**Záródolgozat**

**Szoftverfejlesztő komplex szakmai vizsga**

**54-213-05**

**Vendéglátóipari elszámoltató rendszer**

**Bálint Dániel**

Csány László Szakgimnázium

Zalaegerszeg

1. **A dolgozat célja………………………………………………1**
2. Tartalmi bevezetés......................................................................1
3. Miért ez?......................................................................................1
4. Szoftver erőforrások…………………………………………………1
5. Észrevételek, tapasztalatok………………………………………...1

**II. Felhasználói dokumentáció.....................................2**

1. A program leírása, funkciók..........................................................2

**III. Fejlesztői dokumentáció.....................................................9**

1. Az adatbázis................................................................................9

2. Osztályok leírása.......................................................................11

3. Saját osztálykönyvtárak.............................................................15

4. UML diagram.............................................................................16

**IV. Tesztkörnyezet..................................................................17**

**V. Összefoglalás.....................................................................18**

**I. A dolgozat célja**

**I.1. Tartalmi bevezetés**

A BlueMaze nevezetű vendéglátóipari elszámoltató rendszer alapvetően a pincérek munkájának segítésére szolgál. Ennek a szoftvernek a segítségével a végletekig leegyszerűsödik a **számlázás** és az eddig oly sok nehézséget okozó kalkuláció is.

A rendelés felvétele után az alkalmazottaknak csak ki kell választaniuk melyik asztal rendelését szeretnék felvinni, egy menüből kiválaszthatók az áruk, melyek ráíródnak az adott vendék számlájára. A program kezeli az asztalokat, pultnál kért rendeléseket és törzsvendégek számára hosszabb távú számla is létrehozható.

Lehetőség van új asztalok, törzsvendégek, áruk felvételére. Az áruk 3 fő kategóriára vannak osztva és ezen belül adhatók nekik besorolások is, hogy könnyen megtalálhatók legyenek. Ezeknek a módosítására vagy törlésére is lehetőség van, amennyiben egy asztal megszűnik, egy áru lekerül az étlapról vagy a törzsvendég nem tart igényt tovább a számlájára.

A külföldi vendégek számára a számlán fel van tüntetve a végösszeg euróba átszámítva is.

**I.2. Miért ez?**

A témaválasztás indoka az, hogy jómagam is vendéglátós családból származom és pontos rálátásom van a szakma nehézségeire és átlátom ezen problémák hogyan oldhatók meg. A mi éttermünkben is használatban van egy hasonló célra készült program, ám ennek havi díja van. Későbbi fejlesztésekkel a BlueMaze pontosabban a saját igényeinkre szabható és felválthatja a jelenlegi szoftvert, természetesen plusz kiadások nélkül.

**I.3. Szoftver erőforrások**

A program elkészítése Netbeans IDE fejlesztői környezet segítségével, java nyelven történt. Azért esett a javára a választás, mert így bármilyen operációs rendszeren futtatható egyéb segédprogramok nélkül. Napjainkban ez a nyelv rendkívül elterjedt és többnyire minden eszköz rendelkezik a futtatókörnyezettel. Közrejátszott az is, hogy a java a programozási „anyanyelvem”, tehát ebben van a legtöbb tapasztalatom. Az adatok tárolására egy SQLITE adatbázist választottam, mivel ennek kicsi a gépigénye, nem kell hozzá külön szervert futtatni és könnyen kezelhető. Az adatbázis szerkesztéséhez az sqlite-jdbc.jar fájl 3.21.0-ás verziója volt a segítségemre. Használatba került még egy saját általam írt flbr.jar fájl is, ami az évek során összegyűlt általános metódusokat tartalmazza, így ezeket nem kell minden alkalommal megírnom.

**I.4. Észrevételek, tapasztalatok**

Fejlesztés során rengeteg tapasztalatra tettem szert mindazon által, hogy eddig még nem látott hibák tucatjaival találtam szembe magam. Személyes véleményem szerint egy ilyen dolgozat megírása megtanít minket arra, hogy hogyan találjuk fel magunkat és hogy hol keressük a megoldást. Számomra a legnehezebb a felhasználói felület megírása volt, hiszen egy könnyen átlátható, kezelhető felületet akartam létrehozni. Figyelembe vettem azt is, hogy a színvilág egyértelműen jelezze a felhasználó számára, hogy éppen mit is csinál. Továbbá a gombok megtervezésénél az is szempont volt, hogy megfelelő méretűek legyenek egy érintőképernyőn történő használathoz.

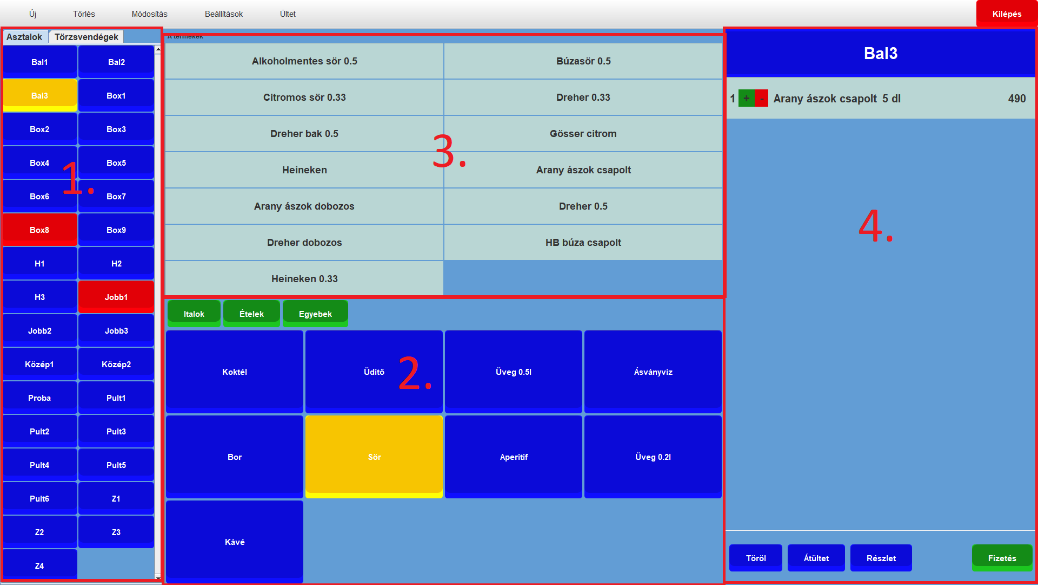
Az eddigi adatbáziskezelési tudásom nagyban növelte az, hogy rengeteg kommunikációt kellett végrehajtanom az adatbázissal és ezek többsége nem hasonlítható egy típusfeladathoz sem, amiken keresztül ezt megtanultam.

1. **Felhasználói dokumentáció**

**II.1. A program leírása, funkciók**

A program nem véletlenül kapta a BlueMaze nevet. Első ránézésre valóban csak egy kék útvesztőnek tűnhet a rengeteg gomb miatt, mindazon által nagyon könnyen kiismerhető.

Alapvetően 4 fő részre osztható a felület.



*(1. ábra)*

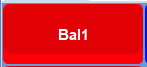
Ahogy az első ábrán az 1-es számmal megjelölt területen is látszik, az asztalok helyezkednek el 2 oszlopban. Ez a felület függőlegesen görgethető, ha esetleg nem férnek ki asztalok. Ezen gombok felett találunk két fület, amik lehetővé teszik az asztalok és törzsvendégek számlái között történő váltást.

A gombokra kattintva sárga színnel jelenik meg a kijelöl elem.



*(2. ábra)*

A kijelölt asztalra tudunk rendeléseket felvinni. Azok a gombok, amik piros színnel jelennek meg azokon már van rendelés, tehát annál az asztalnál már ülnek vendégek.



*(3. ábra)*

A kék színnel megjelenők a szabad asztalok.



*(4. ábra)*

Törzsvendégek esetében a piros szín nem jelenik meg, hiszen egy számla azért jött létre, hogy tároljon valamilyen rendelést.

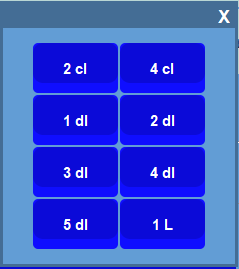
A 2-es számmal jelölt részen azt a menüszerkezetet találjuk, ami segít egy áru könnyű megtalálásában. A zöld 3 fő gombbal válaszhatunk az étel, ital és egyéb kategóriák között. Ezen belül a felhasználó tetszőlegesen állíthatja be, hogy milyen besorolásokat szeretne létrehozni. A kijelölt besorolás itt is narancssárga színnel jelenik meg. Nézzünk meg egy egyszerű példát:



*(5. ábra)*

Az ételek menüponton belül választottuk ki a frissensülteket. Ezzel a módszerrel nem kell hosszú listákat végig lapoznunk, vagy egy keresőbe beírni a nevét, hogy megtaláljuk, amit keresünk.

A besorolásokra kattintva el is érkezünk a 3. területhez, ahol a konkrét termékek jelennek meg. Értelemszerűen csak azok, amik abba a kategóriába tartoznak, amit kiválasztottunk. Egy árura kattintva láthatjuk ahogy fel is íródott az adott asztal számlájára (bemutatás későbbiekben). A termékeknél a program rákérdezhet a rendelni kívánt mennyiségekre. Például, ha csapolt sört szeretnénk rendelni, egy felugró ablakban ki kell választanunk, hogy mennyit is szeretnénk. Röviditalokra való tekintettel 2cl-től indul és az egyszerre legnagyobb rendelhető egység az 1 liter.



*(6. ábra)*

Ételeknél a mértékegység az egyszerűség kedvéért darabszámban van megadva.

A 4-es számú egység feladata a rendelések listázása, kisebb módosítások végrehajtása és a fizetés végrehajtása, számlanyomtatás. A felvett árukat abc sorrendben láthatjuk. Egy sorban egyetlen áru adatai vannak feltüntetve, ezek a mennyiség, megnevezés, egyes esetekben a mértékegység, valamint az ár. Ha olyan árut választunk ki, ami már van a számlán, akkor a már felvettnek a mennyisége fog növekedni. Amennyiben nem akarunk a menükből újra kiválasztani egy árut, lehetőségünk van a plusz és mínusz gombok használatára.



*(7. ábra)*

Ezek a gombok lehetővé teszik az egyszerű növelését valamint csökkentését az adott áruk mennyiségének. A mínusz gomb egy áru számláról való törlését is lehetővé teszi, amennyiben a mennyiség eléri a nullát.

Annak érdekében, hogy figyelemfelkeltő legyen melyik asztal rendelését is látjuk éppen, a lista felett található egy kék mező, amiben az adott asztal neve jelenik meg. Azt hogy melyik asztal rendelései jelennek meg, láthatjuk a már említett kijelölt gombokból is a bal oldalon.

Ezen a komponensen alul még találunk 4 gombot. A törlés gomb az aktuális asztalon lévő minden rendelést letöröl. Az átültet gomb akkor válik hasznossá mikor a vendég úgy dönt, hogy átül egy másik asztalhoz és a rendelésit is át szeretnénk vinni az új asztalra. Erre a gombra kattintva felugrik egy ablak, amin kiválaszthatjuk hová is ült át az illető. Ebben az ablakban megjelennek azok az asztalok is ahol már ülnek, arra az esetre, ha két vendég úgy dönt, egy asztalhoz ülnek. Ilyenkor a rendeléseik is egy számlára kerülnek.

Ugyanebben az ablakban a lista legalján kezdődnek a törzsvendégek nevei is. Ez azért van, mert így nem csak a törzsvendég számlájára lehet írni, hanem egy komplett asztal rendeléseit is át lehet rá vinni.

A fizetés gombra kattintva az asztalról lekerülnek a rendelések és egy virtuális számla jelenik meg a képernyőn. Ez csak egy összefoglaló előnézet a számláról, hivatalos számlaként nem nyomtatható, mivel nem tartalmazza a cég hivatalos adatait, valamit az áfa sincs feltüntetve rajta. A végleges számla nyomtatásának lehetősége a későbbiekben lesz kifejlesztve.

 A program alapvető kezelését és beállításait a képernyő tetején elhelyezkedő menü sávval lehet elvégezni.

*(8. ábra)*

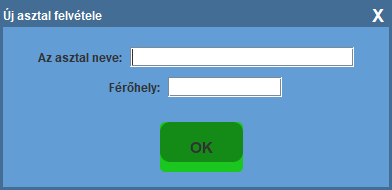
Az első menüpont az Új gomb, ami Áru, Asztal, Besorolás, Törzsvendég almenükre bomlik. Az Áru gomb segítségével lehet felvenni új árukat az adatbázisba. Az adatok bekérésére a 9. ábrán látható párbeszédablak szolgál.



*(9. ábra)*

A megnevezéshez az áru nevét írjuk, ami későbbiekben majd a számlán is megjelenik. Árhoz az adott áru árát adjuk meg. Ha azt szeretnénk, hogy a program kérdezzen rá a mértékegységre, valamint hogy mennyit szeretnénk felvenni, akkor az Egységes menüpontban válasszuk az igen lehetőséget. Ezek után az árat már érdemes liter árban megadni. Besorolást ki lehet választani a már meglévők közül, vagy az Új gombbal hozzáadni egy újat.

Új asztalt a 10. ábrán látható ablakkal tudunk felvenni.

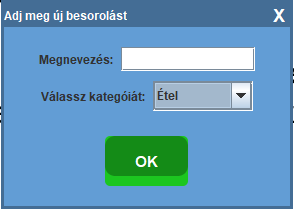


*(10. ábra)*

Itt csak az asztal nevét kell megadni, valamint hogy hány ember tud leülni oda.

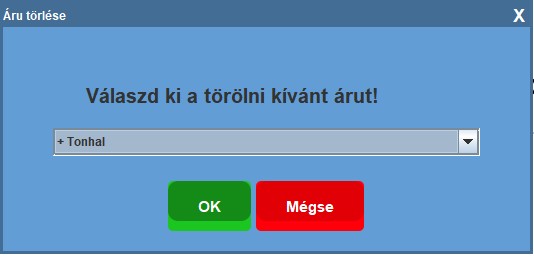
A férőhelyet azért kell megadni, mert van egy olyan lehetősége a programnak, hogy újonnan jött vendégeknek felajánlja, hogy melyik asztalokhoz tudnak leülni. Két azonos nevű asztal nem hozható létre!

Besorolás hozzáadásához egy megnevezést kell megadnunk és hogy milyen kategóriába szeretnénk elhelyezni. Ez lehet étel, ital vagy egyéb. Lásd 11. ábrán.



*(11. ábra)*

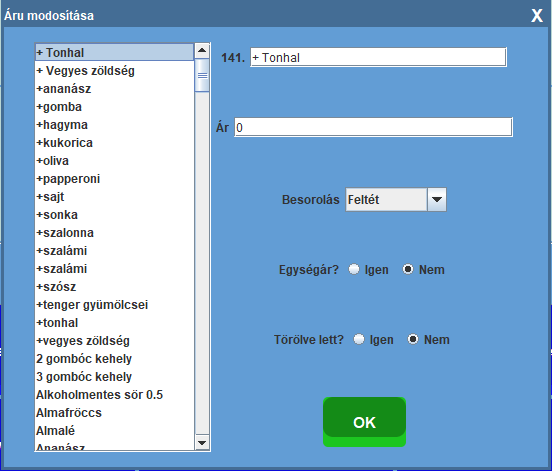
A második menü ezen a szalagon a Törlés. Ennek a gombnak a segítségével a fentiekben említett dolgokat tudjuk törölni az adatbázisból, ha már nincs rá szükség a továbbiakban. A 12. ábra egy áru törlését mutatja be.



*(12. ábra)*

Ez után következik a Módosítás menüpont. Itt tudjuk az esetleg elírt, elrontott vagy megváltozott árukat, asztalokat stb. módosítani.

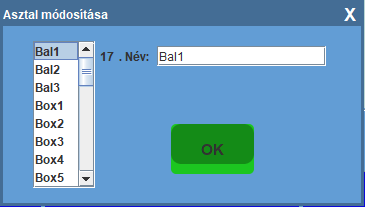
Elsőként egy áru módosítása:

**

*(13. ábra)*

Ebben az ablakban bal oldalt az egész adatbázisban szereplő összes áru megjelenik. A görgő segítségével válasszuk ki azt, amelyiket módosítani akarjuk. Jobb oldalon megjelenik az adott áru sorszáma (ez nem módosítható) és a neve. Ezt tetszőlegesen átírhatjuk. Alatta a termék ára módosítható. Itt figyeljünk oda, hogy egységárról van-e szó. A besorolását egy lehulló menüből választhatjuk ki. Érdekesség, hogy a törölt áruk nem törlődnek véglegesen és itt vissza is tudjuk állítani. Ennek azért van jelentősége, hogy a régen elmentett számlákon még megjelenjen. Természetesen a törölt státuszú áru nem vihető fel új számlára.

Egy asztal módosítása is hasonlóan zajlik. Egy görgethető sávból válasszuk ki az asztalt, majd tetszőlegesen átírhatjuk a nevét.

**

*(14. ábra)*

A besorolások és a törzsvendégek is ugyanilyen ablakban módosíthatók.

A beállítások menüpontban a program különböző részleteit lehet beállítani.

Itt kapott helyet az euró árfolyamának beállítása is. Azt kell megadni, hogy 1 euró hány forintot ér. A program indítása a windows-zal lehetőség még fejlesztés alatt áll, tehát csak a későbbiekben lesz üzemképes. Amennyiben erre valaki igényt tart, van lehetőség a gép azonnali leállítására a program bezárása után. Ez a két funkció együttes bekapcsolásával elérhető, hogy a gépen ne használhassanak más programot, amíg ez fut. Az „SQLITE lekérdezés” feliratú gomb egy rendszergazdai lehetőség. Itt biztonsági okokból csak lekérdezéseket lehet végrehajtani az adatbázisból. A tallózás gombbal kiválaszthatjuk azt a képet, ami a cég logója. Ez fog megjelenni a számlák tetején.

Az utolsó menüpont az ültetés. Ennek az a funkciója, hogy ha jön egy vendég, vagy többen, akkor itt csak meg kell adnunk, hogy hányan vannak és a program felajánlja a szabad asztalokat, ahol elférnek. Egy tetszőleges asztalra kattintva a főprogramban ki is jelölődik a választott asztal és már vihetjük is fel a rendeléseket.

A képernyő jobb felső sarkában kapott helyet a kilépés gomb. Ezzel lehet bezárni az alkalmazást. Futás közben automatikusan ment mindent a program, de mindezek ellenére egy biztonsági kérdést kapunk, hogy biztosan ki akarunk-e lépni.

**III. Fejlesztői dokumentáció**

**III.1. Az adatbázis**

Táblák:

**Aruk**:

ID *A tábla rekordjainak azonosítója (szám), ez a kulcs*

Megnevezés *Az áruk neve (szöveg)*

Ár  *Az áruk ára (szám)*

BesorolasID *Besorolas tábla azonosítója (szám)*

Egyseges *Azt jelzi, hogy ár egységár-e (logikai)*

Torolve *Azt jelzi, hogy az áru törölve lett-e (logikai)*

**Asztalok**:

ID *A tábla rekordjainak azonosítója (szám), ez a kulcs*

Nev Az asztal neve (szöveg)

Ferohely A férőhely, amennyien le tudnak ülni (szám)

Torolve Azt jelzi, hogy az áru törölve lett-e (logikai)

**Beallitasok:**

Arfolyam *Az euró árfolyama (szám)*

Inditas *Indítsa-e a programot a windowsal (logikai)*

Leallitas *Leállítsa-e a windowst a programmal (logikai)*

Logo *A logó elérési útja (szöveg)*

**Besorolasok:**

ID *A tábla rekordjainak azonosítója (szám), ez a kulcs*

Megnevezes *A besorolás neve (szöveg)*

TipusID *Tipusok tábla azonosítója (szám)*

Torolve *Azt jelzi, hogy a besorolás törölve lett-e (logikai)*

**Rendeles:**

AruID *Az Aru tábla azonosítója (szám)*

AsztalID *Az Asztalok tábla azonosítója (szám)*

TorzsID *A Torzsvendegek tábla azonosítója (szám)*

Mennyiseg *A Rendelés mennyisége (szám)*

Fizetendo *Amennyibe az adott áru kerül (szám)*

Mertekegyseg *Amennyit az áruból rendeltek (szöveg)*

**Szamlak:**

Szamlaszam *A tábla rekordjainak azonosítója (szám), ez a kulcs*

Datum A fizetés dátuma. (szöveg)

ReszletID A Szamlareszlet tábla azonosítója (szám)

**Szamlareszlet:**

ID *A tábla rekordjainak azonosítója (szám), ez a kulcs*

Vendeg *Asztal vagy törzsvendég neve kerül ide. (szöveg)*

Mennyiseg *A Rendelés mennyisége (szám)*

AruMegnevezes *Az áruk neve (szöveg)*

Fizetenndo *Amennyibe az adott áru kerül (szám)*

Mertekegyseg *Amennyit az áruból rendeltek (szöveg)*

**Tipusok:**

ID *A tábla rekordjainak azonosítója (szám), ez a kulcs*

Megnevezes A típusok megnevezése (szöveg)

**Torzsvendegek:**

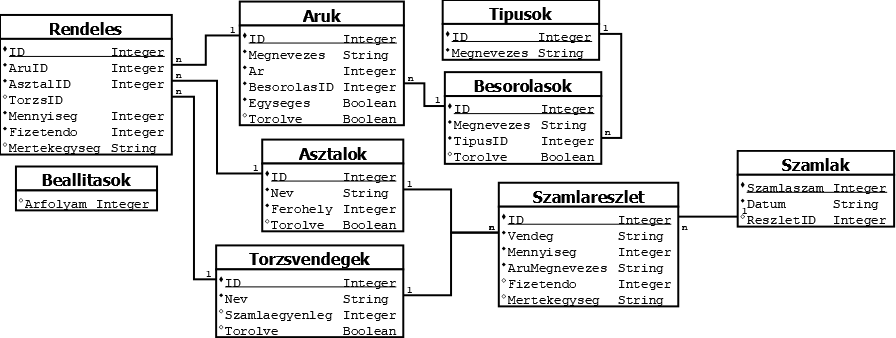
ID *A tábla rekordjainak azonosítója (szám), ez a kulcs*

Nev A törzsvendég neve (szöveg)

Szamlaegyenleg A felhalmozott tartozások (szám)

Torolve A törzsvendég törölve lett-e (logikai)

Az adatbázisban, a táblák közötti kapcsolatokat a 15. ábra mutatja be.

*(15. ábra)*

**III.2. Osztályok leírása**

A „szakdoga” csomag tartalma:

**Szakdoga.java**

Ez az osztály a program belépési pontja, itt található a main metódus. Az egyik fő feladata az adatbázissal történő kapcsolat létrehozása. Ennek a kapcsolatnak a megszakítása is itt történik egy úgynevezett leállítási kampóban, ami közvetlen a fő ablak bezáródása után fut le. A program legelső indításakor átmásolja a számára szükséges fájlokat egy helyi, számára írható és olvasható helyre. A későbbiekben ezekkel fog dolgozni. A grafikus felület betöltése előtt még a „betolt” metódus beolvassa a beállításokat az adatbázisból.

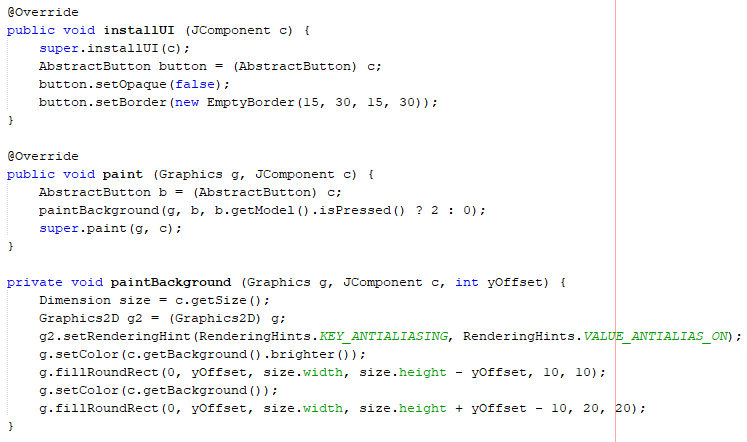
**Ablak.java**

Ez az osztály tartja a felhasználói felület főbb komponenseit és a menüsávot.

Az elrendezésért és az ablak felépítéséért az „epit” metódus felel. A JFrame osztályból öröklődik. A gombok kezeléséhez és a párbeszédablakok megnyitásához beépített osztályokat használtam. A menük kezelése MouseAdapter osztály származtatásával lett megoldva. Az almenük működését az ActionListener osztály implementálása végzi.

**Gomb.java**

A JButton osztályból öröklődik. Csupán egy beépített osztályt tartalmaz, ami örökli a BasicButtonUI-t. Mindezekre azért van szükség, mert ez csak egy keret lesz a további gombok számára. A következő felülírt metódusok végzik a grafikai változtatásokat a gombon:



*(16. ábra)*

**Keret.java**

JDialog osztályra épül, szintén csak dizájn célja van. Az összes párbeszédablak ebből öröklődik. Csupán egy „epit” metódust tartalmaz és a szükséges grafikus komponenseket. Az ablak undecorated tulajdonsággal rendelkezik, tehát a fejlécben saját tálca, méretező és bezáró gombok lettek definiálva. A fejléchez hozzá lett adva két MouseAdapter is, amik az ablak mozgathatóságát végzik. Az ablak fő komponense a pKozep nevű panel, ami BorderLayout elrendezéssel rendelkezik. Örököltetés után célszerű a felépített panelünket a pKozep panel CENTER részére helyezni.

**PanelCenter.java**

A nevéből is adódóan, alapvetően ez egy panel. Két részre van osztva, egy fenti és egy lenti részre. A fenti panelen található az a terület ahol a kiválasztott besorolású áruk jelennek meg. Az árukat tartalmazó panel GridLayout elrendezést kapott, és át lett adva egy ScrollPane-nek, hogy görgethető legyen. Az áruk 2 oszlopban jelennek meg, a teret a lehető legpontosabban kitöltve. A lenti panel tartalma egy CardLayout.

Ez három réteget tart magában, amik a: pItal, pEtel és a pEgyeb. Az imént említett panelekre a gombokat a „feltolt” meódus helyezi el egy sqlite lekérdezés után. Mindhárom szintén GridLayout elrendezést kapott, csak itt négy oszlopban helyezkednek el a gombok. Az osztály egyik legfontosabb metódusa a publikus „frissit”. Ennek a szerepe az, hogy más osztályok is elérjék, és ha változás történik, ez az összes felolvasott gombot letörli a panelekről és újra felolvassa az adatbázisból.

**PanelWest.java**

Ez a panel a képernyő bal oldalán helyezkedik el. Feladata az asztalok és a törzsvendégek gombjainak tartása. Ez a két rész között egy TabbedPane-nel lehet váltogatni. A gombok megnyomásával az adott asztal vagy törzsvendég jelölődik ki. Módosítások lehetősége miatt, ez az osztály is kapott egy „frissit” metódust.

**PanelEast.java**

Itt jelenik meg az eredménye annak, amit az előző két panelen kiválasztottunk. Ebben két frissítő metódus is van, mivel tudnia kell, hogy a törzsvendégek vagy asztalok tábláiból kell kiolvasnia az adatokat. Az egyszerűség kedvéért az itt megjelenő áruk egy beépített osztályként kaptak helyet. Ezeken található két rendkívül hasznos gomb, a plusz és a mínusz. Ezek feladata az adott áru mennyiségének növelése vagy csökkentése. Ezt úgy érik el, hogy például a plusz gomb egy ugyan olyan áruk visz fel az adatbázisba, 1 mennyiséggel. A panelt lefrissítve, SQL szinten ezek csoportosítva lettek, így azt látjuk, hogy kattintáskor növelődik az áru mennyisége és persze a végösszeg is. A csökkentés is hasonló elven működik, csak itt kitörlünk egy sort az adott termékből. Amennyiben a mennyiség 1 és csökkentjük, a frissítés után már nem fog megjelenni a listán.

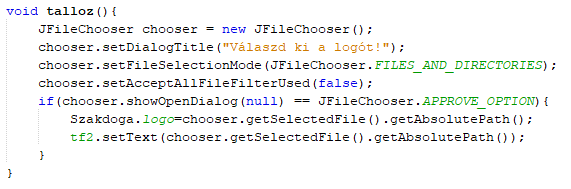
A „Parbeszed” csomag tartalma:

**Atultet.java**

Itt lép először használatba a Keret osztály, hiszen abból öröklődik. Az ablak felépítése rendkívül egyszerű, csak az asztalok és a törzsvendégek gombjai jelennek meg. Egyetlen említésre méltó egysége az „ultet” metódus. Ennek a feladata a tényleges áthelyezése a rendeléseknek a kiválasztott asztalra vagy számlára. Azok fognak áthelyeződni, amik a PanelEast területén láthatóak. Az áthelyezés nehézsége hogy mielőtt a rendeléshez tartozó asztal vagy törzsvendég ID-t átírná az adatbázban, ki kell deríteni, hogy a kiválasztott gomb asztal vagy számla-e, valamaint hogy amit át akarunk helyezni áruk, jelenleg asztalon vagy számlán vannak-e. Ezt úgy végzi el hogy a törzsvendégek táblájában megpróbálja megkeresni azt a nevet amit a megnyomott gombon van, ha megtalálja akkor egy számláról van szó, ha pedig nem, akkor csak asztal lehet. Az átírás után meghívódnak a frissítő metódusok.

**Beallitasok.java**

Ennek a JDialognak a megnyitásakor az adatbázisból felolvasódnak a beállítások, és a felületen ezek pontosan megjelennek. Az egész programban itt található az egyetlen olyan részlet, ami nem kapott külön dizájnt. A cég logójának a kiválasztásához használt FileChooser-ről van szó. Rendkívül egyszerű metódus, ezért is szerettem volna megtartani az alap kinézetét, hogy minden platformon az adott környezet stílusát vegye át.



*(17. ábra)*

**Beker.java**

Párbeszédablak szerepet tölt be. Az indító metódusban az ablak megjelenítése van, majd miután bezáródik, visszatér egy String értékkel. A metódus publikus, hogy példányosításkor is meg lehessen hívni.

**public** String **run**(){

**this**.setVisible(**true**);

**return** vissza;

}

A felületen csupán egy felirat található, annak közlésére, hogy milyen adatot szeretnénk bekérni, valamint egy TextArea a bevitt adatnak. Hibakezelést nem végez, így ezt a bekért adattal a későbbiekben kell megtenni.

**Fizet.java**

Alapvetően egy BorderLayout elrendezésű ablak, de csak a NORTH része van használatban. Kombinálva egy függőleges irányú BoxLayout-al, azt a megjelenést éri el, hogy a számlára helyezett áruk a képernyő tetejétől lefele folynak, függetlenül a mennyiségtől, hiszen a számla bármilyen hosszú lehet. A korlátlan hosszúságot az ablakban egy görgethető panel imitálja.

A fizetés végrehajtásával az adatbázisban a „Rendelesek” tábla tartalma átíródik a véglegesen tárolt számlák közé, ahol már nem szerkeszthető. Az adott asztal tartalma eltűnik, készen a következő vendégre.

**Kerdez.java**

Igen és nem lehetőségeket felajánló ablak. Kérdések eldöntésére alkalmas. Amennyiben igen a válasz, az ablak visszatérési értéke 1, ha nem a válasz, azon esetben 2. A visszatérő változó alap értéke -1, aminek akkor van szerepe, ha az ablakot az X gombbal zárják be, tehát nem adtak választ.

**Lekerdez.java**

Fejlesztői lehetőség! Eben az ablakban egy SQL lekérdezést futtahatunk le, tesztelés céljából. Az adatok egy TextArea-ban jelennek meg. Biztonsági okokból csak lekérdezéseket lehet végrehajtani, hogy esetlegesen hozzáértő emberek ne tudjanak belekontárkodni az adatokba.

**Mennyiseg.java**

Az italok mértékegységének eldöntését végzi. Akkor nyílik meg, ha egy egységárral rendelkező árut akarunk megrendelni. A választási lehetőségeink: 2cl, 4cl, 1dl, 2dl, 3dl, 4dl, 5dl, 1l. Az egységár mértéke 1 liter. A kevesebb mennyiségek arányosan osztják le a liter árával. A visszatérési érték egy két elemű tömb, aminek az első eleme az ár, a második a mennyiség. Az itt visszatérő ár miatt szerepel a rendelések táblában egy fizetendő mező, mivel 5dl sör nem ugyanannyiba kerül, mint ebből a sörből 1 liter.

**Ultet.java**

Az ablak megjelenítése előtt egy adatbekérést hajt végre. Egy számot vár, ami az újonnan jött vendégek számát jelenti. Lekérdezéssel kideríti, melyek azok az asztalok, ahol elférnek és még nem ülnek ott más vendégek. Ez nem kötelező művelet, csupán egy lehetőség, ha túl sok asztal van. A potenciális asztalok közül egyre kattintva ez a párbeszédablak bezáródik és kijelölődik a választott asztal a fő ablakban.

**Uzen.java**

Ez egy közönséges üzenet párbeszédablak. Csak azért létezik, hogy jobban nézzen ki, mint egy JOptionPane. A „Keret” osztályból öröklődik.

A „Parbeszed.Modosit” csomag tartalma:

Ebben a csomagban helyezkedik el az **AruModosit.java**, **AsztalModosit.java**, **BesorolasModosit.java** és a **TorzsModosit.java** fájl. Ezek az osztályok mind párbeszédablakok, amiken a nevükből adódóan egyes módosításokat hajthatunk végre az adatbázisban. Azért nem kerülnek egyesével bemutatásra, mert ugyan az a feladatuk, csupán a felület elrendezése más. Minden ablak megjelenésekor beolvassa a hozzá tartozó adatokat, ezt követően minden paraméter tetszőlegesen módosítható.

A „Parbeszed.Torol” csomag tartalma:

Itt található az **AruTorol.java** és egy **Torol.java** fájl. Az előbbi ablak áruk törlésére való, az utóbbi pedig alkalmas asztal, besorolás és törzsvendég törlésére is. Ezt úgy éri el, hogy meghíváskor át van adva neki az ablakban megjelenő üzenet, egy SQL lekérdezés, ami feltölti az adott adatokkal az ablakot és egy SQL törlő parancs.

A „Parbeszed.Torol” csomag tartalma:

Tartalma: UjAru.java, UjAsztal.java, UjBesorolas.java. Azért nem került ide egy külön osztály új törzsvendég felvételére is mivel az „UjAsztal” osztály erre is megfelel.

**III.3. Saját osztálykönyvtárak**

A „Libraries” közé hozzá lett adva egy FLBR.jar nevű fájl. Ez egy saját fejlesztésű osztály, ami hasznos metódusokat és konstansokat tartalmaz. Ez használja az sqlite-jdbc.jat-t is. Itt helyezkednek el az adatbázis csatlakozásához és operálásához szükséges metódusok is.

/\*\*

\* Ez a metódus egy sqlite adatbázishoz hozza létre a kapcsolatot.

\* @param fajl Az adatbázis neve és kiterjesztése.

\*/

**public static void** **kapcsolodik**(String fajl) {

**try**{

kapcsolat=DriverManager.getConnection("jdbc:sqlite:"+fajl);

parancs = kapcsolat.createStatement();

}**catch**(Exception var2){

var2.printStackTrace();

System.exit(1);

}

}// kapcsolodik metódus vége.

/\*\*

\* Ez a metódus egy sqlite adatbázis kapcsolatát bontja.

\*/

**public static void** **kapcsolatBont**() {

**try**{

kapcsolat.close();

}**catch**(Exception var1){

var1.printStackTrace();

System.exit(0);

}

}// kapcsolatBont metódus vége.

/\*\*

\* Ez a metódus egy sql parancsot hajt végre.

\* @param s A végrehajtandó sql parancs.

\*/

**public static void** **vegrehajt**(String s) {

**try**{

parancs.execute(s);

}**catch**(Exception var2){

var2.printStackTrace();

System.exit(0);

}

}// vegrehajt metódus vége.

/\*\*

\* Ez a függvény egy sql lekérdezést hajt vége.

\* @param s A végrehajtandó sql lekérdezés.

\* @return String tömb generikus vektor. Vektor elemei a sorok, \* tömb elemei az oszlopok.

\*/

**public static** Vector<String[]> **lekerdez**(String s){

**try**{

Vector<String[]> v = **new** Vector();

ResultSet ere = parancs.executeQuery(s);

ResultSetMetaData meta = ere.getMetaData();

**int** n = meta.getColumnCount();

**while**(ere.next()) {

String[] t = **new** String[n];

**for**(**int** i = 1; i <= n; ++i) {

t[i - 1] = ere.getString(i);

}

v.add(t);

}

**return** v;

}**catch**(Exception var7){

var7.printStackTrace();

**return null**;

}

}// lekerdez függvény vége.

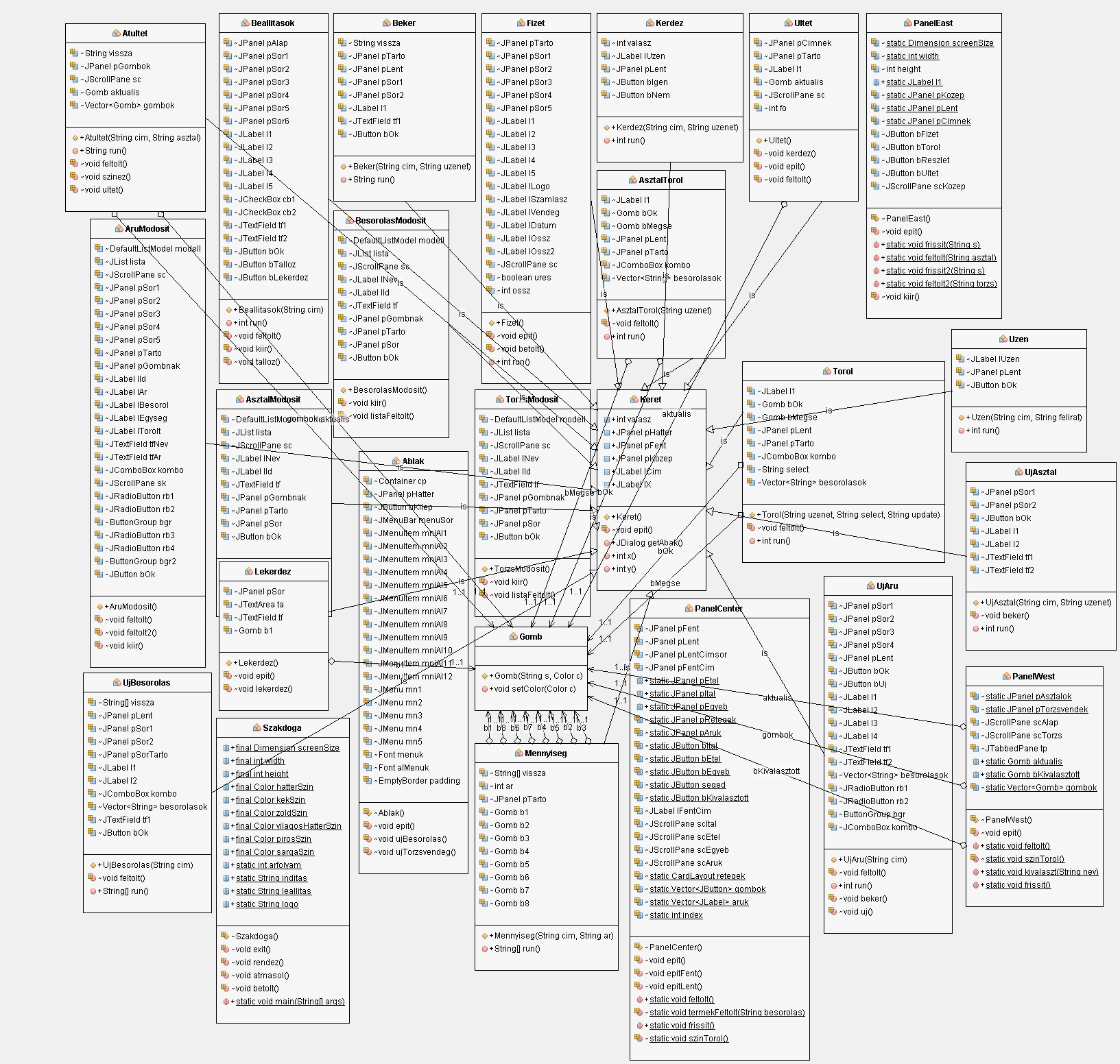
Az ezekben a metódusok használt változók globálisan vannak definiálva.

**public static** Connection kapcsolat;

**public static** Statement parancs;

**public static** Vector<String[]> sel = **new** Vector();

**III.4. UML diagram**

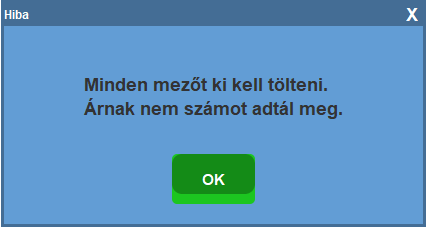


**IV. Tesztkörnyezet**

A program tesztelve lett Windows és Mac OS alatt is, azonban a telepítőkészlet elkészítése után már csak Windows operációs rendszeren futtatható, ugyanis az elérési utak ahonnan fut, csak ott léteznek.

Különböző felbontású és mérető képernyőkön is le lett tesztelve, méret szerint 11”-től 15”-ig. Az ablak teljes mértékben reszponzív, a szövegek olvashatóak és a gombok is lehetőségükhöz mérten elférnek. Nagyon minimális képernyő méretek esetére a program nem enged megadni például asztalnévnek 10 karakternél többet.

A beviteli adatok minden esetben le vannak kezelve, nem lehetséges üres mező bevitele, és ahol számot vár a program, ott csak azt fogadja el. A mínusz számok megengedett felvitele nem hiba, ugyanis létezik például a „pizza levét” lehetőség, ami annyit tesz, hogy ha egy pizzára nem kérnek egy feltétet, azt le lehet venni róla és ez leszámítódik az árból is. A hibaüzenet ablakot a 18. ábrán látható példa szemlélteti.



*(18. ábra)*

**V. Összefoglalás**

Érzésem szerint sikerült megvalósítanom azt, amit elképzeltem. Természetesen még rengeteg mindent lehet hozzáadni a programhoz, de az rendelkezésre álló idő ennyit engedett. Egyetlen komponenst kifejlesztése maradt le, ami pedig a részletfizetés. Ezen kívül van még pár ötlet a fejlesztéshez:

* Tényleges számla nyomtatása egy USB eszközön.
* Bevélek lebontása időszakokra, napi zárás.
* Egy telefonos alkalmazás, amin fel lehet venni a rendeléseket és a központi géphez érkeznek be ebbe a programba. Ehhez tartozna egy bejelentkezési rendszer, hogy követni lehessen ki mit csinál.
* Készletvezetés valós időben, raktárkészlet ellenőrzés.

Az adatbázisban szereplő törzsvendégek nevei nem valós személyek és a termékek árai sem a valóságot tükrözik.

**Záródolgozat értékelő lap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Koncepció** | **Módszer** | **Felépítés** | **Stílus** |
| **Jeles** | A témaválasztás kifejtése és indoklása kellő részletességgel történik. A vizsgázó egyéni kutatási eredményei világosan megfogalmazódnak. | A vizsgázó a kérdésfelvetése szempontjából megfelelő módszert választ, ezt kellő- képpen meg is indokolja. Adatai elemzése után csakis olyan következ- tetéseket von le, amelyek megfelelően alá vannak támasztva. | A dolgozat felépítése logikus, jól tagolt. A követelményrendszer elemei teljesülnek. A vizsgázó a formai követelményeket betartja. | A gondolatok megfogalmazása egyértelmű, a szövegre jellemző a választékos és szabatos szóhasználat, a pontos mondatszerkesztés. A vizsgázó a szakszavakat megfelelően használja. |
| **Jó** | A témaválasztás kifejtése és indoklása megtörténik, de a szakirodalmi beágyazottság csak részlegesen valósul meg. | A vizsgázó a kérdés- felvetése szempont- jából megfelelő módszert választ, de a módszer kiválasz- tásának indoklása nem kellően alapos. | A dolgozat felépítése követi az előírásokat, csak apróbb szerkezeti aránytalanságot tartalmaz. A vizsgázó a formai követelményeket betartja. | A gondolatok megfogalmazása általában egyértelmű. A szövegben csak kisebb nyelvi- nyelvtani hibák fordulnak elő. |
| **Köze- pes** | A témaválasztás kifejtése és indoklása megtörténik ugyan, de a kérdésfelvetés szempontjából fontos szakirodalmat a vizsgázó nem veszi figyelembe. | A vizsgázó olyan módszert választ, mely csak korlátozottan alkalmas a kérdésfelvetés megválaszolására, adatgyűjtését kisebb hiányosságok mellett dokumentálja. | A dolgozat felépítése nagyjából követi az előírásokat, előfordulnak szerkezeti aránytalanságok. A vizsgázó a formai követelményeket betartja. | A gondolatok megfogalmazása néhol homályos, előfordulnak benne kisebb mondatszerkesztési és nyelvtani hibák. A szakszavak használata néhol pontatlan. |
| **Elég- séges** | A témaválasztást a vizsgázó legfeljebb csak egyéni érdeklődésével indokolja. | A vizsgázó ad-hoc módon választ módszert, választását nem indokolja. Olyan következtetéseket von le, melyek nem állják meg a helyüket, ellentmondásokhoz vezetnek. | A főbb gondolatok nem különülnek el egymástól (pl. összefüggésbeli hiányosságok). | A gondolatok megfogalmazása több esetben nem egyértelmű, a dolgozat nyelvhasználata nem elég igényes, és több nyelvtani hibát tartalmaz. A szakszavak használata bizonytalan. |
| **Elég- telen** | A témaválasztás kifejtése és indoklása nem történik meg. | A vizsgázó ad-hoc módon választ módszert, mely a kérdésfelvetés megválaszolására nem alkalmas, A vizsgázó következtetései mellett nem érvel. | A dolgozat nem teljesíti az előírt követelményeket. | A dolgozat szókincse egysíkú, mondat- szerkesztése nem alkalmazkodik a dolgozat normáihoz. Gyakori a szövegben a nyelvtani hiba. Nem használ, vagy rosszul alkalmazza a szakszavakat. |