SZAKDOLGOZAT

Márta Krisztián

NYÍREGYHÁZA, 2023



**Angol szavak tanulását segítő alkalmazás fejlesztése Java nyelven**

Márta Krisztián

Pogramtervező informatikus BSc

Dr. Kerényi Gábor, Külső óraadó

**2023**

**Tartalomjegyzék**

[1. Bevezetés 5](#_Toc132666564)

[1.1. Témaválasztás indoklása 5](#_Toc132666565)

[1.2. Célkitűzés 5](#_Toc132666566)

[2. Nyelvoktató alkalmazások 6](#_Toc132666567)

[2.1. Történelmi áttekintés 6](#_Toc132666568)

[2.1.1. Az angol nyelv rövid történelmi áttekintése 6](#_Toc132666569)

[2.1.2. Az oktatás és a számítástechnika kapcsolatának fejlődése 7](#_Toc132666570)

[2.1.3. Az oktatás és a számítástechnika kapcsolata napjainkban 8](#_Toc132666571)

[2.2. Duolingo 9](#_Toc132666572)

[2.3. Quizlet 10](#_Toc132666573)

[3. Java programozási nyelv bemutatása 11](#_Toc132666574)

[3.1. Alkalmazásfejlesztés Javában 11](#_Toc132666575)

[3.2. A Java nyelv főbb jellemzői 12](#_Toc132666576)

[3.3. Adattípusok 13](#_Toc132666577)

[3.4. Vezérlési szerkezetek 14](#_Toc132666578)

[3.4.1. Feltételes utasítások 14](#_Toc132666579)

[3.4.2. Ciklusszervező utasítások 15](#_Toc132666580)

[4. A tervezés menetének leírása 16](#_Toc132666581)

[4.1. A követelmény specifikáció meghatározása 16](#_Toc132666582)

[4.1.1. Regisztrációs felület 16](#_Toc132666583)

[4.1.2. Bejelentkezési felület 17](#_Toc132666584)

[4.1.3. Menü felülete 17](#_Toc132666585)

[4.1.4. Tematikus szószedeteket tartalmazó felület 17](#_Toc132666586)

[4.1.5. Szókártyákat tartalmazó szótanuló felület 17](#_Toc132666587)

[4.1.6. Tesztfeladatsor felülete 17](#_Toc132666588)

[4.1.7. Szerkesztőnézet felülete 17](#_Toc132666589)

[4.1.8. Ranglista felület 17](#_Toc132666590)

[4.2. Adatbázis megtervezése 17](#_Toc132666591)

[5. A fejlesztés menetének leírása 17](#_Toc132666592)

[6. Eredmények 17](#_Toc132666593)

[7. Összefoglalás 17](#_Toc132666594)

[8. Irodalomjegyzék 18](#_Toc132666595)

[9. Ábrajegyzék 20](#_Toc132666596)

[10. Táblázatjegyzék 21](#_Toc132666597)

# Bevezetés

## Témaválasztás indoklása

Tanulmányaim során az informatikai sokféle ágazatával megismerkedtem, melyek közül a grafikus felhasználói felülettel rendelkező alkalmazások tervezése és fejlesztése áll hozzám a legközelebb. Jó néhány ilyen alkalmazást készítettem már, kezdve az egyszerűbbektől a bonyolultabbakig, melyeket főként Java és C# programozási nyelvek segítségével fejlesztettem. Így szakdolgozatom témájának kiválasztásakor egyértelmű volt számomra, hogy ebbe az irányba induljak el.

Ezt követően döntést hoztam arról, hogy milyen nyelven és melyik integrált fejlesztői környezetben fejlesszek. A Java nyelvre és a Apache NetBeans fejlesztői környezetre esett a választásom, melyekkel a tanulmányaimból adódóan sok tapasztalattal rendelkezem, és célom, hogy még több tapasztalatot szerezzek a fejlesztés alatt.

Miután minden tisztázódott bennem a fejlesztési és technikai kérdésekkel kapcsolatban, már csak az alkalmazás tematikáját kellett kiválasztanom. Ekkor épp angolt tanultam, ebből adódóan jött az ötlet, hogy az angol tanulás köré építsem fel az alkalmazást, valamint mindenképp szerettem volna, olyan applikációt fejleszteni, mely hasznos és könnyen kezelhető a felhasználó számára. Így született meg, hogy a szakdolgozatom témája, ami nem más, mint angol szavak tanulását segítő alkalmazás fejlesztése Java nyelven.

## Célkitűzés

Célkitűzésem, hogy egy olyan alkalmazást készítsek, melynek segítéségével a felhasználó hatékonyan és könnyedén memorizálhatja az angol szavakat és azok magyar jelentését. Szeretném segíteni a tanulást képek segítségével, hiszen egy információ hosszútávon való raktározása sokkal egyszerűbb, ha egy képhez kötjük. Ezen felül az alkalmazásban lehetővé teszem a felhasználó számára, hogy személyre szabhassa a tanulási folyamatát, valamint próbára tegye megszerzett tudását teszt segítségével, melynek végén értékelést kap.

Több felhasználó számára is szeretném biztosítani a személyre szabott tanulási folyamatot, ezért többfelhasználós szoftver készítését tervezem.

# Nyelvoktató alkalmazások

A történelmi áttekintésnél fontosnak tartottam, hogy bemutassam az angol nyelv történetét, az oktatás és a számítástechnika kapcsolatát, valamint a digitalizáció szerepét a nyelvoktatásban.

A nyelvek tanulása közben egy szó vagy egy kifejezés jelentésének nagyobb értelmet adhatunk azáltal, ha ismerjük az eredetét. Ilyenkor a nyelv történelmének egy szakaszát akár részletesebben is megismerhetjük, mely nagyban elősegíti a nyelv elsajátítását. Ennek a folyamatnak a megalapozását szeretném elősegíteni, azzal, hogy bemutatom az angol nyelv történelmét röviden. [1]

Az emberek a mai világban már a digitális eszközöknek köszönhetően sokkal könnyebben tehetik magukévá a tudást, mint felmenőink. Sokan nem is tudják elképzelni, hogy az elődjeink, hogyan tudtak tanulni digitális eszközök nélkül, emiatt tartottam fontosnak bemutatni az oktatás és a számítástechnika kapcsolatának fejlődését, valamint a digitalizáció szerepét a nyelvoktatásban.

A történelmi áttekintés után pedig szeretnék bemutatni röviden két ismertebb nyelvoktató alkalmazást.

## Történelmi áttekintés

### Az angol nyelv rövid történelmi áttekintése

Az angol nyelv az indoeurópai nyelvcsaládba tartozik, és a nyelv a Brit-szigeteken alakult ki. Nagy-Britannia őslakossága a kelta népcsoport volt, akiket a Római Birodalom idején Julius Ceaser kiszorított a mai Anglia területéről a walesi és a skót hegyekbe. A rómaiak kivonulása után nem sokáig örülhettek a kelták, hiszen 450 körül jött 3 germán törzs (angelek, szászok, jüttök) és újra kiszorították a kelta népet a hegyekbe, akik magukkal is vitték a kelta nyelvet. Ekkor kezdődött el az angol nyelv fejlődése, mely 3 korszakra osztható. [1][2]

Az első korszak az Óangol, mely 450-től 1066-ig tehető. Másnéven angolszász nyelvnek is nevezzük, ennek elnevezése a két nagy törzs, az angol és a szász elnevezésből ered. Az angol nyelv őse a fentebb említett megszálló népek nyelve. A vikingek gyakran betörtek az országba, így kerültek be az óangol nyelvbe dán és norvég szavak. Ennek következtében történt, hogy elkezdték elhagyni a ragokat, hogy a dánok is megértsék mit mondanak. Jelentős hatása volt még az óangol nyelvre a latinnak, hisz az írni és olvasni tudó emberek latinul tudtak, ennek hatására rengeteg jövevényszó került a nyelvbe. [1][2]

Ezt követő korszak a középangol, mely 1066-től 1500-ig tartott. Hódító Vilmosnak köszönhetően Anglia normann uralom alá került, ami azt eredményezte, hogy a francia lett a hivatalos nyelv, és ennek erős hatása volt az angolra. Francia anyanyelvű vezetők irányították az állami hivatalokat, viszont az átlagemberek még angolul beszéltek, ezek keveredése miatt alakult ki a középangol. Az 1348-as nagy pestis járvány elpusztította a lakosság egyharmadát, a legtöbb vezetővel együtt, ezután újból angol anyanyelvű emberek kerültek az állami hivatalok élére. A hosszú időn át tartó francia behatásnak nagy mértékű hatása volt a nyelvre. [1][2]

A harmadik korszak a modern angol, amely a XV. századtól kezdve egészen napjainkig tart. Ennek az időszaknak a kezdetén már hasonlított a nyelv a maihoz, bár kisebb változások történtek. Készült egy egységes helyesírási rendszer, lefordították a Bibliát, változott a kiejtés, megszűnt a tegező forma, a nyelvtani szabályok leegyszerűsödtek és ezek mellett Shakespeare munkásságának is nagy szerepe volt, hogy kialakuljon a napjainkban használatos nyelv. [1][2]

### Az oktatás és a számítástechnika kapcsolatának fejlődése

Ősrégi technológiák az osztályteremben:

* 1650: A gyarmati korszakban népszerűek voltak a fából készült, nyomtatott leckével ellátott könyvek, amelyek segítették a diákokat a versek megtanulásában. [3]
* 1870: A diagépek elődje, a Magic Lantern vetítette ki az üveglemezekre nyomtatott képeket. [3]
* 1890: Megjelentek a krétatáblák és az iskolai táblák. [3]
* 1900-ban megjelentek a ceruzák, mellyel papírra írtak és elterjedése elavulttá tette az iskolai táblákat. [3]
* Az 1920-as években a tanulás új formája alakult ki, a rádiós tanórák révén. A New York-i Oktatási Tanács rádióállomáson keresztül közvetítette az órákat a hallgatóknak. [3]
* 1930-ban bevezették az írásvetítőt [3]
* 1940-ben megjelent a golyóstoll. [3]
* 1950-ben megjelent a fejhallgató, aminek segítségével a leckéket ismétléssel lehet megtanulni, ennek következtében jöttek létre a nyelvi laboratóriumokat, melyek napjainkban is használatosak. [4]
* 1970-ben megjelentek a kézi grafikus számológépek, melyek nagy sikert arattak a matematika területén. [4]
* 1972-ben megjelent a Scantron tesztelési rendszer, amely lehetővé tette a tanárok számára, hogy hatékonyabban és eredményesebben osztályozzanak. [3]

A személyi számítógépek kora:

* A számítógépek a 80-as években kerültek a mindennapi használatba. [3]
* 1981-ben mutatták be az első hordozható számítógépet. [3]
* 1984-ben az Egyesült Államokban az állami iskolákban átlagosan 92 diákra jutott egy számítógép. [3]
* Az 1985-ös évben jelent meg szintén a CD-ROM meghajtó. Egy lemezen rengeteg adatot lehetett tárolni. [3]
* 1996: Az internet eljutott az oktatási intézményekbe. [3]
* 1999-ben megjelent az interaktív táblák, melyek a kézzel írt jegyzetek és az interaktív technológia keverékei. A korai változatok vezetékkel csatlakoztak az asztali számítógépekhez, míg a legújabbak kompatibilisek okostelefonokkal és táblagépekkel. [3]
* 2007-2010: Az okostelefonok és a táblagépek kora. Megjelent az iPad, ami új magasságokba emeli az osztálytermi oktatást. [3]

### Az oktatás és a számítástechnika kapcsolata napjainkban

Az oktatás és a számítástechnika közötti kapcsolat nagyon fontos szerepet tölt be a mai modern társadalomban. A technológia rohamos fejlődése új módszerek és eszközök alkalmazására ad lehetőséget az oktatás területén. Az oktatási intézményekben az informatikai eszközök, valamint az oktatást segítő alkalmazások száma folyamatosan nő, melyek segítik a tanárok és a diákok életét.

A diákok már nem toll, papír és jegyzetfüzet segítségével jegyzetelnek az órákon, hanem okostelefonjukon, laptopjukon, tabletjeiken. Vannak olyan diákok is akik a távoktatás lehetőségével élve otthonról végzik tanulmányaikat. A tanárok pedig könnyebben és interaktívabban adhatják át a tudásukat a diákoknak, valamint digitális eszközökkel, interaktív táblákkal, prezentációkkal és videókkal teszik élményszerűbbé az oktatást. [5]

A mai digitalizált világban viszont már nem a tanár a legfőbb információforrás a diákok számára, hisz az internet segítségével bárhonnan, bármikor és bárhogyan hozzájuthatnak információhoz. [5]

Az interneten található számos információforrás, ilyenek például az alábbiak:

* Videómegosztó portálok
* E-learning platformok, rendszerek
* Ingyenes és fizetős kurzusok
* Szakmai fórumok
* Online szótárak, fordítók
* Tudásbázisok
* Alkalmazások mindenféle platformra
* Digitálisan elérhető könyvek, könyvtárak
* Podcastek
* Online lexikonok

Fontos, hogy ha ezeken a forrásokat vesszük igénybe tájékozódjunk arról, hogy mennyire hitelesek.

## Duolingo



1. ábra: Duolingo logó [6]

2013-ban a Duolingot (1. ábra) a legjobb nyelvtanuló alkalmazásnak választotta a Google. Az alkalmazásban több mint 40 nyelv közül választhatunk. A játékos oktatásra épül, ugyanis szinteket kell teljesíteni, melyek egy adott tematikára épülnek. A szintek végén a felhasználó próbára teheti tudását a szintugró tesztekkel. Található benne egy olyan funkció, amely a mindennapi és rendszeres tanulásra ösztönöz, ez pedig a széria számláló, amely megmutatja, hogy hány napot töltöttünk sorozatban tanulással. A példamondatok mindig egy-egy új szót tartalmaznak. Ezeket ismétlés útján megtanulhatjuk hallás utáni értéssel, kiejtéssel, fordítással, vagy akár feleletválasztós kérdésekkel is. [7]

Előny:

* Csak regisztrációra van szükség a tanulás elkezdéséhez.
* Azonnal leellenőrizhető a feladat.
* Számítógépen és okostelefonokon is működőképes.
* Nem kell egyetlen egy leckéért sem fizetni.
* Magyar nyelven is elérhető a felhasználói felület.
* Átlátható és könnyen kezelhető felhasználói felület.

Hátrány:

* A zajos tanulás zavarhatja a környezetünket.
* A mondatok elég gyakran nem életszerűek.

## Quizlet



2. ábra: Quizlet logó [8]

A Quizlet (2. ábra) segítségével lehetőségünk van szókártyákat készíteni. Idegen nyelv tanulásakor lehet a kártya egyik oldala az idegen nyelvű szó, másik pedig a magyar jelentése. Használhatjuk fogalmak és a hozzájuk tartozó definíciók gyakoroltatására is, ilyenkor viszont nagyon figyelni kell arra, hogy a hosszú válaszok beírása nehézkes, mert a program csak a pontosan megegyező karaktersorozatot fogadja el. Generálhatunk vele tesztfeladatsorokat, mely változatos feladatokat ad, valamint ki is nyomtathatjuk a szókártyákat. [9]

Előny:

* Csak regisztrációra van szükség a tanulás elkezdéséhez.
* Számítógépen és okostelefonokon is működőképes.
* A szókártyák addig gyakorolhatók, amíg minden szó rögzül.
* Könnyen kezelhető felhasználói felület.

Hátrány:

* Egy szókártyakészlet maximum 50 szót tartalmazhat.
* Csak angol nyelvű felhasználói felület.

# Java programozási nyelv bemutatása



3. ábra: Java programozási nyelv logója [10]

A Java programozási nyelvet (3. ábra) az 1990-es évek közepén fejlesztette ki a Sun Microsystems. A Java alapítója, James Gosling az Oak nevet szerette volna adni az új nyelvnek, de már létezett ilyen nevű programozási nyelv. A legenda szerint a Jáva szigetéről nevezték el a nyelvet, mivel a fejlesztés során rengeteg a szigetről származó kávét fogyasztottak el. [11]

A Java megnevezés nem csak egy programozási nyelvet jelöl, hanem a programok megírásához és futtatásához szükséges szoftverkörnyezetet, amit Java platformnak nevezünk. A platform részei a Java program, a Java fordító (compiler), a programokat futtató virtuális gép (JVM) és egy fejlesztői programcsomag (Java SDK). A JVM (Java Virtaul Machine) futtatja a számítógépen a programokat, mely biztosítja hardvertől és operációs rendszertől való függetlenséget, azaz a hordozhatóságot. A Java SDK (Java Software Development Kit), vagy ahogy napjainkban hívják JDK (Java Development Kit), egy előre kidolgozott sokoldalú fejlesztői programcsomag, mely szabadon felhasználható komponenseket és könyvtárakat tartalmaz. [11]

## Alkalmazásfejlesztés Javában

Amennyiben Java nyelven szeretnénk programok készíteni, csak egy egyszerű szövegszerkesztőre van szükségünk (Jegyzettömb, Notepad++), illetve a Java fordítóprogramra, amely lefordítja a forráskódot bájtkódra. A lefordított bájtkódot ezután a JVM tudja lefuttatni (4. ábra), melyet akár parancssorból is elvégezhetünk. Az említett módszer eléggé körülményessé teszi a fejlesztést, azonban sokan hívei még ennek. [12]



4. ábra: A forráskódtól a programig [13]

A legtöbb modern programozási nyelvhez elérhető egy IDE (Integrated Development Environment), magyarul integrált fejlesztői környezet, mely tartalmaz egy a nyelvre szabott szövegszerkesztőt, a fordítóprogramot, hibakeresőt, tesztelési lehetőségeket és számos egyéb, a fejlesztést segítő eszközt. A Javához is érhető el ilyen integrált fejlesztői környezet a legismertebbek a NetBeans, Eclipse és IntelliJ. Ezek közül majdnem mindegy melyikkel dolgozunk, hiszen ezek a fejlesztői környezetek minden segítséget megadnak a fejlesztéshez, nekünk csupán annyi dolgunk van, hogy szintaktikailag és szemantikailag is helyes Java forráskódot készítsünk, amelyet a fordító működőképes bájtkódra tud fordítani. [12]

A lefordított bájtkódot a JRE (Java Runtime Environment), futtatja le az adott hardverkörnyezetben. A JRE, vagyis a Java futtatható környezet magában foglalja a virtuális gépet (JVM), valamint a futtatáshoz szükséges legfontosabb osztályokat is. [12]

## A Java nyelv főbb jellemzői

* Platformfüggetlen, azaz hordozható: Akármilyen operációs rendszeren fejleszted az alkalmazásodat, azt egy másik operációs rendszeren is lehet majd futtatni, amennyiben az támogatja a Java futtatókörnyezetet. A fejlesztést követően a Java fordító lefordítja bájtkódra a kódot, melyet a JVM értelmezni tud. Mivel a virtuális gép minden platformon fut, a Java kódodat is tudod majd futtatni bárhol. [14]
* Objektumorientált: Alapja az objektum, melynek attribútumai jelentik az objektumban tárolt adatokat, a metódusok pedig a rajtuk értelmezhető és végezhető műveleteket. A második alappillér az öröklődés, mely azt jelenti, hogy egy osztályból létrehozhatunk egy másik osztályt, úgy, hogy az örökli az ősosztály minden attribútumát és metódusát, ezenfelül újabbakkal is ki tudjuk egészíteni. A harmadik alappillér a többalakúság, vagy másnéven polimorfizmus, ami annyit jelent, hogy egy adott metódus azonosítója közös lehet egy osztályhierarchián belül, viszont a hierarchia minden osztályában a tevékenységet végrehajtó metódus megvalósítása specifikus lehet. [12]
* Biztonságos: Megbízható, vagyis robusztus a Java, mert a programozási hibák egy részét megakadályozza, másik részét pedig kiszűri futás közben ezzel is segítve a fejlesztő munkáját. Megakadályozza a rosszindulatú programok rendszerbe kerülését is, mivel a virtuális gép csak a szigorú szabályoknak megfelelő bájtkódot fordítja le. Szintaktikailag követi a C++-t, viszont sokkal biztonságosabb és megbízhatóbb nyelvet építettek, köszönhető ez annak, hogy a Javában nincs goto utasítás, nincsenek mutatók, nincs többszörös öröklődés, nincs rekordtípus, és az operátorok nem felülírhatók. [11]

## Adattípusok

Egy fejlesztés során nagyon sok adattal kell dolgoznunk, és ezekről el kell döntenünk, hogy milyen értéket képviselnek és milyen műveleteket végezhetünk velük. Kétféle adattípust különböztetünk meg Javában, a primitív és a referencia adattípusokat.

I. táblázat: Primitív adattípusok [15]

|  |  |  |
| --- | --- | --- |
| Típus | Leírás | Méret/Formátum |
| byte | bájt méretű egész szám | 8 bit |
| short | rövid egész szám | 16 bit |
| int | egész szám | 32 bit |
| long | hosszú egész szám | 64 bit |
| float | egyszeres pontosságú lebegőpontos valós szám | 32 bit |
| double | dupla pontosságú lebegőpontos valós szám | 64 bit |
| char | Unicode karakter | 16 bit |
| boolean | logikai érték | true vagy false |

Referencia adattípusok közé tartozik a tömb, az osztályok, és az interfészek.

## Vezérlési szerkezetek

### Feltételes utasítások

if szerkezet: A benne lévő utasítások, csak akkor futnak le, ha az if-ben megadott feltétel teljesül. Ha egyetlen egy utasítás adunk meg az if szerkezetben, akkor nem kell utasításblokkot használnunk.

Szintaktika:

if (feltétel) {

utasítás(ok);

}

Az if szerkezet kibővülhet egy else ággal. Az else ágban lévő utasítások csak akkor futnak le, ha az if ág feltétele nem teljesül. Egy if utasításnak csak egyetlen egy else ága lehet.

Szintaktika:

if (feltétel) {

utasítás(ok);

} else {

utasítás(ok);

}

Az else if szerkezetnek köszönhetően több if ágat is egymásba lehet ágyazni. Ha az első feltétel teljesül, akkor az ahhoz tartozó utasítások fognak végrehajtódni, ha a második feltétel teljesül, akkor az ahhoz tartozó utasítások fognak végrehajtódni és ez így tovább. Amennyiben nem teljesül egyik feltétel sem, akkor a végén található else ág parancsai fognak lefutni.

Szintaktika:

if (feltétel) {

utasítás(ok);

} else if (feltétel) {

utasítás(ok);

} else if (feltétel) {

utasítás(ok);

} else {

utasítás(ok);

}

switch case szerkezet: A switch-case segítségével tudjuk egyszerűbbé tenni a hosszú if else szerkezeteket. A switch kap egy kifejezést, melyet először kiértékel. Ha talál egyezést, akkor az adott case ágban lévő utasításokat lefuttatja, majd az ezután következőket is. Azonban, ha csak a kifejezésnek megfelelő case ág utasításait szeretnénk végrehajtani, akkor minden case ágat megkell szakítani, melyet a break kulcsszó segítségével tudunk megvalósítani.A szerkezet legvégén található egy default ág, melynek utasításai akkor futnak le, ha a kifejezés értékét egyik case ág sem kezelte le.

Szintaktika:

switch (kifejezés) {

case érték:

utasítás(ok); break;

case érték:

utasítás(ok); break;

case érték:

utasítás(ok); break;

default:

utasítás(ok); break;

}

### Ciklusszervező utasítások

A ciklus vagy másnéven iteráció az ismétlődő utasítások végrehajtására szolgál. A ciklusok mindig tartalmaznak egy feltételt, és a ciklusmagban lévő utasítások újra ás újra lefutnak, mindaddig amíg a feltétel igaz. Háromféle ciklusszervező utasítást különböztetünk meg, melyeket röviden bemutatok.

while ciklus: A while ciklus egy feltételes ciklus, amely előltesztelő. A feltételes ciklusokat általában akkor használjuk, ha nem tudjuk a ciklusról, hogy hányszor kell lefutnia futtásidőben. A while ciklus megvizsgálja először a feltételt és ha igaz, akkor lefut a ciklusmag és kezdődik elölről, ha nem igaz kilép a ciklusból. Előfordulhat, olyan is, hogy a while ciklus egyetlen egyszer sem fut le.

Szintaktika:

while (feltétel) {

utasítás(ok);

}

do-while ciklus: A do-while ciklus úgyszintén feltételes ciklus, amely hátultesztelő, ebből következik, hogy a ciklusmagban szereplő utasítások egyszer mindenképpen végrehajtódnak. A ciklusmag legelső lefutása után kiértékeli a feltételt, amely, ha igaz, akkor újra fut a ciklus, ha nem akkor pedig kilép a ciklusból.

Szintaktika:

do {

utasítás(ok);

} while (feltétel);

for ciklus: A for ciklus egy számláló ciklus, de szokták még nevezni növekményes ciklusnak is.

Szintaktika:

for( inicializálás; feltétel; ciklusváltozó növelése ) {

utasítás(ok);

}

Az inicializálás az, ahol deklaráljuk a ciklusváltozót és adunk neki egy kezdeti értéket. A feltétel a belépési feltétel, amely, ha igaz akkor, végrehajtódik a ciklusmag, egyébként kilép a ciklusból. Ciklusváltozó növelése alatt pedig a ciklusváltozó értékének változtatását értjük, ami nem feltétlen csak növelés lehet, hiszen csökkenthetjük is a ciklusváltozó értékét.

# A tervezés menetének leírása

## A követelmény specifikáció meghatározása

Egy szoftver tervezése és fejlesztése során a legfontosabb kezdeti lépés, a követelmények rögzítése, ezt nevezzük követelmény specifikációnak. A specifikációk a fejlesztés során nagyban elősegíti a munkánkat, hiszen tartalmaz minden olyan funkciót és annak leírását, amelyeket a programnak tudnia és tartalmaznia kell. Az alkalmazásom esetében ezen specifikációkat felületenként elkülönítve fogom kifejteni.

### Regisztrációs felület

Célkitűzéseim között említettem már, hogy szeretném, ha több felhasználó is tudná használni az alkalmazást személyre szabottan. Ebből kifolyólag az alkalmazás rendelkezik egy regisztrációs felülettel, ahol a felhasználó adatait regisztrációkor eltudjuk tárolni adatbázisba.

A regisztráció során a felhasználónak az alábbi adatokat kell megadnia:

* e-mail cím
* felhasználónév
* jelszó
* jelszó újra

A jelszó és a jelszó újra beviteli mezők mellett található egy-egy szem ikon, amely arra szolgál, hogy segítségével felfedhessük, majd újra elrejthessük a jelszó mezők tartalmát.

A regisztráció a „REGISZRÁCIÓ” feliratú gombra való kattintással, vagy az ENTER billentyűzet leütésére történik meg, ha minden megadott adat helyes, hiszen megkell vizsgálnunk a beírt adatok helyességét. Ha hibásan megadott adatokat észlel az alkalmazás akkor azt jelzi hibaüzenetek segítségével.

Hibás adatok az alábbiakat lehetnek:

* Ha üresen hagyta a felhasználó bármelyik mezőt.
* Ha a felhasználó által megadott e-mail cím formátuma nem megfelelő.
* Ha a felhasználó által megadott e-mail címről már regisztráltak.
* Ha a felhasználó által megadott felhasználónév már foglalt.
* Ha a felhasználó által megadott jelszó rövidebb, mint 8 karakter.
* Ha a felhasználó által megadott jelszó nem tartalmaz legalább egy kisbetűt, egy nagybetűt és egy számot.
* Ha a felhasználó által megadott jelszavak nem egyeznek.

Amennyiben minden megadott adat helyes, akkor az adatokat el kell mentse a megfelelő adatbázisba, majd tájékoztassa a felhasználót a sikeres regisztrációról és irányítsa át a bejelentkezési felületre.

A regisztrációs felületen biztosítani kell azt is, hogy visszatérhessünk a bejelentkezési felületre, ha mégsem szeretnék regisztrálni. Ezt a „MÁR VAN FIÓKOM” gombra való kattintással érhetjük el.

### Bejelentkezési felület

INNEN FOLYTATÁS

### Menü felülete

### Tematikus szószedeteket tartalmazó felület

### Szókártyákat tartalmazó szótanuló felület

### Tesztfeladatsor felülete

### Szerkesztőnézet felülete

### Ranglista felület

## Adatbázis megtervezése

# A fejlesztés menetének leírása

# Eredmények

# Összefoglalás

# Irodalomjegyzék

1. Az angol nyelv rövid történelmi áttekintése:

<https://behappynyelviskola.hu/az-angol-nyelv-tortenete/?cn-reloaded=1>

(2023. 04. 09.)

1. Az angol nyelv rövid történelmi áttekintése:

<https://tudnivalo.webnode.hu/angol/az-angol-nyelv-eredete/> (2023. 04. 09.)

1. Az oktatás és a számítástechnika kapcsolatának történelmi bemutatása:

<https://elearninginfographics.com/evolution-classroom-technology-journey-pen-keyboard-infographic/> (2023. 04. 09.)

1. Az oktatás és a számítástechnika kapcsolatának történelmi bemutatása:

<https://elearninginfographics.com/the-history-of-classroom-technology-infographic/> (2023. 04. 09.)

1. Horváth Ildikó, A digitális oktatás legújabb eszközei és módszerei, Széchenyi István Egyetem, 2017:

<https://www.hte.hu/documents/3102649/4239802/HT_2017_1_2_Horvath.pdf>

(2023. 04. 09.)

1. 1. ábra: Duolingo logó:

<https://d35aaqx5ub95lt.cloudfront.net/images/f2a2e608c854822ad2563a09595e7827.png> (2023. 04. 10.)

1. Duolingo bemutatása:

<https://www.angliaikisokos.com/az-5-legjobb-ingyenes-nyelvoktato-alkalmazas/>

(2023. 04. 10.)

1. 2. ábra: Quizlet logó:

<https://play-lh.googleusercontent.com/hiQHKRhpuGu4pWAFhpto9H7qWKSdX-BjKbDAtZYgm_jfoD0cN7MCllGOn6L3XWo-6Q> (2023. 04. 10.)

1. Quizlet bemutatása:

<http://refpedi.hu/sites/default/files/hir_kepek/A%20Quizlet%20feladatkészítő_2018_január_BMA_VÉGLEGES.pdf> (2023. 04. 10.)

1. 3. ábra: Java programozási nyelv logója:

<https://logos-download.com/wp-content/uploads/2016/10/Java_logo_icon.png>

(2023. 04. 10.)

1. Kovács Zsuzsanna, JAVA programozási nyelv NetBeans fejlesztőkörnyezetben, Observans Kft., Budapest, 2009:

<https://informatika.gtportal.eu/letoltes/SZINFO13_TJ.pdf> (2023. 04. 10.)

1. Tömösközi Péter, Programozás Javában, Eger, 2013:

<https://mek.oszk.hu/14200/14282/pdf/14282.pdf> (2023. 04. 10.)

1. 4. ábra: A forráskódtól a programig:

<https://www.scaler.com/topics/java/how-java-program-works/> (2023. 04. 10.)

1. A Java nyelv főbb jellemzői:

<https://bluebird.hu/java/> (2023. 04. 10.)

1. Nagy Gusztáv, Java programozás, 2007:

<https://nagygusztav.hu/sites/default/files/csatol/java_programozas_1.3.pdf> (2023.04.11)

# Ábrajegyzék

[1. ábra: Duolingo logó [6] 8](#_Toc132491678)

[2. ábra: Quizlet logó [8] 9](#_Toc132491679)

[3. ábra: Java programozási nyelv logója [10] 10](#_Toc132491680)

[4. ábra: A forráskódtól a programig [13] 11](#_Toc132491681)

# Táblázatjegyzék

[I. táblázat: Primitív adattípusok [15] 13](#_Toc132664074)