Italrecept nyilvántartó

Készítette:
Név: Varga Bence Gyula
Neptun-azonosító: QFFOK2
Laborvezető neve: Csala Bálint

Tartalom

Felhasználói dokumentáció

Feladat

A program italreceptek nyilvántartására, keresésére és szűrésére szolgál ehhez grafikus kezelőfelületet biztosít a felhasználónak. A felhasználónak lehetősége van új recepteket létrehozni az adatbázisba, itt meglehet adni a recept nevét, alapanyagait és opcionálisan egy rövid leírást az elkészítéséről. A recepteket lehet törölni és szerkeszteni is a létrehozás után. Ki lehet listázni az összes receptet ekkor az összes recept név jelenik meg egy listában, majd azokra kattintva megjelennek a kiválasztott recept tulajdonságai (összetevők, elkészítési folyamat) illetve lehetőség van alapanyag alapján keresni az italreceptek között. A program képes egy előre megadott szöveges fájl alapján elkészíteni a megfelelő adatbázist, ilyen fájlt a program is képes létrehozni az aktuális adatbázisról.

Környezet

Minimum rendszerkövetelmény

1. Microsoft Windows 8 vagy későbbi verziója, MacOs 10.13 vagy ennél újabb verzió, bármilyen Linux verzió.
2. 4GB RAM vagy ennél több.
3. Egér és Billentyűzet.

Szükséges szoftverkövetelmény

1. Java 11 vagy ennél újabb JDK/JRE.
2. A garantált tökéletes működéshez Eclipse IDE for Java Developers 4.18-as verzió vagy ennél újabb.
3. (Vagy bármilyen más Java futtatására alkalmas IDE.)

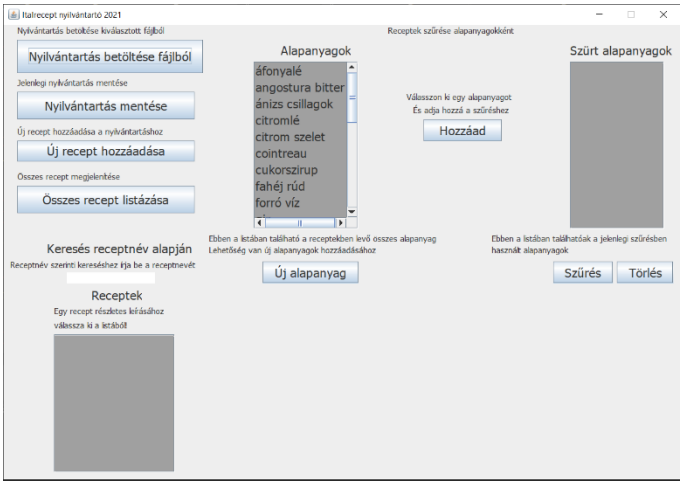
Használat

Program indítása

Az italrecept/src/main mappa megnyitása, majd az általunk kiválasztott környezetben a Main.java fájl elindítása.

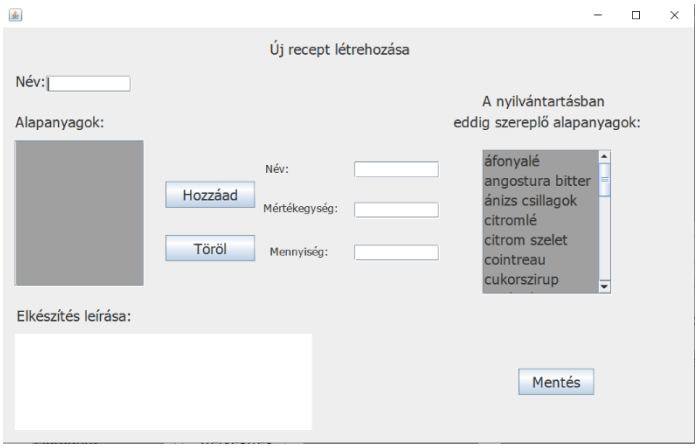
Program használata

Indítás után a felhasználó egyből a főmenüvel találkozik, melyben egérrel tud navigálni.



A program első indításakor rendelkezik egy alap nyilvántartással, melyben szerepelnek előre megadott sablon alapanyagok és italreceptek. Ha a program elindul, akkor betölti a nyilvántartásban szereplő alapanyagokat, ezek az alapanyagok szereplenek az italreceptekben.
A felhasználónak lehetőség van a nyilvántartás felépíteni egy fájlból ezt a „Nyilvántartás betöltése fájlból” gombra kattintva teheti meg. Ekkor megjelenik egy fájl választó ablak, ahol kilehet választani a fájlt, ami alapján a nyilvántartást

szeretnénk létrehozni. Fontos azonban, hogy a program nagyon szigorú megkötéseket ad a fájl szerkezetéről. Első sorban a fájl formátuma „txt” kell hogy legyen. Egy sor felépítési pedig a következő: receptnek a neve majd egy „;” karakterrel elválasztva jönnek az alapanyagok. Egy alapanyag felépítése úgy néz ki, hogy alapanyag neve „-„ karakter ez után pedig jön az alapanyag mennyisége és egy ilyen karakter „-„ végül pedig szerepel az alapanyag mértékegysége. Minden alapanyagot ilyen „/” karakterrel kell elválasztani. Az utolsó alapanyagnál már nem kell használni a „/” karakter helyette viszont a „;” karaktert igen, majd legvégül jöhet a recept elkészítésének leírása itt már nincsen semmilyen megkötés.
Egy példa sor:
Gin Tonic;gin-4.0-cl/tonic-2.0-dl/citrom szelet-1.0-db/jég-3.0-db;A pohárba jégkockára öntjük a gint és a tonicot…
Amennyiben a felhasználó nem megfelelő fájlt választott ki erről értesíti a program. Ha sikerült a fájl beolvasása akkor az „Összes recept listázása” gombra kattintva már meg is jelennek a receptek.
A nyilvántartást természetesen el is lehet menteni, sőt nagyon ajánlott minden használat után mivel a következő indításkor csak így garantált a nyilvántartás adatvesztesség nélküli működése. A mentést a „Nyilvántartás mentése” gomra kattintva lehet, a sikeres mentésről visszajelzést kap a felhasználó.



A nyilvántartáshoz az „Új recept hozzáadása” gombra kattintva van lehetőség, ekkor a következő ablak jelenik meg.
A „Név” mezőnél a recept nevét lehet megadni itt bármilyen karakter megadható a felhasználóra van bízva, azonban fontos hogy az itt megadott név alapján kerül a recept a nyilvántartásba.
Az ablak közepén megjelenő három mező a recepthez hozzáadni kívánt alapanyag adatait tartalmazza(név, mértékegység, mennyiség) ide kézzel beírva lehetőség van megadni az adatokat majd a „Hozzáad” gombbal lehet hozzáadni a receptünkhöz. Ha ez sikeres volt akkor az alapanyagok listában meg is jelenik az általunk létrehozott alapanyag. Amennyiben mégse szeretnénk ezt az alapanyagot a receptünkhöz hozzáadni, akkor a „Töröl” gomb megnyomásával lehetőség van kitörölni a listából. Továbbá a jobb oldali listában szerepelnek a nyilvántartásban eddigi szereplő és általunk létrehozott alapanyagok, ez a munkák megkönnyítésre szolgál nem kell mindig kitölteni az összes adatott ehhez a listából ki kell választani egy elemet és a program automatikusan kitölti a név és mértékegység mezőt. Elég csak a mennyiséget megadni kézzel viszont a felhasználónak természetesen lehetősége van ezeket az adatokat is felül írni, így továbbra is módosíthatók az adatok. Ha bármilyen probléma felmerül az alapanyag hozzáadásakor, akkor azt a program jelzi a felhasználónak és az ott leírtak követendők. Az elkészítés leírásának megadásakor nincsen semmilyen megkötés bármilyen szöveget meglehet adni. Ha elkészültünk a recept adatinak megadásával, akkor a „Mentés” gombra kattintva lehet hozzáadni a nyilvántartáshoz a receptet a mentés sikerességéről értesítést ad

a program. Ha nem szeretnénk újabb receptet hozzáadni a nyilvántartáshoz, akkor be lehet zárni az ablakot.
A főmenüben található „Összes recept listázása” gomb kilistázza a nyilvántartásban szereplő összes receptet a „Receptek” nevű listába. Ebben a listában magyar ABC szerinti betűrendben jelennek meg a receptek. Egy receptet kiválasztva (melyet az egér bal gomb lenyomásával tudunk megtenni) automatikus megjelenik a kiválasztott recept adatai. Többek között látjuk a recept nevét és az elkészítéshez szükséges alapanyagok mennyiségét és mértékegységét. Illetve a recept leírását, hogy milyen módon lehet elkészíteni. Tovább megjelenik még két gomb mely a recept szerkesztésére és törlésére kínál lehetőséget. A „Szerkesztés” gombra kattintva egy nagyon hasonló ablak jelenik meg, mint az új recept hozzáadásakor. Viszont itt már mindent adat előre ki van töltve és ezeket az adatokat lehet szerkeszteni akárcsak egy új recept hozzáadásakor. Lehetőség van a recept nevének megadására, a receptben szereplő alapanyagok módosítására, akár a mennyiség megváltoztatására vagy ki is lehet törölni, illetve új alapanyagot is hozzálehet adni. A mentés gombra kattintva a program elmenti az eddig változtatásokat a receptről. A sikeres mentésről visszajelzést ad a program ezután már be lehet zárni az ablakot. Ha kiszeretnénk törölni egy receptet a nyilvántartásból, akkor ez a törlés gombra kattintva tehető meg a program megkérdezi, hogy biztos ki szeretnénk törölni a receptet az „igen” gombra kattintva a recept törlődik, ha a „nem” opciót választjuk akkor a törlés nem lép érvénybe. A receptek között lehetőség van név alapján is keresni, ehhez nem kell mást tenni, mint a „Keresés recept név alapján” mezőbe begépelni az általunk keresett nevet, a program automatikusan keres és frissíti a listát minden egyes karakter után. A nyilvántartásban szereplő receptek között alapanyagokként is lehet szűrni és keresni. Ehhez az alapanyagok listából ki kell választani az alapanyagot, ami alapján keresni szeretnénk majd a „Hozzáad” gomb lenyomásával megjelenik a „Szűrt alapanyagok” listában. Lehetőség van még további alapanyagokat hozzáadni a kereséshez vagy akár a „Szűrés” gombra kattintva elindul a keresés és a „Receptek” listában meg is jellenek azok a receptek, amik tartalmazzák a kiválasztott alapanyagot vagy alapanyagokat. A szűrésből egy alapanyagot a „Törlés” gomb segítségével lehet kitörölni. A főmenüben található még egy gomb „Új alapanyag” ez a gomb egy új ablakot nyit meg, ahol meglehet adni a nyilvántartáshoz hozzáadni kívánt alapanyag adatait (név, mértékegység) a „Mentés” kattintva az alapanyag hozzáadódik a nyilvántartáshoz. Ennek sikerességéről visszajelzést ad a program és ezután már be lehet zárni.

Fejlesztői dokumentáció

Feladat

A program feladata, hogy egy recepteket tartalmazó txt fájlból felépítsen egy italrecept nyilvántartást és lehetőséget biztosítson a felhasználónak új receptek hozzáadásához, receptek szerkesztéséhez, törléséhez. Továbbá a receptek között lehet keresni név szerint és alapanyag szerint. A nyilvántartást lehet menti vagy esetleg betölteni egy előre elkészített fájlból.

Fejlesztői környezet

A program windows 10-es operációs rendszeren készült és a Eclipse IDE for Java Developers 4.18-as verziójú fejlesztő környezetben. Fejlesztés során a Java 11.0.11-es SDK került használatba. A grafikus felület a Java Swing Windowbuilder bővítménnyel készült ez lehetőséget biztosított a könnyebb, átláthatóbb GUI létrehozásában.

Forrás fájlok

A programnak két fontos forrás fájlja van az egyik a „register.txt” ami kezdettben tartalmaz néhány előre megadott italreceptet. Ahhoz, hogy a program tökéletesen működjön nagyon szigorú szerkezeti felépítést kell betartani. Egy sor felépítése így néz ki: „a recept neve ”;” „alapanyag neve” „-„ „alapanyag pontos mennyisége (ez lehet egész vagy akár decimális szám is viszont figyelni kell, hogy tizedespontot kell használni.) „-„ „alapanyag mértékegysége” és minden alapanyagot „/” karakterrel kell elválasztani. Ez után jön egy „;” karakter és lehet írni a recept elkészítésének leírását. Íme egy példa sor:

Gin Tonic;gin-4.0-cl/tonic-2.0-dl/citrom szelet-1.0-db/jég-3.0-db;A pohárba jégkockára öntjük a gint és a tonicot és alaposan megkeverjük…..

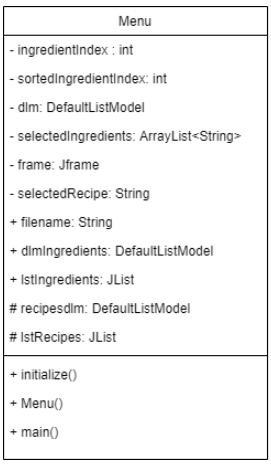
Fontos, hogy minden recept külön sorba kerüljön csak így garantált az, hogy a program tökéletesen felismeri és eltudja különíteni a recepteket. Ilyen szöveges fájlt akár kézzel is elkészíthet a felhasználó mivel erre a programban van lehetősége viszont, ha nem megfelelő a fájl szerkezete akkor erről értesíti a program. A másik fontos fájl az „ingredients.txt” a program minden betöltés elött felhasználja a „register.txt” fájlban szereplő recepteket és összegyűjti a bennük szereplő alapanyagokat. Az alapanyagokól minden tulajdonság tárolásra kerül. Viszont a felhasználó létre tud hozni szabadkézzel is alapanyagot, amely nem feltétlenül szerepel valamelyik receptben így a „register.txt” fájlba se kerül be az „ingredients.txt” fájl ezt a célt szolgálja, hogy ide kerülnek mentésre azok az alapanyagok, amelyeket a felhasználóhoz létre, de egyelőre nem használt fel egyetlen receptben sem.

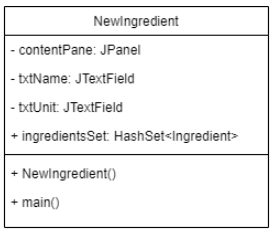
Projekt felépítés

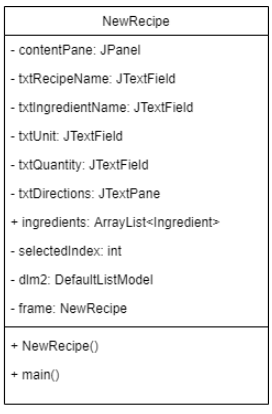
A program három nagyobb részre (csomagra) van bontva: appearance, exceptions, main.

Appearance csomag

Ebben a csomagban található osztályok, függvények a megjelenítést és a felhasználói interakciók kezelését szolgálja. Négy osztály található benne: EditRecipe, Menu, NewIngredient, NewRecipe.







Menu osztály

Ez az osztály a program főmenüjét jeleníti meg a felhasználó számára. Az osztálynak vannak private, public és protected adattagjai is.

Adattagok

private int ingredientIndex = Kezdetben nulla kezdőértékkel rendelkezik azonban változik az értéke, ha a felhasználó kiválaszt egy alapanyagot a listából és a listában elfoglalt pozíció indexére módosul.
private int sortedIngredientIndex = Kezdetben nulla kezdőértékkel rendelkezik és akkor változik az értéke, ha a felhasználó a szűrt alapanyagok listájában kiválaszt egy elemet és a kiválasztott elem lista indexét veszi fel.

private ArrayList<String> selectedIngredients= Kezdetben egy üres lista, amely stringeket tartalmazhat. Ebbe a listába kerülnek azok az alapanyagok, amelyek alapján a felhasználó szűrni szeretne az italreceptek között
private JFram frame = Ez a keret ahol a főmenü megjelenítése történik, ehhez adódnak hozzá a különféle panelek, amelyek gombokat, listákat és szövegeket tartalmaznak.
private String selectedRecipe = Kezdetben egy üres string és majd a felhasználó által kiválasztott recept nevét fogja tárolni
public String fileName = Létrehozáskor egy üres string, mely akkor kap értéket amikor a felhasználó kiválaszt egy betölteni kívánt fájlt. Azért public a láthatósága mivel, ha sikeres a fájl betöltés akkor ezt a fájl nevet továbbitani kell más osztálynak.
public static DefaultListModel dlmIngredients = Ez egy alap lista modell, ami az alapanyagok JListnek a felépítésére szolgál a public láthatóság azért kell, mert különböző osztályoknak el kell érni és módosítania kell.
public static JList lstIngredients = Létrehoz egy JListet, ami majd tartalmazza az összes alapanyagot.
protected static DefaultListModel recipesdlm = Egy alap lista modell, ami a recepteket tartalmazó lista felépítésre szolgál. A protected szint beállítása azt a célt szolgálja, hogy csak csomagon belül van felhasználva ez az adattag.
protected static JList lstRecipes = JList lista, ami az összes receptet tartalmazza.

Függvények

public void initialize() = A főmenüben található összes label, lista és gomb beállítását tartalmazza ezentúl figyeli és kezeli a felhasználó interakciókat. Többek között nézi ha elindult az alkalmazás, akkor az alapanyagokat előre betölti egy listába, mikor melyik alapanyag/recept lett kiválasztva. Rengeteg lokálist változója van, amik a megjelenítést szolgálják.
public static void main() = Létrehozza egy menü típusú változót és láthatóvá teszi a keretet. Ez az osztály építi fel az alapokat és indítja el az ablakot.

NewRecipe osztály

Ez az osztály jeleníti meg egy új recept létrehozásához szükséges ablakot, illetve kezeli az ablak használata során felmerülő felhasználói interakciókat.

Adattagok

private JPanel contentPane = Erre a panelra kerülnek a gombok, szövegek, listák és beviteli mezők.
private JTextField txtRecipeName = Beviteli mező, ahol a felhasználó megtudja adni a recept nevét. Bármilyen karakter elfogadható nincsen kikötés viszont nem lehet üresen hagyni kötelező kitölteni.
private JTextField txtIngredientName = Beviteli mező, ahol a felhasználó megtudja adni a recepthez hozzáadni kívánt alapanyag nevét. Bármilyen karakter elfogadható viszont nem lehet üres.
private JTextField txtUnit = Beviteli mező, ahová a felhasználó megadhatja a recepthez hozzáadni kívánt alapanyag mértékegységét. Ez sem lehet üres, de bármilyen karakter elfogadható.
private JTextField txtQuantity = Beviteli mező, ahol a felhasználó megtudja adni mekkora mennyiséget szeretne az alapanyagból hozzáadni a recepthez.
private JTextPane txtDirections = Egy nagyobb beviteli mező, ahol meglehet adni a recept elkészítésének leírását.
public static ArrayList<Ingredient> ingredients = Az újonnan létrehozott recept alapanyagait tartalmazza. Csak Ingredient osztály elemeket tartalmazhat.

Függvények

public NewRecipe = Az osztály konstruktora itt rengeteg lokális változó található, létrehozza az ablak megjelenítését és kezeli a felhasználó által bevitt adatokat.
public static void main = Meghíváskor beállítja, hogy látható legyen az ablak.

EditRecipe osztály

Egy már létrehozott recept szerkesztését valósítja meg. Felépítése nagyon hasonló a NewRecipe osztályhoz.

Adattagok

private JPanel contentPane = Fő panel elem, amely a gombok, listák, beviteli mezők kerülnek
private JTextField txtRecipeName = Beviteli mező, ahová előre betöltődik a recept eddigi neve és a felhasználó szerkeszteni tudja. Bármilyen karakter elfogadható nincsen kikötés viszont nem lehet üresen hagyni kötelező kitölteni.
private JList lstIngredients = Alapanyagokat tartalmazó lista, ahová automatikusan betöltődnek ez eddig megadott alapanyagok nevei. Ebből a listából törölni és szerkeszteni is tud a felhasználó.
private JTextField txtIngredientName = Beviteli mező, ahol a felhasználó megtudja adni a recepthez hozzáadni kívánt alapanyag nevét. Bármilyen karakter elfogadható viszont nem lehet üres. Amennyiben a felhasználó kiválaszt egy előre megadott alapanyagot akkor automatikusan kitöltésre kerül a mező, de ezt tudja módosítani a felhasználó.
private JTextField txtUnit = Beviteli mező, ahová a felhasználó megadhatja a recepthez hozzáadni kívánt alapanyag mértékegységét. Ez sem lehet üres, de bármilyen karakter elfogadható. Amennyiben a felhasználó kiválaszt egy előre megadott alapanyagot akkor automatikusan kitöltésre kerül a mező, de ezt tudja módosítani a felhasználó.
private JTextField txtQuantity = Beviteli mező, ahol a felhasználó megtudja adni mekkora

mennyiséget szeretne az alapanyagból hozzáadni a recepthez.
private JTextPane txtDirections = Egy nagyobb beviteli mező, ahol megjelenik a recept eddigi elkészítésének leírása. A felhasználónak lehetősége van a teljes szöveg szerkesztésére.
public static ArrayList<Ingredient> ingredients = Ingredient osztályokat tartalmazó lista. Itt találhatóak az eddigi alapanyagok, amelyek szerepelnek a receptben.
private int index = Kezdetben egy nulla kezdőértékkel rendelkező változó és annak függvényében változik, hogy a felhasználó melyik előre megadott alapanyagot választja ki. Ennek a lista indexét tartja nyilván.
private String selectedRecipe = Ez az adattag a szerkesztésre kiválasztott recept nevét tartalmazza.

Függvények

public EditRecipe(String recipe) = Az EditRecipe osztály konstruktora paramétérként megkapja a szerkesztésre kiválasztott recept nevét.
public static void main() = Létrehozza és beállítja az ablak láthatóságát.

NewIngredient osztály

Létrehoz egy új ablakot, ahol a felhasználónak lehetősége van létrehozni egy új alapanyagot.

Adattagok

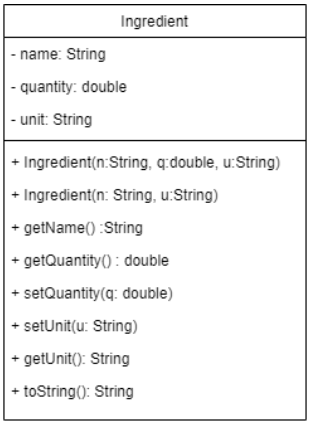
private JPanel contentPane = Fő panel, amely tartalmazza a gombokat, beviteli mezőket.
private JTextField txtName = Beviteli mező, ahová a felhasználónak lehetősége van megadni a nyilvántartáshoz hozzáadni kívánt alapanyag nevét. Bármilyen karakter elfogadható.
private JTextField txtUnit = Beviteli mező, ahol a felhasználó megtudja adni a létrehozott alapanyag mértékegységét.
public static HashSet<Ingredient> ingredientsSet = A program ide gyűjti külön az olyan alapanyagokat, amelyek egyelőre egyetlen receptnél sincsenek hozzáadva.

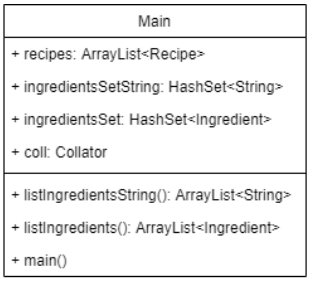
Függvények

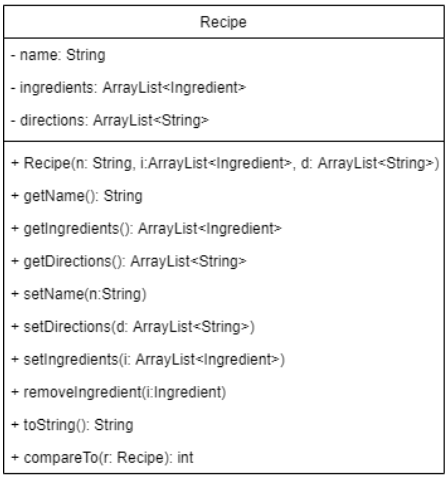
public NewIngredient() = Konstruktor létrehozza a gombokat, szövegeket és beállítja a tulajdonságait illetve kezeli a és elmenti a felhasználó által megadott adatokat.
public static void main() = Létrehozza az ablakot és beállítja, hogy meghíváskor látható legyen.

Main csomag

Ebben a csomagban a program fő osztályai, amiből felépül az egész nyilvántartás.







Recipe osztály

Adattagok

private String name = A recept nevét tárolja.
private ArrayList<Ingredient> ingredients = Egy Ingredientseket tartalmazó lista. Ebben a listában vannak tárolva a recepthez hozzáadott alapanyagok.
private ArrayList<String> directions = Egy Stringeket tartalmazó lista. Itt van eltárolva egy recept elkészítésének leírása.

Függvények

public Recipe (String,ArrayList<Ingredient>,ArrayList<String>) = A „Recipe” osztály konstruktora paraméterként megkell adni a recept nevét, egy „Ingredienteket” tartalmazó listát és egy Stringeket tartalmazó listát.
public String getName() = Visszatérési értékként visszaadja a recept nevét.
public ArrayList<Ingredient> getIngredients() = Visszatérési értékként megadja egy recepthez tartozó alapanyagok listáját.
public ArrayList<String> getDirections() = Visszatérési értékként visszaadja a recepthez tartozó elkészítési leírás listát.
public void setName (String name) = Beállítja a name adattagot a paraméterként kapott értékre.
public void setDirections (ArrayList<String>) = Beállítja a directions adattagot a paraméterként kapott értékre.
public void setIngredients (ArrayList<Ingredient>) = Beállítja az ingredients adattagot a kapott paraméterre.
public void removeIngredient (Ingredient) = Kiveszi a paraméterben kapott „Ingredient”

elemet az „ingredients” listából.
public String toString() = Visszatérési értékként megadja azt a formátumot, ahogy az osztály kiszeretnénk írni.
public int compareTo (Recipe) = Összehasonlít két receptet a neveik alapján és megmondja melyik van előrébb a Magyar ABC-ben.

Ingredient osztály

Egy alapanyagról tárol tulajdonságokat.

Adattagok

private String name = Az alapanyag nevét tárolja.
private double quantity = Az alapanyag mennyiségét tárolja.
private String unit = Egy alapanyag mértékegységét tárolja.

Függvények

public Ingredient (String,double,String) = Az „ingredient” osztály konstruktora paraméterként megkell adni az alapanyag nevét, mennyiségét és mértékegységét.
public Ingredient (String,String) = Az „ingredient” osztály másik konstruktora itt azonban elég csak két paraméter adni, a recept nevét és mértékegységét. Akkor van használatban, amikor a felhasználó csak egy alapanyagot hoz létre, de nem tudja még a mennyiségét.
public String getName() = Visszaadja a name adattag értékét.
public double getQuantity() = Visszaadja a quantity adattag értékét.
public void setQuantity (double) = A quantity adattag értékét beállítja a paraméterként kapott értékre.
public void setUnit (String) = A „unit” adattag értékét beállítja a paraméterben kapott értékre.
public void getUnit() = Visszaadja a „unit” adattag értékét.
public String toString() = Visszatérési értékként visszaadja, hogy az osztály adattagjait milyen formátumban szeretnénk kiíratni.

Search osztály

Ebben az osztályban találhatóak a keresési műveletek, ha bármilyen keresés van a programban akkor az innen kerül meghívásra.

Függvények

public static int searchByName (ArrayList<Recipe>,String) = Végig megy a paraméterként kapott listán és összehasonlítja a Recipe.getName() függvény által visszaadott értéket a paraméterként kapott Stringgel, ha talál egyezést akkor visszaadja a listában elfoglalt indexét.
public static String searchByIngredientName (ArrayList<Ingredient>,String) = Végig megy a paraméterben kapott listán és összehasonlítja az „Ingredient.getName() függvény által visszaadott értéket a paraméterben kapott stringgel amennyiben van egyezés, akkor visszaadja az alapanyag nevét.
public static ArrayList<Recipe> searchByText (ArrayList<Recipe>,String) = Végig megy a paraméterként kapott listán majd a receptek nevét átkonvertálja kisbetűre és ha a recept neve tartalmazza a paraméterként kapott Stringet, akkor hozzáadja egy listához a receptet.

public static ArrayList<Recipe> searchByFilteredIngredients (ArrayList<Recipe>, ArrayList<String>) = Végig megy az összes recepten majd minden recept alapanyagán és ha benne van paraméterben megadott összes szűrt alapanyag, akkor hozzáadja egy listához majd ez a listát fogja visszaadni.

FileManagement osztály

Ez az osztály felelős a fileokkal való műveletekért írja és olvassa az „register.txt” és az „ingredients.txt” fájlt.

Függvények

public static void readFile = Beolvassa azt a fájlt, amit paraméterben kap. Mivel a beolvasott fájl megfelel a követelményeknek, így hiba nélkül végig tud menni a fájl sorain. A kapott sort feldolgozza megfelelően szétbontja és külön tömbökben tárolja majd ebből készít egy „Recipe” osztályt, amit hozzáad a fő ArrayListhez.
public static void writeFile = Végig megy a paraméterként kapott listán feldolgozza a kapott adatokat előállít egy sort és kiírja egy txt fájlba ezt egészen addig csinálja, amíg van elem a listában.
public static void readIngredientsFile = Végig megy a „ingredients.txt” fájl sorain feldolgozza a kapott adatokat majd hozzáadja a megfelelő listához. Ez a fájl az, ami azokat az alapanyagokat tartalmazza, amik nem tartoznak egyetlen recepthez sem.
public static void writeIngredientsFile = Végig megy azokon az alapanyagokon, amik egyik receptben sem szerepelnek és kiírja őket egy fájlba.

Main osztály

Ez az osztály a program fő osztálya itt vannak a program fő elemei például az összes receptet tartalmazó lista és maga a program is ezzel az osztállyal indítható.

Adattagok

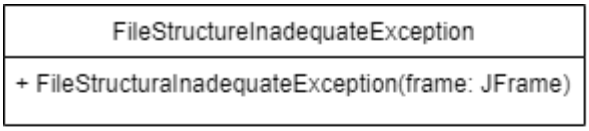
public static ArrayList<Recipe> recipes = Egy „Recipe” osztályokat tartalmazó lista ebben van tárolva a program összes receptje.
public static HashSet<String> ingredientsSetString = Ebben a halmazban van eltárolva a nyilvántartásban szereplő összes alapanyag neve.
public static HashSet<Ingredient> ingredientsSet = Egy „Ingredient” osztályokat tartalmazó halmaz, ami tartalmazza az összes alapanyagot, ami a nyilvántartásban szerepel.
public static Collator coll = Egy collator, ami segít a Magyar ABC szerint betűrendbe rakni a recepteket, alapanyagokat.

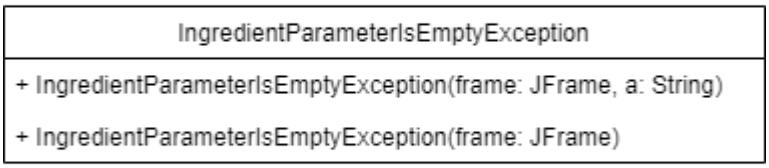
Függvények

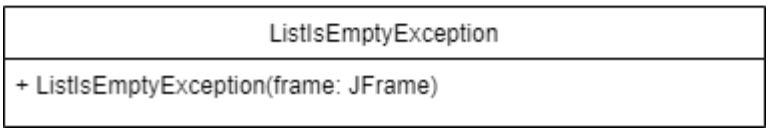
public static ArrayList<String> listIngredientsString() = Végig megy az összes alapanyagon, ami a nyilvántartásban szerepel. ABC szerint rendezi őket majd hozzáadja az ingredientsSetStringhez.
public static ArrayList<Ingredient> listIngredients() = Végig megy az összes alapanyagon, ami szerepel a nyilvántartásban és hozzáadja az „ingredientsSet” halmazhoz.
public static void main() = Ezzel a függvénnyel kell elindítani a programot. Meghívja a fájl beolvasásokat majd a program megjelenítését.

Exceptions csomag

Ez a csomag tartalmazza a programban található kivételkezeléseket.







FileStructureInadequateException osztály

Akkor kerül meghívásra, ha a betölteni kívánt fájl szerkezete nem megfelelő ezt grafikusan meg is jeleníti a felhasználónak.

IngredientParameterIsEmptyException osztály

Akkor kerül meghívásra, ha a felhasználó egy alapanyag megadásakor vagy létrehozásakor üresen hagyja az adatok mezőit.

ListIsEmptyException

Akkor kerül meghívásra, ha a felhasználó olyan listából szeretne törölni, ami üres és nem tartalmaz semmilyen adatot. Erről grafikus megjelenítéssel értesíti a felhasználót.