

# SZAKDOLGOZAT



MISKOLCI EGYETEM

## Webalkalmazás fejlesztése szervezeti folyamatok kezeléséhez

**Készítette:**

Utry Máté Attila

Programtervező informatikus

**Témavezető:**

Piller Imre

MISKOLC, 2020

MISKOLCI EGYETEM

Gépészmérnöki és Informatikai Kar

Alkalmazott Matematikai Intézeti Tanszék

Szám:

## SZAKDOLGOZAT FELADAT

Utry Máté Attila (YR10UU) programtervező informatikus jelölt részére.

**A szakdolgozat tárgyköre:** kulcsszavak, hasonlók

**A szakdolgozat címe:** Webalkalmazás fejlesztése szervezeti folyamatok kezeléséhez

**A feladat részletezése:**

*A különféle szervezetek (például intézmények, vállalatok) ügymenete nagyon bonyolult, és egyedi. A dolgozat egy olyan alkalmazás elkészítésének bemutatását célozza, amely konkrét szervezeti formától függetlenül képes azok folyamatainak modellezéséhez eszközt biztosítani.*

*A folyamatok alatt itt elsősorban olyan, véges állapotú automatával leírható műveletsorokat értünk, amely esetenként további, összetettebb (például szerepkörökhöz) kötött feltételek, külső állapotok teljesülését is szükségessé teszik. A dolgozatban ezen állapotgép és feltételrendszer leírási módjához szükséges alkalmazást kell megtervezni és implementálni.*

*Az alkalmazásnak webes környezetben, szerver-kliens architektúrának megfelelően kell elkészülnie. Ehhez szerver oldalon a Falcon nevű, Python alapú mikrokeretrendszer ad alapot, míg a kliens megvalósításához egy elterjedt JavaScript alapú keretrendszert kell használni. Az adatokat az alkalmazásnak relációs adatbázisban kell tárolnia.*

**Témavezető:** Piller Imre (egyetemi tanársegéd)

**A feladat kiadásának ideje:**

.....  
szakfelelős

## EREDETISÉGI NYILATKOZAT

Alulírott **Utry Máté Attila**; Neptun-kód: YR10UU a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős Programtervező informatikus szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy *Webalkalmazás fejlesztése szervezeti folyamatok kezeléséhez* című szakdolgozatom saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, ..... év ..... hó ..... nap

.....

Hallgató

1.

szükséges (módosítás külön lapon)

A szakdolgozat feladat módosítása

nem szükséges

.....

dátum

.....

témavezető(k)

2. A feladat kidolgozását ellenőriztem:

témavezető (dátum, aláírás):

konzulens (dátum, aláírás):

.....

.....

.....

.....

.....

.....

3. A szakdolgozat beadható:

.....

dátum

.....

témavezető(k)

4. A szakdolgozat ..... szövegoldalt

..... program protokollt (listát, felhasználói leírást)

..... elektronikus adathordozót (részletezve)

.....

..... egyéb mellékletet (részletezve)

.....

tartalmaz.

.....

dátum

.....

témavezető(k)

5.

bocsátható

A szakdolgozat bírálatra

nem bocsátható

A bíráló neve: .....

.....

dátum

.....

szakfelelős

6. A szakdolgozat osztályzata

a témavezető javaslata: .....

a bíráló javaslata: .....

a szakdolgozat végleges eredménye: .....

Miskolc, .....

.....

a Záróvizsga Bizottság Elnöke

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>1</b>
<b>2. Üzleti folyamatok és modellezésük</b>	<b>2</b>
2.1. Mik azok az üzleti folyamatok? . . . . .	2
2.2. Üzleti folyamatok modellezése . . . . .	4
2.2.1. Modellezés folyamatábrával . . . . .	4
2.3. Egy folyamatokat szemléltető szoftveres eszköz . . . . .	7
2.3.1. Draw.io . . . . .	7
<b>3. Megvalósítás/Specifikáció</b>	<b>9</b>
<b>4. Tervezés</b>	<b>11</b>
4.1. Táblázatok . . . . .	11
4.2. Ábrák . . . . .	11
4.3. További környezetek . . . . .	12
<b>5. JavaScript implementáció</b>	<b>13</b>
<b>6. Folyamat példák</b>	<b>14</b>
6.1. A fejezet célja . . . . .	14
6.2. Tartalom és felépítés . . . . .	14
6.3. Amit csak említés szintjén érdemes szerepeltetni . . . . .	14
<b>7. Összefoglalás</b>	<b>16</b>
<b>Irodalomjegyzék</b>	<b>17</b>

# 1. fejezet

## Bevezetés

A fejezet célja, hogy a feladatkiírásnál kicsit részletesebben bemutassa, hogy miről fog szólni a dolgozat. Érdemes azt részletezni benne, hogy milyen aktuális, érdekes és nehéz probléma megoldására vállalkozik a dolgozat.

Ez egy egy-két oldalas leírás. Nem kellenek bele külön szakaszok (section-ök). Az irodalmi háttérbe, a probléma részleteibe csak a következő fejezetben kell belemenni. Itt az olvasó kedvét kell meghozni a dolgozat többi részéhez.

## 2. fejezet

# Üzleti folyamatok és modellezésük

Dolgozatom első tartalmi fejezetének alapjául a(z) [1] forrás blogbejegyzéseit használtam, ezeket dolgoztam fel és egészítettem ki önálló gondolatokkal.

### 2.1. Mik azok az üzleti folyamatok?

Elsőként nézzük meg, mit nevezhetünk egy folyamatnak. A folyamat szó más-más jelentéssel bír annak függvényében, hogy hol használjuk: például mást jelent a hétköznapi életben, mint az informatikában. A mi esetünkben a folyamat előre meghatározott vagy tetszőleges sorrendben elvégezhető tevékenységek kapcsolatrendszerét jelent.

Tehát gyakorlatilag a folyamat olyan tevékenységek halmaza, amelyek egymással kölcsönhatásba lépnek egy adott cél elérése érdekében. Ezt a jelentését használhatjuk az üzleti folyamatokra is, hiszen ha egy üzleti célt tűzünk ki magunk elé, akkor annak megvalósulása is több tevékenység egymásutániságából, több lépésből fog adódni. Ezeket a lépéseket együttesen nevezzük *üzleti folyamatnak*.

Alapvetően kétféle megközelítése van a lépések végrehajtásának sorrendjének:

- Tevékenységek tetszőleges sorrendben elvégezhetőek.
- Csak egymás utáni, meghatározott sorrendben követhetik egymást.

Az üzleti folyamatok minden vállalat életében napi szinten jelen vannak és egy áttekinthető láncolatot alkotnak a termék előállításához szükséges nyersanyagok beszerzésétől kezdve a munkafolyamatokon keresztül egészen a piacra kerülésig, és a piacon való sikeres vagy sikertelen szereplésig.

Üzleti folyamatokról általában akkor beszélünk, ha egy adott vállalat termékét, szolgáltatását, vagy menedzsmentjét akarjuk elhelyezni a gazdasági életben. Belátható, hogy a termék minősége és mennyisége összefügg a gyártó menedzsmentjének arculatával, piaci ismertségével, és a termék iránti kereslet erejével. Ezeket az összefüggéseket jeleníti meg egy vállalat reklám, PR, és marketing tevékenysége, melyről később lesz szó. Közismert példa, hogy az MGM stúdió világviszonylatban piacvezető termékeket (filmeket) állít elő, de sikere a filmgyártás és forgalmazás költségeinek (reklám, PR, marketing) mértéke azonos vagy a forgalmazás irányában nagyobb mértékű.

Hogy érthetőbb legyen mit is jelent a gyakorlatban az üzleti folyamat, nézzük rá egy egyszerű példát:

Adott egy startup (vagy magyarosabban *korai fázisú vállalkozás*[2], ahol szükségessé válik bizonyos irodaszerek beszerzése. Magát a beszerzés menetét egy kizárólag kötött sorrendben elvégezhető folyamat fogja leírni. Nézzük a folyamat lépéseit:

1. Első lépésben konkrétan meghatározzák, hogy melyek azok az eszközök, amelyekre szükség van, és miből hány darab szükséges. Ezeknek az összeírása az első feladat.
2. Miután összegyűjtötték a megrendelendő tárgyakat, meg kell vizsgálniuk, hogy a vállalkozás mekkora anyagi kerettel rendelkezik, és ebből mennyit tudnak a megrendelésre költeni. Ha rendelkeznek megfelelő pénzüsszeggel, akkor minden rendben van, tovább lehet lépni. Ha viszont nem, akkor valamilyen szempont alapján ki kell húzni a listáról bizonyos termékeket (például a legszükségtelebbeket). Ez utóbbi termékek megrendelése vagy elvetése a következő lépéstől is függ.
3. Ha már rendelkezésükre áll a lista és a pénzügyi fedezet, utána kell nézniük, hogy az adott eszközök hol szerezhetőek be a legalacsonyabb áron (ehhez segítséget tud nyújtani például a <https://www.arukereso.hu/> oldal, ami ár szerinti növekvő sorrendben kilistázza, hogy egy adott termék mely online webáruházakban és milyen áron érhető el). Természetesen nem csak az alacsony ár, hanem más szempont is szóba jöhet (például a garancia időtartalma az adott termékre, a kiszállítás díja, stb.).

Ennél a lépésnél derül ki, hogy pontosan mennyi összeget kell a meglévő pénzügyi keretből az új eszközök megrendelésére fordítani. Lehetséges, hogy marad még pénz egy korábban az anyagi keret szűkösségére való hivatkozás miatt lehúzott termék megvásárlására.

4. Mindezek után elindulhat a megrendelés folyamata. Ez többféleképpen is történhet:
  - Az eszközöket online rendeljük meg. Ennek előnye, hogy egyszerre több helyen is megfigyelhetjük a termékeket, utána járhatunk az azokat árusító weboldalak hitelességének, megbízhatóságának, véleményeket olvashatunk róla, és nem kell személyesen megjelennünk az adott áruházban. Hátránya, hogy időbe telik a termékek kiszállítása, és adott esetben számolnunk kell a szállítási költséggel is. Ez utóbbinak mértéke áruházanként eltérő. Ugyanakkor ma már léteznek olyan webáruházak is, ahol a megrendelt terméket személyesen is át lehet venni egy előre meghatározott üzlethelységben, így azokhoz gyorsabban juthatunk hozzá, viszont ebben az esetben a megrendelőre hárul a szállítási költség.
  - Megtehetjük, hogy nem rendeljük meg előzetesen a termékeket, hanem azok beszerzésére személyesen megyünk be az áruházakba. Ezt akkor célszerű alkalmaznunk, mikor egy nagyobb áruházat fogunk meglátogatni, ahol nagy eséllyel az összes termék rendelkezésünkre fog állni, így azokat egy helyről azonnal meg is tudjuk venni, és elvinni. Nagy előny ebben az esetben (szemben az online rendeléssel), hogy ki is tudjuk próbálni az adott termékeket, hogy hogyan működnek, van-e valamilyen hibájuk.
5. Ezek után, hogy a megrendelt termékek eljutottak az irodába, utolsó lépésként már csak szét kell osztani azokat aszerint, hogy mely terméket ki igényelte.

Ezzel a végére is értünk egy egyszerűbb üzleti folyamatnak.



Azért ezt a fenti példát hoztam szemléltetésékként, mert ez valamilyen formában minden vállalkozásnál jelen van. Ha jobban belegondolunk, láthatjuk, hogy gyakorlatilag szinte minden tevékenységet (így adott termék vagy termékek beszerzését is) folyamatban végzünk. Fontos megjegyezni, hogy ennél a példánál, és az ehhez hasonlóknál a lépéseket csak adott sorrendben hajthatunk végre. Például nem vihetünk el egy terméket az áruházból úgy, hogy csak majd később fizetünk érte.



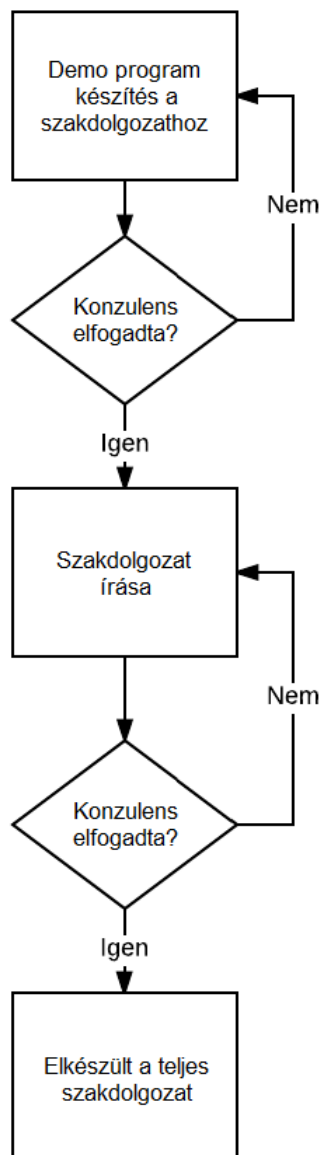
2.1. ábra. Üzleti intelligencia (forrás: piacesprofit.hu)

## 2.2. Üzleti folyamatok modellezése

Az előző részben olvashattuk, hogy nagyon sok mindent folyamatban végzünk. Akár egy legegyszerűbb hétköznapi tevékenységet is. Egy vállalat esetében pedig az ilyen tevékenységekből jóval több van, melyek lényeges szerepet játszanak annak működésében. Emiatt fontos az, hogy a vállalat bizonyos folyamatai rögzítésre kerüljenek. Ezért van szükség a folyamatok *modellezésére*. Ennek egy egyszerű, átlátható formája a folyamatábra használata.

### 2.2.1. Modellezés folyamatábrával

A folyamatábra gyakorlatilag egy olyan eszköz, amely segítségével grafikusán ábrázolhatóak a tevékenységek, folyamatok. A különböző folyamatokat különböző szimbólumokkal jelölhetjük rajta, közöttük sorrendiséget állíthatunk fel meghatározva ezáltal azt, hogy időben melyik esemény melyik után vagy előtt következik be. Hogy érthetőbb legyen, miről is van szó, a 2.2 ábra egy nagyon egyszerű folyamatábrát szemléltet, amin a szakdolgozatom készítésének legfőbb lépései helyezkednek el.



2.2. ábra. Egy egyszerű folyamatábra

A fent látható folyamatábra egyszerűségét az is adja, hogy nem teljes. Nincsen kiindulópontja sem végpontja. Olyan, mintha egy nagyobb folyamatból csak néhány tevékenységet ábrázolnánk. Ahhoz, hogy teljes legyen a folyamatábra, szükség van kezdő- és végállapotra.

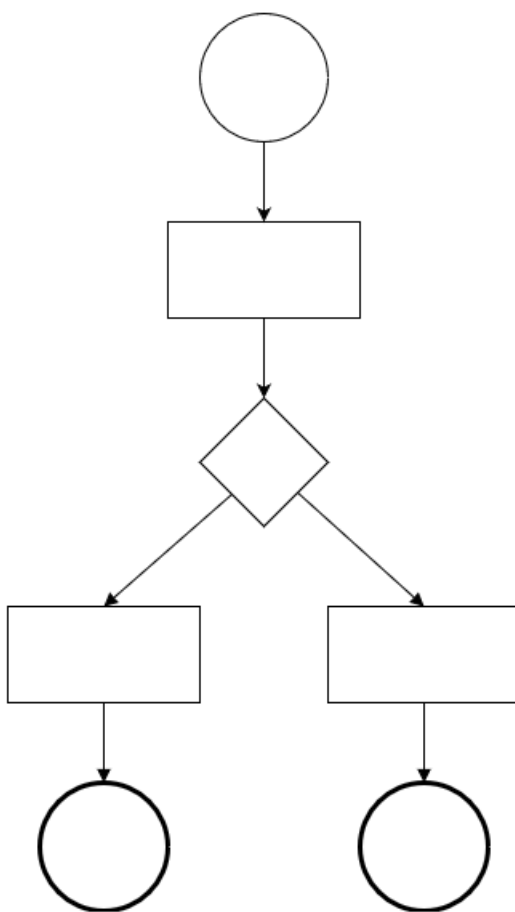
Nézzük tehát, hogy mitől lesz teljes egy folyamatábra, illetve hogy általánosságban milyen részeket tartalmaz. Az alábbi (leggyakoribb, szinte minden folyamatábrán megtalálható) szimbólumok a következő jelentéssel bírnak:

- *Téglalap* jelöli a tevékenységet. Ebből van a legtöbb a folyamatábrán, hiszen ezek szemléltetik magát a folyamatot, az események egymás utáni történését.
- *Rombusz* alak jelenti az átjárókat. Ezek útválasztóként funkcionálnak azáltal, hogy egy feltételt szabnak meg, melynek kiértékelésétől függ, hogy melyik tevékenység fog következni. Általában két másik tevékenység követi, amelyek közül

az lesz a soron következő, amelyikre az útválasztó kiértékelése a megfelelő (igaz vagy hamis) értéket adja.

- *Üres fehér kör vékony fekete körívvel* fejezi ki a "START" állapotot (*kezdőállapot*). Itt kezdődik el a reprezentált folyamat. Legalább egy "START" szimbólumot tartalmaznia kell minden folyamatábrának.
- A folyamat befejezését az *üres fehér kör vastag fekete körívvel* jelöli. Ezt "END" állapotnak (*végállapotnak*) is szokták nevezni. Ebből is legalább egynek lennie kell, hiszen ahogy elkezdődik egy folyamat, úgy az be is fejeződik egy bizonyos idő után. Előfordul olyan folyamatábra, ami több végállapotot tartalmaz, mint kezdőállapotot. Ez nem hiba, hiszen a való életben is gyakran fejeződik be egy elkezdett folyamat többféleképpen.
- A különböző folyamatokat nyilak kötik össze, melyeket *szekvenciafolyamnak* neveznek. Fontos, hogy ezek nem lehetnek csak vonalak nyíl nélkül, hiszen egyértelműen meg kell határozniuk, hogy melyik tevékenységből melyik másik következik.

Példaként tekintsük a 2.3 ábrát, ami egy teljes folyamatábrát mutat, mely a fent említett szimbólumok közül mindegyikből tartalmaz legalább egyet (beleértve a kezdő- és végállapotokat is).



2.3. ábra. Egy üres, de teljes folyamatábra (Forrás: draw.io)

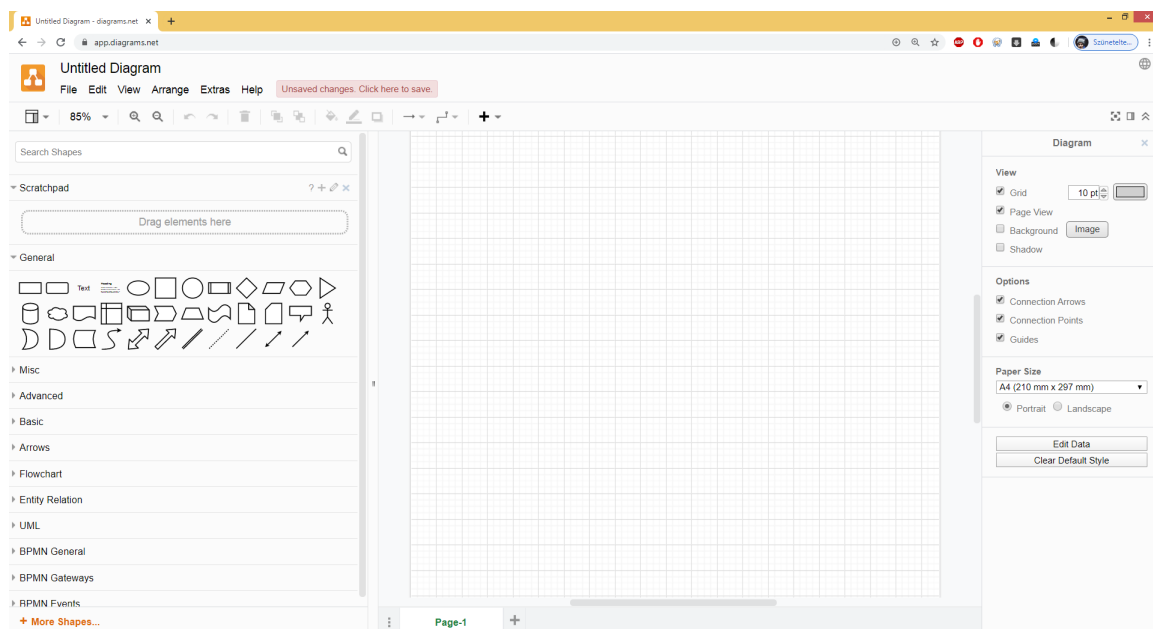
Természetesen léteznek ennél sokkal összetettebb, többféle szimbólumot tartalmazó

folyamatábrák is, azonban már egy ekkora ábrán is meg lehet jeleníteni egyszerűbb üzleti folyamatokat.

## 2.3. Egy folyamatokat szemléltető szoftveres eszköz

### 2.3.1. Draw.io

Folyamatok ábrázolására már nagyon sok kész program létezik. Például a 2.3 ábrát is egy ilyen folyamatábrázoló szoftver segítségével készítettem el, melynek neve *Draw.io*[3]. A Draw.io a legismertebb és legsokoldalúbb ingyenes folyamatábrázoló alkalmazás. Az alkalmazás nem igényel letöltést, a böngésző segítségével el tudjuk érni akár a rövidebb [www.draw.io](http://www.draw.io) URL címen keresztül. További előnye még, hogy regisztrációt sem igényel, hanem azonnal az oldal betöltése után már kezdhethetjük is a folyamataink modellezését.



2.4. ábra. A Draw.io nyitóoldala (Forrás: draw.io)

Láthatjuk, hogy rengeteg lehetőség tárul elénk. A modellezéshez nagyon sokféle szimbólum közül választhatunk számunkra megfelelőt. Természetesen mindegyik más jelentéssel bír, azonban a program teljes egészében a felhasználóra bízta azok használatát: nem határoz meg integritási feltételeket, így a felhasználó szabályoktól függetlenül alakíthat ki számára kedvező ábrákat.

Használata viszonylag egyszerű, felhasználóbarát. A felhasználó kiválaszt egy elemet a bal oldali eszköztárból, és azt kattintással, vagy akár behúzással megjeleníti a képernyő közepén található négyzetrácsos részen. Az így megjelenített szimbólumokat nyomva tartott bal egérgomb mellett az egér mozgásával lehet áthelyezni.

Az általános szimbólumokon kívül (amik a program bal oldali menüjében le vannak nyitva *General* néven) különböző nyilatokat, egyéb formákat használhatunk, UML diagramokat készíthetünk, illetve az üzleti folyamatmodell és jelölés (*Business Process Model and Notation*, BPMN) sajátos alakzatait is igénybe vehetjük. De ha még ez se

lenne elég, akkor lehetőség van az eszköztárat kibővíteni további alakzatok hozzáadásával, amit a bal alsó sarokban található **"More Shapes. . ."** (további alakzatok) gomb megnyomásával.

Elkészített folyamatábráink / diagramjaink mentésére is többféle módot kínál az alkalmazás. Lehetőség van PNG, SVG, HTML, és XML formátumba is menteni a munkánkat, amiket a saját eszközünkre, vagy akár Google Drive-ra, GitHub-ra, és még sokféle felületre azonnal exportálhatunk is.

Összegezve tehát a Draw.io egy rendkívül hasznos és változatos alkalmazás, amely sokféle megvalósítási lehetőséget kínál a felhasználók számára. Egyszerű használatának köszönhetően népszerű a felhasználók körében. Üzleti folyamatok modellezésére kiválóan használható. Egyetlen hátránya lehet az, hogy (mivel böngészőből indítható el) internetkapcsolat szükséges hozzá, de ma már szinte bárhol vagyunk, könnyedén csatlakozni tudunk a világhálóhoz.

## 3. fejezet

# Megvalósítás/Specifikáció

A 3. fejezetben kellene részletesen leírni, hogy milyen feladatokat és hogyan old majd meg az elkészített alkalmazás.

Ez a fejezet mutatja be a megvalósítás lépéseit. Itt lehet az esetlegesen előforduló technikai nehézségeket említeni. Be lehet már mutatni a program elkészült részeit.

Meg lehet mutatni az elkészített programkód érdekesebb részeit. (Az érdekesebb részek bemutatására kellene szorítkozni. Többségében a szöveges leírásnak kellene benne lennie. Abból lehet kiindulni, hogy a forráskód a dolgozathoz elérhető, azt nem kell magába a dolgozatba bemásolni, elegendő csak behivatkozni.)

A dolgozatban szereplő forráskódrészletekhez külön vannak programnyelvenként stílusok. Python esetében például így néz ki egy formázott kódrészlet.

```
import sys

if __name__ == '__main__':
    pass
```

A stílusfájlok a `styles` jegyzékben találhatók. A stílusok között szerepel még C++, Java és Rust stílusfájl. Ezek használatához a `dolgozat.tex` fájl elején `usepackage` paranccsal hozzá kell adni a stílust, majd a stílusfájl nevével megegyező környezetet lehet használni. További példaként C++ forráskód esetében ez így szerepel.

```
#include <iostream>

class Sample : public Object
{
    // An empty class definition
}
```

Stílusfájlokból elegendő csak annyit meghagyni, amennyire a dolgozatban szükség van. Más, C szintaktikájú nyelvekhez (mint például a JavaScript és C#) a Java vagy C++ stílusfájlok átszerkesztésére van szükség. (Elegendő lehet csak a fájlnevet átírni, és a fájlban a környezet nevét.)

Nyers adatok, parancssori kimenetek megjelenítéséhez a `verbatim` környezetet lehet használni.

```
$ some commands with arguments
1 2 3 4 5
$ _
```

---

A kutatás jellegű témáknál ez a fejezet gyakorlatilag kimaradhat. Helyette inkább a fő vizsgálati módszerek, kutatási irányok kaphatnak külön-külön fejezeteket.

## 4. fejezet

# Tervezés

A 4. fejezetben osztálydiagram kellene, illetve az adatok mentése kapcsán érdemes kliens-szerver architektúrák és REST API-s leírást is beletenni.

Itt kezdődik a dolgozat lényegi része, úgy érte, hogy a saját munka bemutatása. Jellemzően ebben szerepelni szoktak blokkdiagramok, a program struktúrájával foglalkozó leírások. Ehhez célszerű UML ábrákat (például osztály- és szekvenciadiagramokat) használni.

Amennyiben a dolgozat inkább kutatás jellegű, úgy itt lehet konkretizálni a kutatási módszertant, a kutatás tervezett lépéseit, az indoklást, hogy mit, miért és miért pont úgy érdemes csinálni, ahogyan az a későbbiekben majd részletezésre kerül.

Ebben a fejezetben az implementáció nem kell, hogy túl nagy szerepet kapjon. Ez még csak a tervezési fázis. (Nyilván ha olyan a téma, hogy magának az implementációnak a módjával foglalkozik, adott formális nyelvet mutat be, úgy a kódpéldákat már innen sem lehet kihagyni.)

### 4.1. Táblázatok

Táblázatokhoz a `table` környezetet ajánlott használni. Erre egy minta a 4.1. táblázat. A hivatkozáshoz az egyedi `label` értéke konvenció szerint `tab:` prefixszel kezdődik.

4.1. táblázat. Minta táblázat. A táblázat felirata a táblázat felett kell legyen!

a	b	c
1	2	3
4	5	6

### 4.2. Ábrák

Ábrákat a `figure` környezettel lehet használni. A használatára egy példa a 4.1. ábrán látható. Az `includegraphics` parancsba Az ábrák felirata az ábra alatt kell legyen. Az ábrák hivatkozásához használt nevet konvenció szerint `fig:-`el célszerű kezdeni.





4.1. ábra. A Miskolci Egyetem címere.

### 4.3. További környezetek

A matematikai témájú dolgozatokban szükség lehet tételek és bizonyításaik megadására. Ehhez szintén vannak készen elérhető környezetek.

**4.1. definíció.** Ez egy definíció

**4.2. lemma.** *Ez egy lemma*

**4.3. tétel.** *Ez egy tétel*

**Bizonyítás.** Ez egy bizonyítás

□

**4.4. következmény.** *Ez egy tétel*

**4.5. megjegyzés.** Ez egy megjegyzés

**4.6. példa.** Ez egy példa

## 5. fejezet

# JavaScript implementáció

A fejezetben be kell mutatni, hogy az elkészült alkalmazás hogyan használható. (Az, hogy hogyan kell, hogy működjön, és hogy hogy lett elkészítve, az előző fejezetekben már megtörtént.)

Jellemzően az alábbi dolgok kerülhetnek ide.

- Tesztfuttatások. Le lehet írni a futási időket, memória és tárigényt.
- Felhasználói kézikönyv jellegű leírás. Kifejezetten a végfelhasználó szempontjából lehet azt bemutatni, hogy mit hogy lehet majd használni.
- Kutatás kapcsán ide főként táblázatok, görbék és egyéb részletes összesítések kerülhetnek.

## 6. fejezet

# Folyamat példák

### 6.1. A fejezet célja

Ez a fejezet még nem a saját eredményekkel foglalkozik, hanem bemutatja, mi a problémakör, milyen módszerekkel, milyen eredményeket sikerült elérni eddig másoknak.

A hivatkozások jelentős része ehhez a fejezethez szokott kötődni. (Egy hivatkozás például így néz ki [4].) Itt lehet bemutatni a hasonló alkalmazásokat.

### 6.2. Tartalom és felépítés

A fejezet tartalma témától függően változhat. Az alábbiakat attól függően különböző arányban tartalmazhatják.

- Irodalomkutatás. Amennyiben a dolgozat egy módszer kidolgozására, kifejlesztésére irányul, akkor itt lehet részletesen végignézni (módszertani vagy időrendi bontásban), hogy az eddigiekben milyen eredmények születtek a témakörben.
- Technológia. Mivel jellemzően kutatásról vagy szoftverfejlesztésről van szó, ezért annak a jellemző elemeit, technikai részleteit itt kell bemutatni. Ez tehát egy módszeres bevezetés ahhoz, hogy ha valaki nem jártas a témakörben, akkor tudja, hogy a dolgozat milyen aktuálisan elérhető eredményeket, eszközöket használt fel.
- Piackutatás. Bizonyos témáknál új termék vagy szolgáltatás kifejlesztése a cél. Ekkor érdemes annak alaposan utánanézni, hogy aktuálisan milyen eszközök érhetők el a piacon. Ez szoftverek esetében a hasonló alkalmazások bemutatását, táblázatos formában történő összehasonlítását jelentheti. Szerepelhetnek képek és észrevételek a viszonyításként bemutatott alkalmazásokhoz.
- Követelmény specifikáció. Külön szakaszban érdemes részletesen kitérni az elkészítendő alkalmazással kapcsolatos követelményekre. Ehhez tartozhatnak forgatókönyvek (*scenario-k*). A szemléletesség kedvéért lehet hozzájuk képernyőkép vázlatokat is készíteni, vagy a használati eseteket más módon szemléltetni.

### 6.3. Amit csak említés szintjén érdemes szerepeltetni

Az olvasóról annyit feltételezhetünk, hogy programozásban valamilyen szinten járatos, és a matematikai alapfogalmakkal sem ebben a dolgozatban kell megismertetni. A spe-

ciális eszközök, programozási nyelvek, matematikai módszerek és jelölések persze jó, hogy ha említésre kerülnek, de nem kell nagyon belemenni a közismertnek tekinthető dolgokba.

## 7. fejezet

# Összefoglalás

Hasonló szerepe van, mint a bevezetésnek. Itt már múltidőben lehet beszélni. A szerző saját meglátása szerint kell összegezni és értékelni a dolgozat fontosabb eredményeit. Meg lehet benne említeni, hogy mi az ami jobban, mi az ami kevésbé jobban sikerült a tervezettnél. El lehet benne mondani, hogy milyen további tervek, fejlesztési lehetőségek vannak még a témával kapcsolatban.

# Irodalomjegyzék

- [1] Forrás: <https://xflower.hu/blog/>, *Az üzleti folyamatokról bővebben* szavakkal kezdődő bejegyzések.
- [2] Forrás: <https://hu.wikipedia.org>.
- [3] Megtekinthető itt: <https://app.diagrams.net/>.
- [4] James H Coombs, Allen H Renear, and Steven J DeRose. Markup systems and the future of scholarly text processing. *Communications of the ACM*, 30(11):933–947, 1987.

# CD Használati útmutató

Ennek a címe lehet például *A mellékelt CD tartalma* vagy *Adathordozó használati útmutató* is.

Ez jellemzően csak egy fél-egy oldalas leírás. Arra szolgál, hogy ha valaki kézhez kapja a szakdolgozathoz tartozó CD-t, akkor tudja, hogy mi hol van rajta. Jellemzően elég csak felsorolni, hogy milyen jegyzékek vannak, és azokban mi található. Az elkészített programok telepítéséhez, futtatásához tartozó instrukciók kerülhetnek ide.

A CD lemezre mindenképpen rá kell tenni

- a dolgozatot egy `dolgozat.pdf` fájl formájában,
- a LaTeX forráskódját a dolgozatnak,
- az elkészített programot, fontosabb futási eredményeket (például ha kép a kimenet),
- egy útmutatót a CD használatához (ami lehet ez a fejezet külön PDF-be vagy Markdown fájlként kimentve).