Általános információk

A diplomaterv szerkezete:

1. Diplomaterv feladatkiírás
2. Címoldal
3. Tartalomjegyzék
4. A diplomatervező nyilatkozata az önálló munkáról és az elektronikus adatok kezeléséről
5. Tartalmi összefoglaló magyarul és angolul
6. Bevezetés: a feladat értelmezése, a tervezés célja, a feladat indokoltsága, a diplomaterv felépítésének rövid összefoglalása
7. A feladatkiírás pontosítása és részletes elemzése
8. Előzmények (irodalomkutatás, hasonló alkotások), az ezekből levonható következtetések
9. A tervezés részletes leírása, a döntési lehetőségek értékelése és a választott megoldások indoklása
10. A megtervezett műszaki alkotás értékelése, kritikai elemzése, továbbfejlesztési lehetőségek
11. Esetleges köszönetnyilvánítások
12. Részletesés pontos irodalomjegyzék
13. Függelék(ek)

Felhasználható a következő oldaltól kezdődő Diplomaterv sablon dokumentum tartalma. Ügyeljen a konzulens nevét és a beadás évét jelölő szövegdobozokra, mert azokra külön ki kell adni a frissítést. A mezők tartalma a sablonban a dokumentum adatlapja alapján automatikusan kerül kitöltésre.

A diplomaterv szabványos méretű A4-es lapokra kerüljön. Az oldalak tükörmargóval készüljenek (mindenhol 2,5 cm, baloldalon 1 cm-es kötéssel). Az alapértelmezett betűkészlet a 12 pontos Times New Roman, másfeles sorközzel.

Minden oldalon – az első négy szerkezeti elem kivételével – szerepelnie kell az oldalszámnak.

A fejezeteket decimális beosztással kell ellátni. Az ábrákat a megfelelő helyre be kell illeszteni, fejezetenként decimális számmal és kifejező címmel kell ellátni. A fejezeteket decimális aláosztással számozzuk, maximálisan 3 aláosztás mélységben (pl. 2.3.4.1.). Az ábrákat, táblázatokat és képleteket célszerű fejezetenként külön számozni (pl. 2.4. ábra, 4.2 táblázat vagy képletnél (3.2)). A fejezetcímeket igazítsuk balra, a normál szövegnél viszont használjunk sorkiegyenlítést. Az ábrákat, táblázatokat és a hozzájuk tartozó címet igazítsuk középre. A cím a jelölt rész alatt helyezkedjen el.

A képeket lehetőleg rajzoló programmal készítsék el, az egyenleteket egyenlet-szerkesztő segítségével írják le.

Az irodalomjegyzék szövegközi hivatkozása történhet a Harvard-rendszerben (a szerző és az évszám megadásával) vagy sorszámozva. A teljes lista névsor szerinti sorrendben a szöveg végén szerepeljen (sorszámozott irodalmi hivatkozások esetén hivatkozási sorrendben). A szakirodalmi források címeit azonban mindig az eredeti nyelven kell megadni, esetleg zárójelben a fordítással. A listában szereplő valamennyi publikációra hivatkozni kell a szövegben. Minden publikáció a szerzők után a következő adatok szerepelnek: folyóirat cikkeknél a pontos cím, a folyóirat címe, évfolyam, szám, oldalszám tól-ig. A folyóirat címeket csak akkor rövidítsük, ha azok nagyon közismertek vagy nagyon hosszúak. Internet hivatkozások megadásakor fontos, hogy az elérési út előtt megadjuk az oldal tulajdonosát és tartalmát (mivel a link egy idő után akár elérhetetlenné is válhat), valamint az elérés időpontját.

Fontos:

* a szakdolgozat készítő/diplomatervező nyilatkozata (a jelen sablonban szereplő szövegtartalommal) kötelező előírás Karunkon, ennek hiányában a szakdolgozat/diplomaterv nem bírálható és nem védhető!
* mind a dolgozat, mind a melléklet maximálisan 15 MB méretű lehet!

Jó munkát, sikeres szakdolgozat készítést, ill. diplomatervezést kívánunk!

FeladatkiÍrás

TODO: pontosan mi idén a policy? Elvileg a tanszékvezető utólag aláírja.

A feladatkiírást a tanszék saját előírása szerint vagy a tanszéki adminisztrációban lehet átvenni, és a tanszéki pecséttel ellátott, a tanszékvezető által aláírt lapot kell belefűzni a leadott munkába, vagy a tanszékvezető által elektronikusan jóváhagyott feladatkiírást kell a Diplomaterv Portálról letölteni és a leadott munkába belefűzni (ezen oldal HELYETT, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatban már nem kell megismételni a feladatkiírást.

C:\Users\szarnyasg\Downloads\bme_logo_nagy.eps

**Budapesti Műszaki és Gazdaságtudományi Egyetem**

Villamosmérnöki és Informatikai Kar

Pogácsasütöde Tanszék

Zöld mesterséges intelligencia

Készítette

Konzulens

2024

Tartalomjegyzék

[Összefoglaló 6](#_Toc396824923)

[Abstract 7](#_Toc396824924)

[1. Bevezetés 8](#_Toc396824925)

[1.1. Formázási tudnivalók 8](#_Toc396824926)

[1.1.1. Címsorok 8](#_Toc396824927)

[1.1.2. Képek 8](#_Toc396824928)

[1.1.3. Táblázatok 8](#_Toc396824929)

[1.1.4. Kódrészletek 8](#_Toc396824930)

[1.1.5. Irodalomjegyzék 9](#_Toc396824931)

[2. Utolsó simítások 10](#_Toc396824932)

[3. Összefoglalás 11](#_Toc396824933)

[Köszönetnyilvánítás 12](#_Toc396824934)

[Ábrák jegyzéke 13](#_Toc396824935)

[Táblázatok jegyzéke 14](#_Toc396824936)

[Irodalomjegyzék 15](#_Toc396824937)

[Függelék 16](#_Toc396824938)

Hallgatói nyilatkozat

Alulírott , szigorló hallgató kijelentem, hogy ezt a szakdolgozatot/diplomatervet (nem kívánt törlendő) meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2024. 05. 16.

Rezeda Kázmér

# Összefoglaló

A szakdolgozat, vagy diplomaterv elkészítése minden egyetemi hallgató életében egy fontos mérföldkő. Lehetőséget ad arra, hogy az egyetemi évei során megtanultakat kamatoztassa és eredményeit szélesebb közönség előtt bemutassa, s mérnöki rátermettségét bizonyítsa. Fontos azonban, hogy a dolgozat elkészítésének folyamata számos csapdát is rejt magában. Rossz időgazdálkodás, hiányos szövegszerkesztési ismeretek, illetve a dolgozat készítéséhez nélkülözhetetlen „műfaji” szabályok ismeretének hiánya könnyen oda vezethetnek, hogy egy egyébként jelentős időbefektetéssel készült kiemelkedő szoftver is csak gyengébb minősítést kapjon a gyenge minőségű dolgozat miatt.

E dokumentum – amellett, hogy egy általános szerkesztési keretet ad a dolgozatodnak – összefoglalja a szakdolgozat/diplomaterv írás írott és íratlan szabályait. Összeszedjük a Word kezelésének legfontosabb részeit (címsorok, ábrák, irodalomjegyzék stb.), a dolgozat felépítésének általános tartalmi és szerkezeti irányelveit. Bár mindenkire igazítható sablon természetesen nem létezik, megadjuk azokat az általános arányokat, oldalszámokat, amelyek betartásával jó eséllyel készíthetsz egy színvonalas dolgozatot. A részletes és pontokba szedett elvárás-lista nem csupán a dolgozat írásakor, de akár más dolgozatok értékelésekor is kiváló támpontként szolgálhat.

Az itt átadott ismeretek és szemléletmód nem csupán az aktuális feladatod leküzdésében segíthet, de hosszútávon is számos praktikus fogással bővítheti a szövegszerkesztési és dokumentumkészítési eszköztáradat.

# Abstract

English translation of the abstract of the thesis work. This summarises the content of the thesis in 0.5–1 pages and is uploaded to the Thesis Work Portal as well.

# Collecting data

For any artificial intelligence model, abundant data is essential. However, one of the most significant challenges in this project has been the lack of publicly available data on public transportation emissions. To overcome this challenge the idea was to use a traffic simulator and configure it in a way that represents real life public transportation.

"Simulation of Urban MObility" (SUMO) is an open source, highly portable, microscopic, and continuous traffic simulation package designed to handle large networks. It allows for intermodal simulation including pedestrians and comes with a wide range of tools for scenario creation. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center.

## Creating a simulation

In SUMO, the OSM Web Wizard is a highly accessible tool for getting started. It allows users to configure a randomized traffic demand and run visual simulations based on an OpenStreetMap excerpt. This approach provides a straightforward way to simulate traffic in real-world locations. For this project, I used the OSM Web Wizard to create a simulation for a smaller area in Budapest.

To achieve this, I navigated to the SUMO tools library in the Windows Command Prompt and executed the following command:

python osmWebWizard.py

In the OSM Web Wizard, I then selected the desired area for simulation. The only modification I made was setting the simulation duration to one day (86400 seconds), the rationale for which will be discussed in the next chapter.

While this method offers ease of use, it is not without its drawbacks. The simulations generated produce numerous warnings, which I chose to overlook for this project because SUMO can resolve them during runtime. Although this is not an ideal solution, it is sufficient for the purposes of maintaining symmetry in both emission and battery simulations. Additionally, this method is impractical for larger areas, as it calculates vehicle numbers based on population density. For a densely populated city like Budapest, this results in an excessive number of vehicles that most computers cannot handle efficiently.

The generated configuration file (osm.sumocfg) can be opened in text editors or in sumo-gui which is basically the same application as sumo, just extended by a graphical user interface. In the next steps, I modified this file to ensure it consistently works with the same imports each time it is run.

## Importing GTFS

GTFS (General Transit Feed Specification) is a standardized format for public transportation schedules and associated geographic information. It enables public transit agencies to publish their data in a format that can be consumed by software applications. GTFS data is typically organized into a collection of text files, each representing a different aspect of the transit system, such as routes, stops, and schedules. In SUMO there are tools, that facilitate the import of schedules (and also routes to a limited degree) into a simulation.

For this project, I utilized the publicly available GTFS schedules for both Volán and BKK vehicles. Initially, I attempted to import the Volán GTFS into SUMO by copying it into the simulation library and executing the recommended commands from the SUMO documentation.

After these attempts proved unsuccessful, I proceeded to work with the BKK GTFS. Following the same steps, I encountered an error due to the absence of a calendar.txt file in the BKK GTFS. This file is conditionally required for defining the regular service patterns for public transit routes. To resolve this, I created an empty calendar.txt file containing only the column names, enabling the import process to proceed.

In the simulation folder, I used the following command:

python "C:/Program Files (x86)/Eclipse/Sumo/tools/import/gtfs/gtfs2pt.py" --gtfs "bkk\_gtfs\_e.zip" --date "20240325" --network "osm.net.xml.gz"

This command runs the gtfs2pt.py Python script to import a single day's schedule into the network. I specified a weekday (March 25, 2024) when public transit operates as usual. Since this method only allows for importing of entire days, I created a day-long traffic simulation in the previous step to accommodate this requirement.

This process generated files that needed to be added to the configuration file, replacing those created by the OSM Web Wizard. I updated the configuration file as follows:

<additional-files value="vtypes.xml,gtfs\_pt\_stops.add.xml,gtfs\_pt\_vehicles.add.xml"/>

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

After completing this step, the stops and added vehicles can be viewed by running the simulation in sumo-gui.

## Importing Elevation Data

kep a szines utakrol

In SUMO, netconvert is a tool used for importing and converting road networks. Elevation data can be imported from a grayscale height map using the netconvert option --heightmap.geotiff. A grayscale height map is a digital image used in computer graphics and geographic information systems (GIS) to represent the elevation of terrain or other surfaces. Each pixel in the image corresponds to a specific geographic location, with the intensity of the grayscale value indicating the height at that point: darker pixels represent lower elevations, while lighter pixels represent higher elevations.

I downloaded a .tif file of Budapest from the CGIAR Consortium for Spatial Information (CGIAR-CSI) website. To import this file, I used the following command in the simulation folder:

netconvert -s osm.net.xml.gz --heightmap.geotiff srtm\_40\_03.tif -o height.net.xml

This command generated a new file, height.net.xml, which includes z coordinates for elevation. The final step was to update the configuration file to use this new network file:

<net-file value="height.net.xml"/>

Elevation data can be visualized in sumo-gui by colouring streets by height at start.

A képen szöveg, képernyőkép, diagram, térkép látható

Automatikusan generált leírás

## Output files

SUMO allows to generate many different measurements. Per default, all are disabled, and In SUMO, various measurements can be generated, but by default, they are all disabled and need to be activated individually. For this project, the key outputs required are emissions, battery usage, and trip information. Emission output provides data on the emissions of all vehicles at each simulation step, battery output tracks battery state for electric vehicles, and trip information aggregates details about each vehicle's journey.

To enable these outputs, I added the following lines to the configuration file:

<output>

<emission-output value="emission.out.xml"/>

<tripinfo-output value="tripinfo.xml"/>

<battery-output value="battery.out.xml"/>

</output>

However, merely adding these lines generates an output file without any data. To collect data, each vehicle must be assigned a device. This can be done explicitly by specifying the vehicle IDs or by randomly assigning a certain percentage of vehicles a device. I opted to explicitly define devices for each vehicle, ensuring that all vehicles have all necessary devices. This was achieved by adding the following lines to the configuration file:

<emissions>

<device.emissions.explicit value="<!--put vehicle ids here-->"/>

</emissions>

<tripinfo>

<device.tripinfo.explicit value="<!--put vehicle ids here-->"/>

</tripinfo>

<battery>

<device.battery.explicit value="<!--put vehicle ids here-->"/>

</battery>

In the placeholder <!--put vehicle ids here-->, the specific vehicle IDs will be listed.

## Getting vehicle IDs

To obtain the vehicle IDs, I utilized the gtfs\_pt\_vehicles.add.xml file, which was created earlier by importing the GTFS data. This file lists all added vehicles in XML format along with their properties, including their IDs. I developed a Python script to read this file, extract the IDs from all vehicle tags, and compile them into a list. Once the list was complete, the script wrote the list elements, separated by commas, into an output file named vehicle\_ids.txt.

From this output file, I randomly selected IDs and pasted them into the configuration file in place of every placeholder. It is crucial for this step that every vehicle is assigned all necessary devices.

<emissions>

<device.emissions.explicit value="C775912413.0, C77776334.0, C763111293.0, C577041748.0, C77704682.0, C76412107.0, C76311425.0, C7778220.0, C731633508.0, C7612853.0"/>

</emissions>

<tripinfo>

<device.tripinfo.explicit value="C775912413.0, C77776334.0, C763111293.0, C577041748.0, C77704682.0, C76412107.0, C76311425.0, C7778220.0, C731633508.0, C7612853.0"/>

</tripinfo>

<battery>

<device.battery.explicit value="C775912413.0, C77776334.0, C763111293.0, C577041748.0, C77704682.0, C76412107.0, C76311425.0, C7778220.0, C731633508.0, C7612853.0"/>

</battery>

## Defining new vehicle types

While analyzing the output data, I noticed that the energy consumption of electric buses was significantly lower than real-life values. This discrepancy arose because the default weight for an electric vehicle using the battery device is set to 1830 kg, which corresponds to a standard car. To address this, I defined a custom vehicle type, named ElectricBus. Using a SUMO example definition of a city bus as a base, I adjusted the vehicle mass from 10,000 kg to 18,000 kg to reflect the average mass of a bus in Budapest.

Additionally, I assigned an emission class to the vehicle type, which all buses have by default. This emission class uses the HBEFA4 model, which is free to use but does not account for elevation data. I selected the class UBus\_Std\_gt15-18t\_Euro-VI\_A-C, as it represents a standard 15-to-18-ton bus and has comprehensive fleet data.

Changing the emission class affects the emission output; however, for electric vehicles, I only need this output to track the bus's position at each simulation step. Therefore, I did not define an emission class for electric buses.

After these steps, the definition of vTypes in the vtypes.xml file is as follows:

<vType id="bus" vClass="bus" emissionClass="HBEFA4/UBus\_Std\_gt15-18t\_Euro-VI\_A-C"/>

<vType id="light\_rail" vClass="rail\_urban"/>

<vType id="tram" vClass="tram"/>

<vType id="ElectricBus" accel="1.0" decel="1.0" length="12" maxSpeed="100.0" sigma="0.0" minGap="2.5" color="1,1,1">

<param key="maximumBatteryCapacity" value="3000"/>

<param key="maximumPower" value="1000"/>

<param key="vehicleMass" value="18000"/>

<param key="frontSurfaceArea" value="5"/>

<param key="airDragCoefficient" value="0.6"/>

<param key="rotatingMass" value="100"/>

<param key="radialDragCoefficient" value="0.5"/>

<param key="rollDragCoefficient" value="0.01"/>

<param key="constantPowerIntake" value="100"/>

<param key="propulsionEfficiency" value="0.9"/>

<param key="recuperationEfficiency" value="0.9"/>

<param key="stoppingThreshold" value="0.1"/>

</vType>

At this stage of the project, I created a copy of the simulation folder and established a naming convention. For simulations involving diesel buses, I appended the suffix "\_e" to the folder name to indicate that the emission output is the primary focus. For simulations involving electric buses, I appended the suffix "\_b" to the folder name to indicate that the battery output is the primary focus.

In the electric bus simulation, I modified the gtfs\_pt\_vehicles.add.xml file by replacing the vehicle type of each bus with the previously defined ElectricBus type. I accomplished this by using the find and replace feature in Notepad to change type="bus" to type="ElectricBus".

For efficiency purposes, I also removed the battery output and battery devices in the diesel bus simulation.

# Bevezetés

Jelen sablon célja, hogy a Mérnök informatikus BSc/MSc, a villamosmérnök BSc/MSc, az egészségügyi mérnök MSc, az gazdaságinformatikus MSc képzések szakdolgozat és diplomaterv munkáihoz sablont nyújtson. A dokumentumot korábbi sablonokból kiindulva Kővári Bence és Szárnyas Gábor állította össze.

A dokumentum a szakdolgozat/diplomaterv szerkesztési elveit követve, bemutatja, hogyan érdemes felépíteni és kidolgozni egy ilyen dolgozatot. Fontos, hogy az itt leírtak nem képviselnek hivatalos egyetemi, sem tanszéki álláspontot, céljuk csupán, hogy mankóként szolgáljanak a nagy munkához.

Az útmutató két nagy egységre bomlik, a 2. fejezet a dolgozattal szemben támasztott tartalmi, míg a 3. fejezet a formai elvárásokat foglalja össze. A leírás feltételezi a magyar nyelvű Word használatát, de természetesen bármilyen más szövegszerkesztő is használható a dolgozat elkészítéséhez.

## Frissítsd a dokumentumot

Ha ezt a dokumentumot nem

* a <http://pub.ext.aut.bme.hu/dip> vagy
* a <https://github.com/FTSRG/thesis-template-word>

címről töltötted le, elképzelhető, hogy nem a legfrissebb változat van nálad. Mielőtt továbbmész, érdemes letöltened a legfrissebb verziót.

## Szakdolgozat vagy diplomaterv

Bizonyára észrevetted, hogy oktatóid és társaid felváltva használják a szakdolgozat és a diplomaterv kifejezéseket. Ennek magyarázata egyszerű. A BSc képzés végén leadandó dokumentumot szakdolgozatnak, az MSc képzés végén leadandót diplomatervnek hívják. Míg a szakdolgozat elkészítésére egy, addig a diplomaterv elkészítésére két félév áll rendelkezésedre, ezért a diplomaterv általában mélyebb/átfogóbb, és kb. 30%-kal hosszabb mű, de ezt leszámítva érdemi különbség a kettő között nincs.

## Témaválasztás

A szakdolgozat/diplomaterv alapvetően arról szól, hogy választasz magadnak egy informatikai kihívást (pl. egy szoftver megtervezését és megvalósítását), ezzel konzulensed segítségével megküzdesz, és az eredményeidet, illetve az azokhoz vezető folyamatot leírod egy 50-70 oldalas dokumentumba.

A témád lehet olyan kihívás is, amit úttörőnek gondolsz, de akár egy rendes „iparos munka” is, amikor is megvalósítasz egy 146. könyvtári nyilvántartó rendszert. Az úttörő, újszerű munka **nem elvárás**, bármit is választasz témának, a te feladatod csak annyi, hogy a tanulmányaid során elsajátított mérnöki eszköztáraddal (illetve szükség esetén annak bővítésével) elemezd a problémafelvetést és adj rá egy elfogadható megoldást. Természetesen fontos, hogy a téma téged is motiváljon, s azt tapasztaltuk, hogy erre az újszerű technológiák alkalmazása, illetve megoldások készítése általában jobban alkalmas.

# A dolgozat szerkezete

Nagyon nehéz általános szabályokat felállítani, vagy kőbe vésett arányokat, oldalszámokat mondani, hiszen minden munka egy kicsit más. Itt mégis megpróbáljuk felvázolni nektek egy általános diplomaterv/szakdolgozat szerkezetét, amitől természetesen el lehet térni, amennyiben a téma ezt megkívánja.

## Fejezetek

A dolgozat számos fejezetből épül fel. Fontos látni, hogy ezeket egyáltalán nem szükséges sorrendben feltölteni. A bevált gyakorlat az, hogy e sablonból kiindulva kitörlöd az ismertető részeket, címsor 1-2-3-mal összerakod a főbb fejezeteket, amikről írni akarsz, kb. olyan terjedelemben, hogy a tartalomjegyzék 1-1,5 oldal hosszú legyen, majd hangulattól és az aktuális munkádtól függően tetszőleges sorrendben elkezded őket tartalommal feltölteni. De nézzük meg, mik is a nélkülözhetetlen szerkezeti elemek.

### Feladatkiírás

Nem sorszámozott oldal, megelőzi még a címet is a dolgozatban. Az elektronikusan beadott változatban ez az oldal kihagyható. A **nyomtatott változatban** ennek az oldalnak a helyére a tanszéki ügyintézőtől kapott**, tanszékvezető által aláírt feladatkiírást** kell befűzni.

### Címoldal

A tanszék, a munka címe, a neved a konzulensed neve és a védés éve automatikus generálódnak a dokumentum metaadataiból. Ezt a **Fájl** menü alatt tudod szerkeszteni, Word 2016-ban a **Minden tulajdonság szerkesztése** gombra kattintva.

### Tartalomjegyzék

A tartalomjegyzéket automatikusan generáld a címsorokból. A tartalomjegyzékben elég az első 3 címsor szintet kezelni (ez a sablon eleve így van beállítva). Figyelj rá, hogy a tartalomjegyzéket (mint minden mezőt) kézzel kell frissíteni, ha változott az elrendezés, vagy új címsorok kerültek a dokumentumba.

### Nyilatkozat

A törlendő részt húzd ki. A nevedet és a dátumot tartalmazó mezők automatikusan generálódnak, nyomtatás előtt frissítsd ezeket. Nyomtatás után ne felejtsd el aláírni.

### Tartalmi összefoglaló

A dolgozat következő eleme egy legfeljebb 1 oldalas magyar nyelvű „Összefoglaló”, illetve az ennek angol fordítását tartalmazó „Abstract”.

A tartalmi összefoglaló készítésének szigorú műfaji szabályai vannak. Bár sokban hasonlít a könyvek hátoldalán látható ajánlókhoz (az ún. blurbökhöz), valójában sokkal több, mint kedvcsináló. Ez alapján az olvasónak el kell tudnia majd dönteni, hogy érdemes-e elolvasnia a dolgozatot, tartalmaz-e számára értékes információkat. Ennek megfelelően nem csak a kedvcsinálás a cél, hanem az is, hogy az eredmények és a kontextus is bemutatásra kerüljön. A tökéletesen összeszedett ajánlásokat [1] foglalja össze. Referenciaként használhatod ennek az útmutatónak az absztraktját is.

Tipp: a tartalmi összefoglalót célszerű a dolgozat végén megírni, hiszen ekkorra áll össze az összes információ, ami szükséges hozzá.

### Bevezetés

Itt kezdődik a dolgozat érdemi része. A bevezetés hossza szakdolgozatban 3-5, diplomatervben 3-7 oldal. Célja a feladat értelmezése, a motiváció leírása, a kontextus megteremtése. Ami a stílust illet, képzeld azt, hogy egy ELTE bölcsésznek próbálod elmagyarázni, mit csináltál és miért. Indíts messziről, pl. beszélj a széles értelemben vett területed (web, mobil stb.) fontosságáról, majd fokozatosan közelíts rá arra a szűkebb problémára amit látsz, s amit megpróbálsz megoldani a dolgozatodban.

Tipikus hiba szokott lenni, hogy motivációként azt írod le, hogy „a konzulensemtől ezt a feladatot kaptam”. Ez nagyon bénán hangzik. Próbáld inkább egy tágabb perspektívából nézni a dolgot és ragadd meg, mitől teszi a te megoldásod/munkád jobbá a világot és erre fókuszálj a bevezetésben.

Kitérhetsz a kapcsolódó technológiák megemlítésére is, de vigyázz, ne ess a 2.2.4. fejezetben bemutatott csapdákba.

A bevezetés praktikus zárása egy olyan alfejezet, ami a dolgozat további szerkezetét ismerteti.

### Irodalomkutatás, technológiák, hasonló alkotások bemutatása

A fejezet terjedelme szakdolgozat esetében kb. 5-10, diplomaterv esetében 7-13 oldal. Ennek a fejezetnek a lénye, hogy bemutassa azt az alapot, amire a munkádat építetted. Ha tudományosabb munkát végzel, itt bemutathatod a szakirodalmat, mások munkáját, eredményeit, amikhez képest a sajátodat majd később meg tudod határozni. Egy tipikus szoftverfejlesztés esetében pedig itt van lehetőséged arra, hogy röviden bemutasd azokat a technológiákat, amikre a munkád során építettél. Fontos, hogy mindig, a témád szempontjából releváns részekről írj, ha például egy webshopot készítesz .NET-ben, teljesen felesleges a reflexióról írnod, annál fontosabbak lehet viszont megemlíteni pl. a jQuery technológiát.

Feltétlenül érdemes kitérned arra, miként álltak a hasonló problémákhoz mások, s bemutatni 2-3 a területen népszerű megoldást, kiemelve azokat az ötleteket, amiket esetleg átvettél belőlük.

### A feladatkiírás pontosítása és részletes elemzése

Tipikus fejezetcím lehet az „Architektúra”, vagy „Tervezés”, a terjedelme 3-5 oldal szakdolgozat és diplomaterv esetén is. A feladat itt már egyáltalán nem mesélgetés, hanem egy objektív elemzésbemutatása arról, hogy mi is a konkrét feladat, amit meg kell valósítani, milyen funkcionális és nem funkcionális követelmények vannak, hogy néz ki a rendszer architektúrája amit terveztél, milyen használati esetek (nem use-case-ek, lásd. 3.6.3.) vannak, s esetleg itt kaphat helyet az adatbázis séma tervezete is. Ez a fejezet viszonylag közel van a „nagykönyv” szerinti követelményspecifikációhoz.

### Önálló munka bemutatása

Ez a blokk egy, vagy több nagyobb fejezetet tartalmaz, mely bemutatja az elkészült megoldás műszaki felépítését, kiemelve az érdekesebb/bonyolultabb megoldásokat és a nem egyértelmű technológiai döntések indoklásait. Terjedelme szakdolgozatban kb. 30, diplomatervben kb. 40 oldal.

Nagyon fontos, hogy szemben az előző blokkal itt nem cél, hogy egy száraz dokumentációt készítsél. Tipikus hiba tud lenni, hogy minden apró részletet és függvényt be akarsz mutatni a programodból. Ennek az eredménye általában csak számos hosszú száraz felsorolás, amit minden jóérzésű bíráló már a második elem után átugrik…

Képzeld el, hogy egy mérnök informatikus barátodnak kell bemutatnod a szoftvert amit készítettél. Tervezz meg egy gondolati szálat, ami mentén be akarod mutatni neki. Például kezdheted az adatbázistól, melyen keresztül bemutatod az adatmodellt, majd az erre épülő üzleti logikai réteget, végül pedig a felhasználói felületet. De haladhatsz vízszintesen is, modulról, modulra a rendszer bemutatása során. A lényeg, hogy legyen valamilyen rendező elv, s ezen belül se ***mindent*** próbálj bemutatni, hanem koncentrálj az érdekesebb, nem triviálisabb megoldásokra.

Rossz példa:

*A WordProcessor osztály a következő funkciókat valósítja meg:*

* ***AddWord****: felvesz egy új szót*
* ***RemoveWord****: törli a megadott szót*
* ***ChangeWord****: megváltoztatja a kijelölt szót*
* ***CreateDocument****: létrehoz egy új dokumentumot.*

Ugyanez helyesen:

*Megoldásomban a szavak kezelését a* WordProcessor *osztály végzi. Első használatkor a* CreateDocument *függvényével tudunk egy új dokumentumot létrehozni, majd különböző függvényekkel manipulálni annak tartalmát.*

A fenti példából az is látszik, hogy a programozási elemek (osztálynevek, függvények) elkülönítését hogyan segítheti egy megkülönböztető betűtípus/formázás (következetes!) használata.

### Önálló munka értékelése, mérések, eredmények bemutatása

E fejezet szakdolgozatban 4-5, diplomamunkában 6-7 oldal lehet. Célja, hogy összegezve bemutassa az eredményeidet. Például ha valamilyen algoritmust fejlesztettél, itt mutathatod be mérésekkel, diagramokkal, hogy milyen teljesítményt produkál a különböző megvalósításaihoz, vagy éppen mások implementációihoz képest. Ha egy weboldalt, vagy mobil alkalmazást készítettél, és készültek hozzá automatizált felülettesztek, vagy egységtesztek, akkor azokat itt bemutathatod. Itt írhatsz a szoftver utóéletéről, pl. kikerült-e a Store-ba, elkezdték-e élesben használni, mik vele a tapasztalatok? Előfordulhat az is, hogy nincs külön mondanivalód ezekről, vagy ezeket logikusabb elmondani az előző fejezetben. Ilyen esetekben ez és az előző fejezet összevonható és összemosható egymással.

### Összefoglaló

Dolgozattípustól függetlenül 1 oldalban foglald össze az eredményeidet. E/1-ben és múlt időben. Megterveztem, megvalósítottam, eldöntöttem, leteszteltem… stb.

Itt kaphatnak helyet további max. 1 oldalban a továbbfejlesztési lehetőségek. Milyen hiányosságait látod a rendszernek, illetve milyen lehetőségeket látsz a továbbfejlesztésére?

### Köszönetnyilvánítások

Őszintén szólva, a legtöbb esetben felesleges blokk, pusztán a konzulensednek megköszönni a segítségét nem szükséges. Neki az a dolga, hogy segítsen. Akkor érdemes ezzel foglalkozni, ha például valamilyen harmadik fél (külső cég, külső konzulens, családod stb.) is említésre méltóan segítette a munkádat és úgy érzed, fontos lenne ebben a formában is kinyilvánítani a háládat.

### Részletes és pontos irodalomjegyzék

Ezt a Word generálja neked. Részletek a 3.10. fejezetben.

### Ábrajegyzék, táblázatjegyzék

Hacsak nem állsz szánalmasan rosszul az oldalszámmal, akkor ilyenekre nincs szükséged.

### Függelék

A függelékek a törzstartalmon kívüli kiegészítések a dolgozathoz. Ide kerülhetnek a nagyobb ábrák, hosszabb példakódok, vagy részletes algoritmus bemutatások, amelyek csak lazábban kapcsolódnak a témához, ezért a dolgozatod törzsében nem mutattad be részletesen. A függelék nem kötelező tartalmi elem, sok esetben teljesen elhagyható.

## Egyéb tartalmi elemek

### Stílus

A szakdolgozat/diplomaterv műfaját tekintve félúton van egy szoftverspecifikáció és egy irodalmi esszé között. Fontos, hogy a leírtak (a nyilvánvalóan szubjektív részeket leszámítva) általában objektív elemzés eredményei legyenek, mérnöki szemléletmódot tükrözzenek. Ugyanakkor kerüld a szoftverspecifikációk száraz stílusát, az értelmetlen felsorolásokat, s próbáld inkább (az objektivitás megtartása mellett) elbeszélő stílusban bemutatni a munkádat.

### E/1

A bíráló számára lényeges, hogy határozottan szét tudja választani az önálló munkádat mások munkájától. Ezért rendkívül fontos, hogy következetesen és sűrűn alkalmazd az egyes szám, első személyű mondatokat és kerüld a passzív mondatok használatát. Ne azt írd, hogy „Ez itt az osztálydiagram”, mivel ez simán jelentheti azt is, hogy az ábrán egy osztálydiagramot látunk, amit a netről letöltöttél, hanem írd azt, hogy „A 4. ábrán bemutat**om** az osztálydiagramot, amit az alkalmazás**om**hoz tervezt**em**”

### Rövidítések

A dolgozatodban alapszabály, hogy **minden** rövidítést legalább az első használatkor ki kell fejteni, és vagy röviden elmagyarázni mit jelent, vagy egy referenciát beszúrni, mely a magyarázatot tartalmazza. Ezt követően a rövidítés már szabadon használható.

Nézd meg a lenti példát! Figyeld meg azt is, hogy forrásként magát a szabványt, és nem a Wikipedia oldalt adtam meg.

Rossz: A weboldalt HTML-ben írtam le.

Jó: A weboldal leírására HyperText Markup Language-t (HTML) használtam [2].

### Technológia megválasztása

Számos esetben előfordul, hogy egy-egy probléma megoldására több különböző technológia áll rendelkezésedre. Amennyiben a feladatkiírásod eleve kikötötte valamelyik technológia (pl. NET, Java stb.) használatát, akkor semmilyen magyarázattal nem tartozol emiatt, használod azt, ami a feladatod volt.

Ha egyébként a feladatkiírás nem tett a technológiai választásra utalást, akkor sincs gond, amennyiben valamely mainstream technológiára esik a választásod, valószínűleg nincs szükség magyarázkodásra (a .NET vs. Java példa ebbe a kategóriába esik).

Amennyiben egy feladat megvalósítására többféle, különböző előnyökkel és hátrányokkal járó technológia áll rendelkezésedre (pl. WinForms vs. WPF) és nem a nyilvánvalóan korszerűbbet választod (pl. WinFormsban akarsz megoldani egy feladatot) akkor mindenképpen szükséges, hogy ezt indokold is (pl. a banki környezetben, ahol dolgozom, ez jelenleg a támogatott fejlesztési nyelv).

Mindenképp kerüld viszont el az amatőr magyarázkodást és a szubjektív érvelést: „Azért választottam az iOS platformot, mert sokkal jobbnak tartom az Androidnál” – az ilyen mondatokat előszeretettel szedik cafatokra a bírálók. Inkább kezeld tényként a technológia választást: „Megoldásomat iOS platformon valósítottam meg”.

# Formázási tudnivalók

Ez a fejezet összeszedi azokat a nélkülözhetetlen elemeket, amelyeket a dolgozat készítése során használnod kell a Word eszköztárából. Feltétlenül olvasd egyszer végig, hátha akad köztük újdonság.

## Általános tudnivalók

A diplomaterv szabványos méretű A4-es lapokra kerüljön. Az alapértelmezett betűkészlet a 12 pontos Times New Roman. Ennek használata nem kötelező, de mindenképpen javasolt talpas (serif) betűtípus használata. A dolgozatot 1,15-ös sorközzel célszerű készíteni. A másfeles sorköz még elfogadható, ennél nagyobb használata azonban tilos.

### Margók

A margók mérete mindenhol 2,5 cm. A kötésmargót az **Oldalbeállítás** | **Kötésmargó** menü alatt tudjuk beállítani. Egyoldalas nyomtatás esetén állítsunk be a baloldalon 1 cm-es kötést. Amennyiben a dolgozat kétoldalas nyomtatással készül, a **Több oldal** beállításnál válasszuk a **Margók tükrözése** opciót.

### Számozás

Minden oldalon – az első négy szerkezeti elem kivételével – szerepelnie kell az oldalszámnak. A fejezeteket decimális beosztással kell ellátni. Az ábrákat a megfelelő helyre be kell illeszteni, fejezetenként decimális számmal és kifejező címmel kell ellátni.

### Irodalomjegyzék

Az irodalomjegyzék szövegközi hivatkozása történhet a Harvard-rendszerben (a szerző és az évszám megadásával) vagy sorszámozva. A teljes lista névsor szerinti sorrendben a szöveg végén szerepeljen (sorszámozott irodalmi hivatkozások esetén hivatkozási sorrendben). A szakirodalmi források címeit azonban mindig az eredeti nyelven kell megadni, esetleg zárójelben a fordítással. A listában szereplő valamennyi publikációra hivatkozni kell a szövegben.

Ez a sablon úgy van kialakítva, hogy a fenti előírásokat eleve kikényszerítse, így általában külön nem kell foglalkoznod velük.

## Stílusok

A Word szövegek egységességét stílusok segítéségével lehet a legegyszerűbben garantálni. Azt javaslom, rögzítsd a **Stílusok** eszköztárt jobb oldalra, így folyamatosan nyomon követheted, hogy milyen stílussal dolgozol, illetve a stílusok közti váltást is sokszor kényelmesebb itt kezelni, mint a fenti eszköztáron.

A dokumentum folyószövegéhez használjuk a **Normál** (angol Word esetén Normal) stílust.



1. ábra: Stílus eszköztár rögzítése

## Címsorok

A fejezetcímek esetén a **Címsor 1–4** (Heading 1–4) stílusokat használjuk. Címsor 4-nél mélyebb fejezetszintre egy ilyen terjedelmű munkában ritkán van szükség, ha ez mégis felmerülne, sokszor inkább a fejezetszerkezetet érdemes átgondolni újból

Tipp: a fejezetcímek aktuális szintjét gyorsan módosíthatod az Alt + Shift + [bal/jobb] billentyűkombinációkkal.

## Másolás, beillesztés

A copy-paste a szép formázás legnagyobb ellensége. Amennyiben a dolgozatodon belül helyezel át részleteket, és a stílusokat szépen következetesen használtad, nem nagyon ronthatsz el semmit. Ha viszont külső forrásból másolsz be szöveget, érdemes minden esetben a beillesztést formázások nélkül végezni. Ezt az opciót a Word mindig felkínálja, a beillesztést követően.



2. ábra: Beillesztés formázás nélkül

Amennyiben olyan szövegrészt mozgatsz kivágás/beillesztés párossal, melyben automatikus sorszámozás volt (irodalomhivatkozás, ábra, táblázat), akkor figyelj rá, hogy a sorszámok nem frissülnek automatikusan.

## Mezőfrissítés

A dolgozatodban számos automatikusan kalkulált mező szerepel (ábra sorszámok, irodalomhivatkozások, tartalomjegyzék stb. Ezeket a Word nem tartja automatikusan karban. A mezők értékét érdemes rendszeresen (de a beadás/nyomtatás/PDF generálás előtt feltétlenül) frissíteni. Ehhez jelöld ki a teljes dokumentumot (Ctrl + A), majd az előugró menüből válaszd a „mezőfrissítés” opciót.

## Helyesírás

A rossz helyesírásra nincs mentség. E fejezetben összeszedem a leggyakrabban látott hibákat, amiknek elkerülésére érdemes odafigyelni. Ettől függetlenül melegen ajánlom, hogy a kész dolgozatod olvastasd át egy barátoddal/családtagoddal, hogy az apróbb, megbúvó hibákat is kiszűrd.

### Elgépelések

Ez mindenkivel megesik. Egy jó helyesírás ellenőrző az elgépelések nagy részét megfogja.

### Egyeztetés hiánya

Az elírások egyik leggyakoribb formája az egyes szám/többes szám egyeztetésének hiánya mondatrészek között, mint például itt: „Petike és a barátnője elment**ek** a boltba és hoz**ott** egy kiló kenyeret”. Ezek a mondatok főleg az utólagos átfogalmazások, belejavítások során keletkeznek, legjobb védelem ellenük az utólagos átolvasás.

### Külföldi szavak, kifejezések

Az idegen szavakkal csak a baj van, nehéz őket leírni, ragozni, kiolvasni, vagyis úgy általában használni. Az általános jó tanácsom, hogy amennyiben csak lehetséges, akkor **magyar vagy magyarosított írásmódú szakkifejezéseket használj**, és a könnyebb olvashatóság érdekében mindig **kerüld az idegen szavak ragozását**.

Ez sokszor nem intuitív szabály, hiszen a beszélt informatikai szlengben előszeretettel használunk olyan fordulatokat, mint „property-vel”, „compile-olom” stb., ezek helyett a „tulajdonsággal”, illetve a „lefordítom” százszor jobban néznek ki papíron, könnyítik az olvasást és csökkentik a hibázás esélyét. Néha (például a tulajdonneveknél) nehezen elkerülhető a ragozás, de kis leleményességgel, ügyes fogalmazással itt is megoldható a probléma. Pl. „Apache-csal” helyett írjuk azt, hogy „Apache webszerverrel”.

Néhány tipp a szoftverfejlesztőknek: property = tulajdonság, event = esemény, method = metódus/függvény, debug = hibakeresés, file = fájl.

Ha mégis belevágsz az angol kifejezések használatába, lelked rajta, de kérlek, legalább ezt a minimális szabályhalmazt olvasd át előtte [3].

### Stb.

Az „stb.” rövidítés azt jelenti „és a többi”, mivel „és” előtt felsorolásoknál nem teszünk veszőt, az „stb.” elé sem teszünk soha. Pl.: asztal, szék, lámpa stb.

### Helyesírás-ellenőrző

Személyes ízlés kérdése, hogy milyen eszközzel (LaTeX/Word), illetve ezen belül milyen nyelvű változattal dolgozol, ugyanakkor azt **meg kell oldanod, hogy legyen mellé magyar nyelvű helyesírás-ellenőrződ**. E nélkül dokumentumot szerkeszteni olyan, mint papíron programozni.

Figyelj rá, hogy a Word megengedi, hogy többféle nyelv is legyen egy dokumentumban, illetve a kívülről beillesztett szövegek nyelvét néha a forrásoldal/forrásdokumentum határozza meg. Érdemes ezekben az esetekben kikényszeríteni, hogy a teljes szakasz/dokumentum nyelve magyar legyen. Érdemes néha tesztelni a helyesírás ellenőrzőt. Ha pl. beírod, hogy  és nem húzza alá pirossal, akkor valószínűleg valami nem stimmel a beállításaiddal.



3. ábra: Ellenőrzés nyelvének megadása

## Képek

A dolgozatodban valószínűleg számos ábrára lesz szükséged, ezek használatánál azonban érdemes pár dologra odafigyelned.

### Beszúrás, formázás

A képhez használd a **Kép** stílust.

Képaláírást a képen jobb gombbal kattintva a Képaláírás beszúrása… opcióval adhatod hozzá, így az automatikusan **Képaláírás** (Caption) stílusú lesz és kezeli a sorszámozást is.



4. ábra: Példa képaláírásra

Az ábra sorszáma mellé mindig érdemes rövid magyarázatot is fűzni. Érdemes tudni, hogy Wordben számos beállítási móddal lehet képet beszúrni. Tapasztalni fogod, hogy számodra a legpraktikusabb a „szöveggel egy sorba” elrendezés lesz, mivel így szépen együtt marad az ábra a szöveggel. Oldaltörésekkor az ábrák méretének módosításával tudod megoldani, hogy ne maradjanak nagy üres felületek a dolgozatodban.

### Képminőség

A raszteres (tehát nem vektorgrafikus) képek használata különös körültekintést igényel. Ezek kiválóan néznek ki a 90 dpi-s monitorodon, ám a 600/1200 dpi-s nyomtatókon kinyomtatva rendkívül bénák lesznek a szép, pixelmentes szövegek és vektorgrafikus ábrák mellett. Hogy lásd, mire számíthatsz, nézzük meg az előző ábrát 5-szörös nagyításban.



5. ábra: A raszteres képek nyomtatásban csúúúúnyák lesznek

Ha tehát lehetséges, használjunk vektorgrafikus ábrákat, vagyis a diagramokat, forráskódot stb. ne képenyőképeken keresztül, hanem közvetlen copy-paste megoldással másoljuk át a dokumentumunkba.

Ha elkerülhetetlen a raszteres képek használata, akkor próbáljunk meg minél magasabb felbontású képet berakni. Bár a képernyőn nem feltétlenül látszik majd a különbség, de a nyomtató ki tudja majd használni a magasabb felbontású képet. A 6. ábra például igen magas felbontású, amit láthatsz is, ha felnagyítod.



6. ábra: Ez az ábra 460 × 750 pixel felbontású

Figyelj rá, hogy a PDF-be mentéskor a Word (2013) alapértelmezésben jelentősen lerontja a képek minőségét, sőt, a vektorgrafikus ábrákból is rasztereset csinál. Ha a nyomtatást PDF alapján akarod csinálni, érdemes külső szoftvert használni a Word 🡪 PDF konverzióra, vagy alaposan eljátszani a Word beállításaival.

## Táblázatok

Bizonyos adatokat táblázatos formában érdemes ábrázolni. Például: „A probléma megoldására a két fő megoldási irány az A és a B opció, melyek különböző funkciókat támogatnak (lásd 1.1. táblázat).”

|  |  |  |
| --- | --- | --- |
|  | A opció | B opció |
| 1. funkció | + | + |
| 2. funkció | + | − |
| 3. funkció | − | + |
| 4. funkció | − | − |

1.1. táblázat. Példa táblázat feliratára

## Kereszthivatkozások

Amennyiben szeretnél egy ábrára, vagy korábbi fejezetre hivatkozni, használd a Word „Hivatkozás/Kereszthivatkozás” menüpontját. Kerüld az így beszúrt szövegek közvetlen ragozását, hiszen az könnyen változhat, amennyiben a forrás változik. Fejezetekre, ábrára sose a teljes szövegükkel, csupán a sorszámukkal („Csak címke és szám”) hivatkozz. Ezt szemlélteti a 7. ábra.



7. ábra: Ábrahivatkozás beszúrása

Tipp: érdemes elkerülni az „előző oldalon”, „következő oldalon” fordulatokat, ugyanis az ábrák végső helyzete a tördelés során még megváltozhat.

Amennyiben az „ábra” szót szeretnénk ragozni (pl. „lásd az x. ábrán”), akkor a fenti megoldás nem alkalmazható. Helyette jelöljük ki a képaláírásban a sorszámot (pl. „1.1.”), kattintsunk a **Könyvjelző** gombra, majd hozzunk létre egy könyvjelzőt (pl. „bmelogo” névvel). Ezután a **Kereszthivatkozás** gombra kattintva a **Hivatkozástípus**t állítsuk **Könyvjelző**re és válasszuk ki a **bmelogo** könyvjelzőt. Sajnos azonban ez a megoldás sem tökéletes: a Word ugyanis nem támogatja a megfelelő határozott névelő beszúrását. Egy újabb ábra beszúrása esetén a névelőket egyesével kell ellenőriznünk – például ha a dokumentum elejére beszúrunk egy új ábrát, akkor a korábbi „az 1. ábrán” hivatkozásból „az 2. ábrán” lesz.

## Irodalomhivatkozások

Az irodalomhivatkozások kezelésére a Word egy kényelmes és jól használható funkciót kínál. Amikor dolgozatodban egy külső műre, weboldalra, könyvre, előadásra stb. szeretnél hivatkozni, használd a „hivatkozás/források kezelése” menüpontot. Itt felveheted az egyes műveket, illetve a hozzájuk kapcsolódó szerzői adatokat. Folyóiratcikkeknél a szerzők mellett szerepeljen a pontos cím, a folyóirat címe, évfolyam, szám, oldalszám tól-ig. A folyóirat címeket csak akkor rövidítsük, ha azok nagyon közismertek vagy nagyon hosszúak. Internet hivatkozások megadásakor fontos, hogy az elérési út előtt megadjuk az oldal szerzőjét és címét (mivel a link egy idő után akár elérhetetlenné is válhat), valamint az elérés időpontját.

Az egyes forrásmunkákra a dolgozat szövegében lehet hivatkozni a „hivatkozás/idézet beszúrása” menüponttal. Figyelj rá, hogy az IEEE[[1]](#footnote-1) stílust használd.



8. ábra: Idézetek és irodalomjegyzék

### Pozícionálás

Az irodalomhivatkozások a szövegtörzsben, ábrák szövegében és táblázatokban is előfordulhatnak, de **fejezetcímekben soha**. Amennyiben egy adott forrás egy egész bekezdésre/fejezetre vonatkozik, akkor is elég, ha az első mondat/első bekezdés után megemlítjük. Szó szerint idézett szöveg esetén az idézet végén, külső forrásból átvett ábra esetén a képaláírásban hivatkozzunk a forrásra.

### Mikor kell hivatkoznom?

Minden külső forrásból átvett képnél, szövegrésznél, olyan szövegeknél, melyek megfogalmazásához külső forrásokat használtál, illetve olyan állítások/technológiák/algoritmusok megemlítésénél, melyek nem feltétlenül egyértelműek egy átlagos műveltségű olvasó számára. Nagyságrendileg egy szakdolgozatban átlagosan 10-20, egy diplomatervben átlagosan 20-30 külső forrást illik megemlíteni. A források megválasztásánál, ha lehet, törekedjél a nyomtatott források keresésére, ezek lehetnek folyóiratcikkek, könyvek, könyvfejezetek, útmutatók, egyetemi jegyzetek stb., és csak akkor hivatkozz weboldalakra vagy Wikipediára, ha ilyet nem találtál.

### Irodalomjegyzék megvalósítása

Az irodalomjegyzék kezelése többféleképpen is megoldható, az alábbiakban két egyszerű módszert ismertetünk.

#### Kereszthivatkozásokkal

A kereszthivatkozásokkal történő irodalomjegyzék egy megfelelően formázott felsorolás, melynek egyes elemeire (bekezdéseire) mutatnak hivatkozások. Jelen dokumentum ezt a megközelítést alkalmazza.

Az Irodalomjegyzékben szereplő hivatkozásokat **Irodalomjegyzék bejegyzés** stílussal formázzuk, a címüket pedig a **Kiemelés** stílussal emeljük ki.

A szövegbe a hivatkozásokat a **Kereszthivatkozás beszúrása** (Insert cross-reference) funkcióval helyezzük el (példa egy így beszúrt hivatkozásra: [1]), így azok később frissíthetők a hivatkozások átrendezése esetén (lásd 2. fejezet).

#### Források kezelése

A kereszthivatkozások alternatívája, hogy a hivatkozott műveket először felvesszük a szerkesztőprogram adatbázisába. Új műveket a **Hivatkozás** fülön a **Források kezelése** alatt az **Új…** gombbal vehetünk fel. A szerzőket érdemes a **Szerző** mező mellett található **Szerkesztés** gomb használatával felvenni.

Az irodalomjegyzéket az **Irodalomjegyzék** gomb alatt az **Irodalomjegyzék** opcióval szúrhatjuk be a dokumentumba. A hivatkozások stílusa a **Stílus** gomb alatt állítható be, a javasolt stílus az **IEEE**.

## Word tippek és trükkök

### Navigációs ablak

Mivel a fejezeteket nagy valószínűséggel nem sorrendben fogod tartalommal feltölteni, érdemes bekapcsolni a navigációs ablakot (nézet/navigációs ablak). Ez gyors és egyszerű keresési és ugrálási lehetőséget biztosít a fejezetek közt



9. ábra: Navigációs ablak

### Megjegyzések

A véleményezés/megjegyzés menüponttal megjegyzéseket fűzhetsz a szöveg tetszőleges részéhez. Ezt konzulensed előszeretettel fogja majd használni az észrevételei megfogalmazásához, de neked is jól jöhet, hogy megjelöld azokat a részeket, ahova még vissza kell térned. Fontos, hogy nyomtatás előtt az összes megjegyzést töröld, vagy külön állítsd be, hogy a megjegyzések ne jelenjenek meg a nyomtatásban.

### Korrektúra

Konzulensed, vagy mások, akiket a dolgozat átolvasására megkérsz legegyszerűbben a korrektúra funkció használatával tudják úgy javítani a dolgozatod, hogy a javításokat te is lásd. Az átolvasáshoz tehát érdemes bekapcsolni a „változtatások követése” funkciót és „Minden korrektúra” mutatására beállítani a rendszert.

Amikor korrektúrázott változatot kapsz vissza valakitől, használd a „Véleményezés ablak”-ot, így egyetlen javítás sem kerülheti el a figyelmedet, s így mindegyiket elfogadhatod vagy elvetheted, igény szerint.



10. ábra: Ellenőrizd, hogy minden bekarikázott funkciót ismersz-e!

### Gyorsbillentyűk

A Word lehetőséget ad tetszőleges funkció gyorsbillentyűre kötésére. Használd a „billentyűparancsok testreszabása” funkciót, hogy gyorsbillentyűt rendelj a leggyakrabban használt gombokhoz.

Érdemes továbbá gyorsbillentyűt rendelni az alapvető stílusokhoz is. Ebben a dokumentumban hozzárendeltük az Alt + K kombinációt a „Kép” stílushoz.

## Kódrészletek

Érdekesebb és bonyolultabb programozási megoldásainkat bátran illusztrálhatjuk kódrészletek beszúrásával. Fontos, hogy a beillesztett kódrészlet mérete álljon arányban annak a fontosságával, tehát ritkán érdemes egy „bekezdésnyi” kódnál többet beszúrni egyszerre. Amennyiben egy bonyolultabb (akár több oldalas) algoritmust szeretnénk bemutatni, annak a kódját érdemesebb függelékbe rakni.

### Formázás

A kódrészletek formázásánál kerüljük a helypazarlást, illetve próbáljuk megelőzni az olvashatóságot rontó sortördelést, akár a forráskód módosításának árán is. Két praktikus tipp Visual Studióhoz:

Másolás előtt érdemes a behúzások mértékét 4-ről 2 karakterre csökkenteni, majd a másolandó kódrészletet újraformázni (Ctrl + K, Ctrl + D), alternatívaként persze utólag is kitörölgethetjük a felesleges szóközöket a kódból.



11. ábra: A behúzások mértékének csökkentése Visual Studióban

A másolandó kódrészlet kijelölésekor tartsuk lenyomva az Alt billentyűt. Így a valóban releváns részekre korlátozhatjuk a kijelölést.



12. ábra: Kódblokk kijelölése Alt + egérkurzor segítségével

Kódrészletek beillesztésére semmiképpen ne képeket használjunk, hanem a közvetlenül beszúrt kódrészletekkel dolgozzunk. A Visual Studióból kimásolt kódok alapvetően jól néznek ki, további szépítésre használjuk a **Kód** stílust.

static void Main(string[] args)

{

var ci = new CultureInfo("en-us");

ci.NumberFormat.CurrencySymbol = "";

Thread.CurrentThread.CurrentCulture = ci;

Console.WriteLine(ci);

Console.WriteLine("{0:c}", 5.66);

}

### Irodalomjegyzék

Az Irodalomjegyzékben szereplő hivatkozásokat **Irodalomjegyzék sor** stílussal formázzuk, a címüket pedig **Irodalomjegyzék forrás** stílussal emeljük ki.

A szövegbe a hivatkozásokat a Kereszthivatkozás beszúrása (Insert cross-reference) funkcióval helyezzük el, így azok automatikusan frissülnek a hivatkozások átrendezésekor.

## Utolsó simítások

Miután elkészültünk a dokumentációval, ne felejtsük el a következő lépéseket:

* Kereszthivatkozások frissítése: miután kijelöltük a teljes szöveget (Ctrl + A), nyomjuk meg az F9 billentyűt, és a Word frissíti az összes kereszthivatkozást. Ilyenkor ellenőrizzük, hogy nem jelent-e meg valahol a „Hiba! A könyvjelző nem létezik.” szöveg.
* Dokumentum tulajdonságok megadása: a dokumentumhoz tartozó metaadatok kitöltése (szerző, cím, kulcsszavak stb.). Ez Word 2013 alatt a **Fájl** | **Információ** | **Tulajdonságok** | **Dokumentumpanel megjelenítése** gombra kattintva érhető el, míg Word 2016-ban a **Fájl** | **Információ** | **Tulajdonságok** | **Minden tulajdonság megjelenítése** alatt**.**
* Kinézet ellenőrzése PDF-ben: a dokumentum elkészítése után feltétlenül ellenőrizzük a kapott PDF dokumentumot is.

# Köszönetnyilvánítás

A köszönetnyilvánítás nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

# Irodalomjegyzék

1. Jeney Gábor, Hogyan néz ki egy igényes dokumentum? Néhány szóban az alapvető tipográﬁai szabályokról, Budapesti Műszaki és Gazdaságtudományi Egyetem, Híradástechnikai Tanszék, Budapest, 2007. május 9., online: <http://mcl.hu/~jeneyg/foliak.pdf>
2. William Strunk Jr., E. B. White, The Elements of Style, Fourth Edition, Longman, 4th edition, 1999.
3. Levendovszky, J., Jereb, L., Elek, Zs., Vesztergombi, Gy., Adaptive statistical algorithms in network reliability analysis, Performance Evaluation – Elsevier, Vol. 48, 2002, pp. 225-236
4. National Instruments, LabVIEW grafikus fejlesztői környezet leírása, <http://www.ni.com/> (2014. aug.)
5. Fowler, M., UML Distilled, 3rd edition, ISBN 0-321-19368-7, Addison-Wesley, 2004

# Függelék

A függelék szövege.

1. IEEE (ejtsd „I triple E”): Institute of Electrical and Electronics Engineers [↑](#footnote-ref-1)