminek a szövege: **"Kérem írja be 1 adag elkészítéshez szükséges hozzávalókat!"** 14-es Bold-ra méretezve.

- a **"Kérem írja be 1 adag elkészítéshez szükséges hozzávalókat!"** címke alatt van egy másik címke aminek a szövege: **"Hozzávalók:"** 14-esre méretezve.

- a **"Hozzávalók:"** címke mellett van egy lista a programban használt neve: **"hozzavalokLista"** aminek a típusa: **"Hozzavalok"** és a listában megjelent hozzávalók mérete Tahoma 14.

- a **"hozzavalokLista"** mellett van egy gomb aminek a szövege: **"Új hozzávaló hozzáadása"** 14-esre méretezve, aminek a programban használt neve: **"btUjHozzavaló"**, és ennek van egy 'Action' eseménye.

- a **"btUjHozzavaló"** gomb alatt van egy másik gomb aminek a szövege: **"Hozzávaló módosítása"** 14-esre méretezve, aminek a programban használt neve: **"btHozzavalóModosito"**, és ennek is van egy 'Action' eseménye.

- a "btHozzavaloModosito" gomb alatt van még egy gomb aminek a szövege: "Hozzávaló törlése" 14-esre méretezve, aminek a programban használt neve: "btHozzavaloTorlese", és ennek is van egy 'Action' eseménye.

- az ablak bal oldalán a "hozzavalokLista" alatt a "Hozzávalók:" címkével egyvonalban van egy címke aminek a szövege: "Elkészítése:" 14-esre méretezve.

- az "Elkészítése:" címke mellett van egy JTextArea aminek a programban használt neve: "taElkeszetese" aminek a benne megjelenő szöveg mérete 14-es.

- az ablak jobb alsó részében van egy gomb aminek a szövege: "Mégsem" 14-es, a programban használt neve: "btMegsem" aminek van egy 'Action' eseménye.

- a "Mégsem" gomb mellett van egy másik gom aminek a szövege: "Kész" 14-es, a programban használt neve: "btKesz" aminek van egy 'Action' eseménye.

## Metódusok:

- **private void btUjHozzavaloActionPerformed(java.awt.event.ActionEvent evt)** ha a felhasználó az "Új hozzávaló hozzáadása" gombra kattint akkor a "darab" adattag értéke 1-el növekszik.

Készít egy HozzavaloDialog objektumot "hozzaAd" néven : "HozzavaloDialog hozzáAd = new HozzavaloDialog(this,true)".

Beállítja a HozzavaloDialog ablak nevét: "Új hozzávaló hozzáadása", és megjeleníti az ablakot.

Megvizsgálja hogy a "hozzaAd.getHozzavalo() == null" -e ha igen akkor nem csinál semmit csak return-el vissza lép. Egyébként készít egy új "Hozzavalo hozzavalo" objektumot ami megkapja a hozzáAd.getHozzavalo() metódus visszatérési értékét.

A "hozzavalo" objektumnak beállítja a "darab"-ját az adattagban megtalálható "darab" értékével.

Megvizsgálja hogy a ModositoDialog ablak neve "Recept módosítása"-e, ha igen akkor az adattagban megtalálható "adatbazisKezelo"-nek meghívja a "hozzavalokKiir(hozzavalo,recept)" metódusát és

paraméterben megadja a "hozzavalo"-t és az adattagban megtalálható "recept"-et. Ezt követően az adattagban megtalálható "hozzavalok" lista megkapja az "adatbazisKezelo.hozzavaloBeolvasasa(recept)" (amiben megadjuk paraméterben az adattagban található "recept"-et) metódusának a visszatérési értékét,

ezután meghívja a "hozzavalokKiir()" metódust. Ha valami probléma lépne fel az adatbázis beolvasása közben akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Hiba az új " + hozzavalo + " hozzávaló adatbázishoz való hozzáadás közben!", az ablak neve: "Hiba!", Error ikonnal.

Ha a ModositoDialog neve nem "Recept módosítása" akkor az adattagban megtalálható "hozzavalok" listához hozzá adja a "hozzavalo"-t és meghívja a "hozzavalokKiir()" metódust.

- **private void btHozzavaloModositoActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a "Hozzávaló Módosítása" gombra kattint akkor megvizsgálja hogy ki van e választva egy hozzávaló, ha nincs akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Kérem válassza ki melyik hozzávalót szeretné módosítani!", az ablak neve: "Értesítés!", Information ikonnal. Ellenkező esetben készít egy új objektumot a HozzavaloDialog osztályhoz "modosit" néven amivel a "HozzavaloDialog(this, hozzavalokLista.getSelectedValue())" konstruktorát hívja meg és paraméterként megadjuk a kiválasztott "hozzavalo"-t.

Beállítja a HozzavalDialog ablak nevét: "**Hozzávaló módosítása**"-ra és megjeleníti az ablakot.

Készít egy új "**Hozzaval**" objektumot "**hozzaval**" néven aminek az értékét beállítja a "**modosit.getHozzaval()**" visszatérési értékével.

Megvizsgálja hogy a ModositoDialog ablak neve: "Recept módosítása"-e ha igen akkor meghívja az "**adatbazisKezelo.hozzavalModositasa(hozzaval)**" metódust (amiben megadjuk paraméterben az új "hozzaval" objektumot) Ezt követően az adattagban megtalálható "**hozzavalok**" lista megkapja az "**adatbazisKezelo.hozzavalBeolvasasa(recept)**" (amiben megadjuk paraméterben az adattagban található "recept"-et) metódusának a visszatérési értékét, ezután meghívja a "**hozzavalokKiir()**" metódust. Ha valami probléma lépne fel az adatbázis beolvasása közben akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Hiba az új " + hozzaval + " hozzávaló adatbázishoz való hozzáadás közben!**", az ablak neve: "**Hiba!**", Error ikonnal.

Ha a ModositoDialog ablak neve nem ez "**Hozzávaló módosítása**" akkor az adattagban megtalálható "hozzavalok" listában megkeresi azt a hozzávalót amelyiknek a "darab" száma megegyezik az új "hozzaval" objektum darabszámával és annak a hozzávalónak a listában felülírja az adattagjait. Ezután az adattagban megtalálható hozzavalokLista "ListModel"-el módosítja a a megfelelő hozzávalót a listában.

- **private void btKeszActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a kész gombra kattint meghívja a "**receptEllenorzes()**" metódust ha igazzal tér vissza akkor megvizsgálja hogy a "**ModositoDialog**" ablaknak a neve : "**Recept módosítása**"-e ha igen akkor az adattagban megtalálható recept(nevét,fajtáját,hozzávalóit,elkészítését) módosítja a ("**tfReceptNev**", a "**tfEtelFajta**", a "**hozzavalok**" adattaggal, és a "**taElkesztese**")-ben található értékekkel.

Ezután meghívja az "**adatbazisKezelo.receptModosit(recept)**" metódusát amiben paraméterként megkapja a "**recept**" adattagot, és bezárja az ablakot. Ha valami gond lenne az adatbázis módosítása közben akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Hiba a " + recept.getNev()+ " recept módosítása közben!**", az ablak neve: "**Hiba!**", Error ikonnal.

Ha a "**ModositoDialog**" ablaknak a neve nem : "**Recept módosítása**" akkor az adattagban megtalálható "**fajta**" értékét beállítja a "**tfEtelFajta**" értékével, és az adattagban megtalálható "recept"-nek beállítja a (nevét,fajtáját,hozzávalóit és elkészítését) a ("**tfReceptNev**", a "**fajta**" adattaggal ,a "**hozzavalok**" adattaggal, és a "**taElkesztese**")-ben található értékekkel, és beállítja a felveteli dátumát is(akkor amikor épp felveszi : ).

Ezután meghívja a "**adatbazisKezelo.kiir(recept)**" metódust amiben paraméterként megadjuk a "**recept**"-et, és bezárja az ablakot. Ha valami gond lenne az adatbázis hozzáadása közben akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Hiba a " + recept.getNev()+ " recept adatbázishoz való hozzáadása közben!**", az ablak neve: "**Hiba!**", Error ikonnal.

- **private void btMegsemActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a "Mégsem" gombra kattint akkor bezárja az ablakot. (nem történik meg az új recept hozzáadása/módosítása, illetve a módosításnál a hozzávalók módosítása megtörténik ha a mégse gombra kattint a felhasználó, de ha elront valamit újra tud (módosítani,hozzáadni,törölni). )

- **private void btHozzavalTorleseActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a hozzávaló törlésére kattint akkor megvizsgálja hogy van e kiválasztva hozzávaló, ha nincs akkor akkor egy

'showMessageDialog' jelenik meg ezzel a szöveggel: "**Kérem válassza ki melyik hozzávalót szeretné törölni!**", az ablak neve: "**Értesítés!**", Information ikonnal. Ellenkező esetben pedig egy 'showConfirmDialog' jelenik meg ezzel a szöveggel: "**Biztos törölni szeretnéd " + hozzavalokLista.getSelectedValue() + " hozzávalót?"**", az ablak neve: "**Megerősítés!**", Question ikonnal. ezután megvizsgálja hogy az (int típusú) eredmény változó "Yes" vagy "No", ha Yes akkor készít egy új Hozzavalo objektumot "**hozzavalo**" néven és értéként a kiválasztott hozzávalót adjuk meg neki.

Ha a ModositoDialog ablak neve : "**Recept módosítása**" akkor megvizsgálja hogy a "hozzavalokLista"-ban az utolsó hozzávalót akarja e a felhasználó törölni, ha igen akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**A hozzávalók listájának utolsó hozzávalóját nem törölheti! Kérem használja a módosítást!**", az ablak neve: "**Értesítés!**", Information ikonnal. Ellenkező esetben meghívja az "**adatbazisKezelo.hozzavaloTorlese(hozzavalo)**" metódusát paraméterként átadva neki a hozzávalót, ezután a "**hozzavalok**" adattagnak megadja a "**adatbazisKezelo.hozzavaloBeolvasasa(recept)**" vissztérési értékét és meghívjuk a "**hozzavalokKiir()**" metódust. Ha valami gond lenne az adatbázisban lévő hozzávaló törlése közben akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Hiba a " + hozzavalo + " hozzávaló törlése közben!**", az ablak neve: "**Hiba!**", Error ikonnal.

Ha a ModositoDialog ablak neve nem : "**Recept módosítása**" akkor ugyanúgy megvizsgálja a "hozzavalok" adattag méretét, ha megegyezik 1-el akkor ki írja ugyanazt a szöveget "**A hozzávalók listájának utolsó hozzávalóját nem törölheti! Kérem használja a módosítást!**" ugyanolyan formában, ellenkező esetben kitörli a "**hozzavalok**" adattagból azt a hozzávalót amelynek a "**darab**" száma megegyezik a törölni kívánt hozzávaló darabszámával.

- **public Recept getRecept()** ez a metódus vissza adja az adattagban megtalálható "recept"-et.

- **public void hozzavalokKiir()** ez a metódus ki írja a "**hozzavalokLista**"-ba az adattagként megtalálható "**hozzavalok**"-at.

- **private Boolean receptEllenorzes()** ez a metódus leellenőrzi hogy a "**tfReceptNev**"-ben van e valami és hogy 45-karakternél kisebb e a szöveg hossza. Ha nincs benne semmi vagy nagyobb akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**A 'Receptnév' mező vagy üres vagy nagyobb 45 karakternél! Kérem módosítsa vagy kattintson a 'Mégsem' gombra!**", az ablak neve: "**Hiba!**", Warning ikonnal, és " return false"-t ad vissza.

Leellenőrzi hogy a "**tfEtelfajta**"-ban van e valami és hogy 45-karakternél kisebb e a szöveg hossza. Ha nincs benne semmi vagy nagyobb akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Az 'Ételfajta' mező vagy üres vagy nagyobb 45 karakternél! Kérem módosítsa vagy kattintson a 'Mégsem' gombra!**", az ablak neve: "**Hiba!**", Warning ikonnal, és " return false"-t ad vissza.

Továbbá ha a ModositoDialog ablak neve nem : "**Recept módosítása**" akkor létre hogy egy új listát "List<Recept> receptek" néven és értéként megkapja az "**adatbazisKezelo.beOlvas(fajta)**" metódust vissztérési értékét, ha valami gond lenne a beolvasás közben akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Hiba a " + fajta + " receptek beolvasása közben!**", az ablak neve: "**Hiba!**", Error ikonnal, és " return false"-t ad vissza.

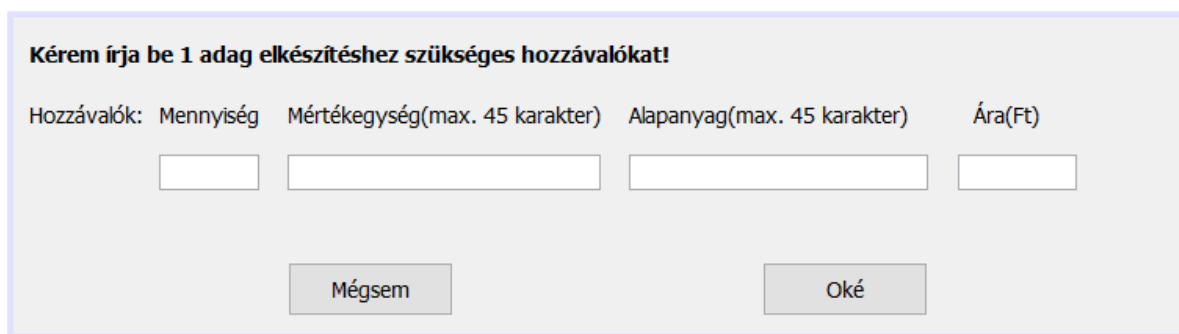
Ezután megnézi hogy van e a receptek között már olyan nevű recept amit a "tfReceptNev"-ben megadtak és ha igen akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Már létezik egy ilyen recept! Módosítsa kérem a recept nevét vagy kattintson a 'Mégsem' gombra!", az ablak neve: "Hiba!", Warning ikonnal, és " return false"-t ad vissza.

Ellenőrzi még hogy a "listModel" üres-e, ha igen akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Kérem adjon hozzá hozzávalókat vagy kattintson a 'Mégsem' gombra!", az ablak neve: "Értesítés!", Information ikonnal, és " return false"-t ad vissza.

Ellenőrzi még a "taElkeszetese" szövegmezőt hogy üres-e, ha igen akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "Kérem töltsse ki az 'Elkészítése' mezőt vagy kattintson a 'Mégsem' gombra!", az ablak neve: "Értesítés!", Information ikonnal, és " return false"-t ad vissza.

## HozzavaloDialog osztály:

ez egy JDialog osztály melynek kinézete:



## Az osztály adattagjai:

private Hozzavalo **hozzavalo**

### 2 konstruktora van:

- **public HozzavaloDialog(java.awt.Dialog parent, boolean modal)** - ami beállítja az ablak helyzetét (**setLocationRelativeTo(null)**) és tartalmazza azokat a metódusokat amit magától legenerál.

- **public HozzavaloDialog(java.awt.Dialog parent, Hozzavalo hozzávalo)** - ami beállítja a "hozzavalo" adattagot a paraméterben kapott hozzávalóval, a "tfMennyiség" szövegmező szövegét beállítja a hozzávalo-ból kinyerhető mennyiséggel, a "tfMertekegyseg" szövegmező szövegét beállítja a hozzávalo-ból kinyerhető mértékegységgel, a "tfAlapanyag" szövegmező szövegét beállítja a hozzávalo-ból kinyerhető alapanyaggal, a "tfAr" szövegmező szövegét beállítja a hozzávalo-ból kinyerhető Árral, illetve tartalmazza azokat a metódusokat amit magától legenerál, és beállítja az ablak helyzetét (**setLocationRelativeTo(null)**).

## Komponensek:

- Az ablak tetején van egy címke aminek a szövege: "Kérem írja be 1 adag elkészítéshez szükséges hozzávalókat!" 14-es Bold-ra méretezve a benne megjelenő szöveg méretét. .

- Alatta az ablak bal oldali szélétől egy címke aminek a szövege: "Hozzávalók:" 14-esre méretezve a szöveg méretét.



- mellette még egy címke aminek a szövege: "**Mennyiség**" 14-esre méretezve a szöveg méretét.
- mellette még egy címke aminek a szövege: "**Mértékegység(max. 45 karakter)**" 14-esre méretezve a szöveg méretét.
- mellette még egy címke aminek a szövege: "**Alapanyag(max. 45 karakter)**" 14-esre méretezve a benne megjelenő szöveg méretét.
- mellette még egy címke aminek a szövege: "**Ára(Ft)**" 14-esre méretezve a szöveg méretét.
- A "Mennyiség" címke alatt egy szerkeszthető szövegmező aminek a programban használt neve : "**tfMennyiség**" 14-esre méretezve a benne megjelenő szöveg méretét.
- A "Mértékegység..." címke alatt egy szerkeszthető szövegmező aminek a programban használt neve : "**tfMertekegyseg**" 14-esre méretezve a benne megjelenő szöveg méretét.
- Az "Alapanyag..." címke alatt egy szerkeszthető szövegmező aminek a programban használt neve : "**tfAlapanyag**" 14-esre méretezve a benne megjelenő szöveg méretét.
- Az "Ára(Ft)" címke alatt egy szerkeszthető szövegmező aminek a programban használt neve : "**tfAr**" 14-esre méretezve a benne megjelenő szöveg méretét.
- Az ablak alján van egy gomb aminek a szövege: "**Mégsem**" 14-esre méretezve, a programban használt neve : "**btMegsem**", aminek van egy 'Action' eseménye.
- A "Mégsem" gomb mellett van még egy gomb aminek a szövege: "**Oké**" 14-esre méretezve, a programban használt neve: "**btOke**", aminek van egy 'Action' eseménye.

## Metódusok:

- **private void btOkeActionPerformed(java.awt.event.ActionEvent evt)** - Ha a felhasználó rá kattint az oké gombra akkor megvizsgálja hogy a szövegmezők üresek-e, ha igen akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Kérem töltsse ki az üres helyeket vagy nyomja meg a 'Mégsem' gombot!**", az ablak neve: "**Hiba**", Warning ikonnal.

Megvizsgálja hogy a "tfMertekegyseg" és az "tfAlapanyag" szövegmezőjébe bevitt szövegének a mérete nagyobb-e 45-nél, ha igen akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**A "" + tfMertekegyseg.getText() + "" nagyobb 45 karakternél!**", vagy a tfMertekegyseg helyett a "**tfAlapanyag.getText()**" jelenik meg, attól függ melyikbe talált hibát, az ablak neve: "**Hiba**", Warning ikonnal.

Megvizsgálja hogy a "tfMennyiség"-hez beírt szöveg double típusú számnak kell lennie, és nem lehet 0. Ha nem számot kap akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Kérem számot írjon be az 'Mennyiség'-hez**", az ablak neve: "**Hiba**", Warning ikonnal. Ha 0-at írnak be akkor ugyanúgy 'showMessageDialog' jelenik meg csak ezzel a szöveggel: "**Nullát nem adhat a 'Mennyiség'-hez!**".

Megvizsgálja hogy a "tfAr"-ba bírt szöveg int típusú számnak kell lennie. Ha nem egész számot kap akkor egy 'showMessageDialog' jelenik meg ezzel a szöveggel: "**Kérem számot írjon be az 'Mennyiség'-hez**", az ablak neve: "**Hiba**", Warning ikonnal.

Ha minden rendben és a "hozzavalo" adattag üres volt, tehát új hozzávalót akarunk a recepthez adni akkor az értékeket beállítjuk a "hozzavalo" adattagba, Ha már volt valami a "hozzavalo" adattagban, akkor módosítja azt.

Ezek után bezárja az ablakot.

- **private void btMegsemActionPerformed(java.awt.event.ActionEvent evt)** Ha a felhasználó a mégsem gombra kattint az ablak bezár és a hozzavalo módosítást/hozzáadást nem hajtja végre.

- **public Hozzavalok getHozzavalo()** A metódus meghívásakor a "hozzavalo" adattagot kapjuk vissza.

## ElkészetesDialog osztály:

ez egy JDialog osztály melynek kinézete:

The screenshot shows a Java Swing dialog box titled "ElkészetesDialog". It contains two identical sections for recipe ingredients. Each section starts with a label "Recept neve:" followed by a text field. Below this is a label "1 adag elkészítéséhez szükséges:" (1 portion required for preparation). Then there is a label "Hozzávalók:" (Ingredients) followed by a list box. Below the list box is a label "Összesen:" (Total) followed by a text field. Below this is a label "elkészítéséhez szükséges:" (required for preparation). Then there is another label "Hozzávalók:" followed by another list box. Below this is another label "Összesen:" followed by another text field. At the bottom of the dialog is a label "Elkészítése:" (Preparation) followed by a text area. At the very bottom are two buttons: "Vissza" (Back) and "Kilép" (Exit).

## Az osztály adatai:

```
private static Recept recept;  
private static int adag;  
private DefaultListModel<Hozzavalok> alapListModel = new DefaultListModel<>();  
private DefaultListModel<Hozzavalok> modositottListModel = new DefaultListModel<>();
```

### 1 konstruktora van:

```
public ElkeszitesDialog(java.awt.Frame parent, boolean modal, Recept recept, int adag)
```

amiben beállítja az adatakat a paraméterben kapottakkal és a ListModeleket.

A felülről nézve az első listának az elnevezése : **alapHozzavalokLista** - ezt állítja be az alapListModelmel.

A másik listának az elnevezése : **modositottHozzavalokLista** - ezt állítja be modositottListModelmel.

Ezen kívül még beállítja az elkészítésDialog ablak helyzetét középre (Null-ra).

és itt hívjuk meg a **kinezetModosito()** metódust.

## Komponensek:

Első címke: "**Recept neve**"- szöveg van benne 14-esre méretezve, mellette egy másik címke **ami üres** , a benne megjelenő szöveg mérete 18-as Bold és a programban használt neve : "**lbReceptNeve**".

Alatta a következő címke aminek a szövege: "**1 adag**" ez 14 -es Bold-ra van állítva a szöveg mérete, mellette még egy címke a következő szöveggel: "**elkészítéséhez szükséges:**" 14-esre méretezve szöveg méretét.

Alatta a következő címke aminek a szövege: "**Hozzávalók:**" 14-esre méretezve, mellette egy lista "**alapHozzavalokLista**" néven van elmentve a programban, 14-es lesz a mérete a "**Hozzavalok**"-nak benne mert ez a típus paramétere.

Alatta egy újabb címke, a szövege: "**Összesen:**" 14-esre méretezve, mellette egy szövegmező(14esre méretezve a benne megjelenő szöveg méretét. ), aminek a programban használt neve: "**tfÖsszesen**" az editable opciót kivettem.

Alatta egy címke ami üres, a programban használt neve: "**lbAdagElkeszítése**" 14 Bold-ra méretezve a benne megjelenő szöveg méretét. és az Auto Resizingje:Horisontal-ra állítva, mellette egy másik címke aminek a szövege: "**elkészítéséhez szükséges:**" 14-esre méretezve.

Alatta egy címke szövege: "**Hozzávalók:**" 14-esre méretezve, mellette egy lista "**modositottHozzavalokLista**" néven van elmentve a programban, 14-es lesz a mérete a "**Hozzavalok**"-nak benne mert ez a típus paramétere.

Alatta egy újabb címke, a szövege: "**Összesen:**" 14-esre méretezve, mellette egy szövegmező(14esre méretezve a benne megjelenő szöveg méretét. ), aminek a programban használt neve: "**tfModositottOsszesen**" az editable opciót kivettem.

Alatta egy címke aminek a szövege: "**Ekészítése:**" 14-es méretezve, mellette egy textArea melynek a

programban használt neve: "taElkeszetese" a benne lévő szöveg 14-esre méretezve.

alatta jobbra elhelyezve egy gomb aminek a szövege: "Vissza" 14-esre méretezve, a programban használt neve: "btVissza" és van egy 'Action' eseménye, mellette még egy gomb aminek a szövege: "Kilép" 14-esre méretezve, aminek a programban használt neve: "btKilep" és van egy 'Action' eseménye.

## Metódusok:

- **private void btKilepActionPerformed(java.awt.event.ActionEvent evt)** - Ha erre a gomra kattint a felhasználó akkor kilép a programból.

- **private void btVisszaActionPerformed(java.awt.event.ActionEvent evt)** - Ha erre a gombra kattint a felhasználó akkor bezárja ezt az ablakot és vissza lép az előzőre.

- **private void kinezetModosito()** - ezt a metódust a konstruktorban hívjuk meg és a következő a funkciója:

- az "lbReceptNev" címke szövegét beállítja az adattagban elmentett Recept nevével.

- a recept hozzávalóit ki írja az "alapHozzavalokLista"-ba

- a "tfOsszesen" szövegmező szövegét beállítja az összeadott hozzávalók árával, Forintba.

- a "lbAdagElkeszetese" címke szövegét beállítja az adag adattaggal (adag + " adag").

- a recept hozzávalóit ki írja az adaggal megszorozva a "modositottHozzavalokLista"-ba.

- minden hozzávalónak a árát megszorozza az adaggal és a "tfModositottOsszesen" szövegmező szövegét beállítja az új árral.

- a "taElkeszites" szövegmezőt beállítja a recept adattagból meghívott elkészítésében található szöveggel.