

# Általános tanácsok szakdolgozat és diplomamunka készítéséhez

A célegyenesben vagy, hogy megkapd a diplomádat. Ha megírod a szakdolgozatodat / diplomamunkádat (és megvéded, meg leállamvizsgázol persze), akkor semmi nem menekít meg attól, hogy programtervező informatikus légy.

Azért írjuk ezt a kis informális szösszenetet, hogy segítsen téged a szakdolgozat vagy diplomamunka írásában<sup>1</sup>. Ez egy útikalauz ahhoz, hogy megfelelően tudd használni az adott szakdolgozat/diplomamunka sablon fájlt, továbbá a szakdolgozattal / diplomamunkával kapcsolatos összes kérdéskört igyekszik körbejárni és irányt mutatni.

Szerzők:

Németh Gábor Árpád

Gludovátz Attila

Gombos Gergő

Budapest, 2021 december 12.

---

<sup>1</sup> ... és remélhetőleg egy picit a konzulenseket is segíti abban, hogy ne ugyanazokat kelljen mindenkinek elmondani, így több pálmafa, napfény, csillogás marad nekik is...

## Tartalomjegyzék

1.	Hasznos tudnivalók .....	1
1.1.	Miben írd a munkát?.....	1
1.2.	Mire figyeljek a dolgozat írásánál?.....	1
1.3.	Hogyan mutassam meg a konzulensnek / témavezetőnek?.....	1
2.	Általános tanácsok .....	2
2.1.	Stílus.....	2
2.2.	Problémák bemutatása, tervezési döntések.....	3
2.3.	Hivatkozások, irodalomjegyzések, szoftverek.....	3
2.4.	Ábrák és táblázatok.....	3
2.5.	Jelölések, rövidítések, definíciók, érthetőség .....	4
2.6.	Formázás .....	4
2.7.	BSc Szakdolgozat specifikus megjegyzések.....	5
2.7.1.	Igeidő.....	5
2.7.2.	Irodalomjegyzék.....	5
2.8.	MSc Diplomamunka specifikus megjegyzések.....	5
2.8.1.	Igeidő.....	5
2.8.2.	Irodalomjegyzék.....	5
3.	A munka ütemezése.....	6
4.	Mi a nyavalyát fog nézni a bíráló? .....	7
5.	Szakdolgozat / Diplomamunka szerkezete .....	8
5.1.	A dolgozat eleji fejezetek .....	8
5.1.1.	Abstract / Kivonat .....	8
5.1.2.	Tartalomjegyzék, rövidítés-, jelölésjegyzék .....	9
5.1.3.	Bevezetés .....	9
5.2.	A BSc Szakdolgozat fejezetei .....	9
5.2.1.	Felhasználói dokumentáció.....	9
5.2.2.	Fejlesztői dokumentáció .....	9
5.3.	MSc Diplomamunka fejezetei .....	10
5.4.	A dolgozat végi fejezetek .....	10
5.4.1.	Összefoglalás (kb. 1 oldal terjedelem) .....	10
5.4.2.	Továbbfejlesztési lehetőségek (amennyiben értelmes, kb. 1 oldal terjedelem) .....	10
5.4.3.	Irodalomjegyzék.....	10
5.4.4.	Függelékek .....	10
6.	Leadás előtti ellenőrzőlista .....	11
7.	Irodalomjegyzék.....	12

# 1. Hasznos tudnivalók

## 1.1. Miben írd a munkát?

- Szabad a választás, de a legtöbbet használt szövegszerkesztők a Word és a Latex. Mind a két eszközhöz található sablon, amiből ki tudsz indulni.

## 1.2. Mire figyeljek a dolgozat írásánál?

- A dolgozaton belüli hivatkozások legyenek megfelelően kezelve. Erre mind a Latex, mind a Word képes.
  - **Word:** *Beszúrás / Hivatkozások szekció / Kereszthivatkozás*), amelyek használhatóak szekciócímek, ábrák, irodalomjegyzéktételek hivatkozásainál.
  - **Latex:** lásd. `\label{}` elemek behivatkozása `\ref{}` kulcsszóval.

## 1.3. Hogyan mutassam meg a konzulensnek / témavezetőnek?

- Mind témavezető más és más, emiatt ezt mindig **egyeztesd le vele**. Azért leírunk pár példát, amit alkalmazni szoktak:
  - **Word korrektúra:** Word tud két tetszőleges file verziót összehasonlítani (*Véleményezés / Összehasonlítás*), de van változások követése opció is (*Véleményezés / Nyomon követés szekció / Változások követése*) - ez utóbbi viszont csak akkor működik, ha bekapcsolod.
  - **Latex forrás:** Latex-nél a forrásfájlokat könnyű diff-elni. Latex választása esetén viszont éppen ezért javasolt a forrásfájlt is mellékelni.
  - **Overleaf:** Ha a konzulens nem szeretne különféle fordítók különféle package-eivel stb. szívní, akkor megosztott [overleaf](#) projektként is lehet írni a munkát.
  - **OneDrive, Google Drive:** A konzulens vérmérsékletétől függően lehet megosztani OneDrive-ban vagy Google Drive-ban egy közös doksit (előbbinél érdemes asztali Word alkalmazást használni, mert a webes megoldás jelenleg (2021 december) még igencsak bugos).
  - **PDF:** Előfordulhat azonban, hogy a konzulens mindig csak egy letisztázott verziót fog kérni tőled a dokumentációd előrehaladásáról, amihez remek választás a PDF formátum (ami mind Word-ből, mind Latex-ből generálható) az egységes megjelenés miatt.

## 2. Általános tanácsok

### 2.1. Stílus

- Szakszöveget írsz és nem este a kocsmában italozgatva beszélgetsz a haverokkal. Tehát ügyelj arra, hogy szabatosan, tényszerűen, precízen fogalmazz! Ha nem így teszel, akkor az adott mondatot „pongyola” kommenttel jelölheti meg a konzulens.
  - Erősebb állításoknál emellett számíts arra, hogy azt irodalmi hivatkozással vagy esetleg bizonyítással kell alátámasztanod.
    - Külön kitérnénk az „optimális” szó használatára. Egyrészt eleve felső fok, tehát tovább nem fokozható (az „optimálisabb” és társai halálfejes hiba). Másrészt nagyon erős kijelentés. Ha valamire azt mondod, hogy optimális, akkor azt biztos indokolnod kell (bizonyítással, hivatkozással stb.).
- Szinonimák használata nem tilos, sőt üdvözlendő! A sok szóismétlés döcögőssé, nehezen olvashatóvá teszi a szöveget. Viszont szinonimákkal gördülékenyebb lehet a szöveg.
  - Szakszavak esetében viszont csak mértékkel a szinonimákkal:
    - Vigyázz arra, hogy ezek tényleg ugyanazt jelentsék, ne ugyanannak a dolognak árnyalatait, fokozatait (pl. magyarul csak „hiba”, de az angolban a „fault”, „error”, „failure” más-más szinteket jelöl) vagy netán mappeléseit (pl. „edge” és „transition” a graph/FSM (Finite State Machine) esetben).
    - Nem biztos, hogy mindenkinek egyértelmű, hogy máshogyan is lehet megnevezni valamit. Ezt ekkor definícióban jelölni kell és inkább a következetes jelölés javasolt (pl. „edge” vs. „arch”, „node” vs. „vertex”).
- Olvasd vissza a szöveget mindig! Egyeztesd a ragokat, igeidőket! A rovar részei: fej-tor-potroh, a mondat részei alany-állítmány-tárgy, mellékmondatnál is ellenőrizd, hogy mire utalsz vissza, és egyeztesd össze a főmondattal. Használj helyesírás ellenőrzőt! Komolyan rossz érzés olyan szöveget olvasni, ahol a konzulensnek az lehet az érzése, hogy azt ő olvassa először – beleértve a dolgozat íróját is...
- Bár nagyon szépek Cicero hosszas ókori latin, már-már barokkos körmondatai (hogy ezzel az anakronisztikus fordulattal éljünk), és bár tagadhatatlanul tud szellemi kihívás lenni egy hét-nyolc soros mondat megértése a különböző egyeztetésekkel, összetett mellékneves jelzőkkel – közbeszúrásokkal – alá-, mellé- és fölrendelt mondataival (zárójelben lévő szellemes megjegyzéseivel), előre és visszahivatkozásaival, pontosításaival és egyéb, a mondatot feleslegesen bővítő szerkezeteivel, hogy csak a legfontosabbakat említsük és – amit típushibaként jónéhány hallgató időről-időre elkövet – neked mégsem kell erre és a bíráló agytorna elé állítására törekedni, sőt. Milyen volt például az előző mondatot olvasni? 😊 Ha egy mondat két sornál hosszabb, akkor fogjál gyanút, hogy szét kellene szabdálni több mondatra.
- Sokszor valamit könnyebb úgy elmagyarázni, hogy mutatsz egy példát. Ha valami nehezen érhető, sok lépésből álló összetett dolgot próbálsz elmagyarázni, akkor megoldás lehet, hogy (1) előbb elmondod röviden, hogy mik lesznek a főbb lépések és miért, (2) majd példával illusztrárod, (3) legvégül leírod precízen lépésről lépésre, hogy még a legkukacosabb matematikus se köthessen bele<sup>5</sup>.
- Remek, hogy megtanultad a különféle UML (Unified Modelling Language) ábrákat, de feleslegesen ne használd őket csak azért, hogy x oldallal több legyen a pályamunkád; csak akkor és ott, ahol tényleg segítik a megértést. Nyilván tématerület függő, hogy kell-e / milyen kell / mennyi kell.
- Érdemes figyelni arra is, hogy múlt vagy jelenidőben írsz a dolgozatot. Ennek kapcsán is azt javasoljuk, hogy a legfontosabb a következetesség, tehát ha valamelyik mellett elköteleződünk, akkor azt tartjuk meg a dolgozat írásának végéig.
- Legyen mindig egy íve a dolgozatnak, ne csapongj, vezesd át a fejezeteket vagy azokon belüli részeket egymásba!

---

<sup>5</sup> A szerzők nem matematikusok, de adjuk meg az ELTE-n lévő sok matekos tanszéknek a tiszteletet... 😊

## 2.2. Problémák bemutatása, tervezési döntések

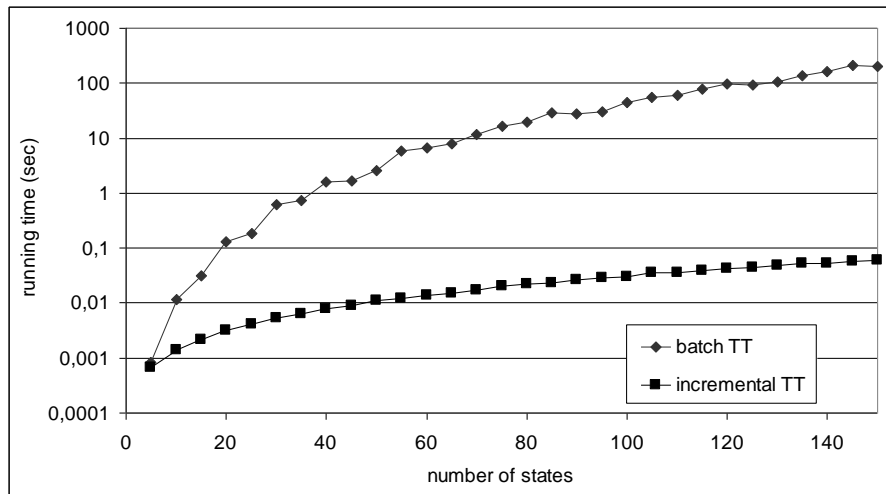
- Ha megvizsgálasz valamilyen problémát, és aztán döntesz valahogy, akkor mindig legyen világos, hogy (1) hoztál egy tervezési döntést (design decision) és hogy azt (2) miért tetted: milyen egyéb alternatívák voltak, azoknak mi volt az előnye és hátránya, és miért az adott megoldást használtad végül.
  - Tervezési döntés lehet például egy programozási nyelv, egy szoftver, egy metodológia, egy algoritmus, egy adatábrázolási mód, egy metrika, vagy egy tesztelési módszertan választása is.
- Az alapprobléma és a tervezési döntés témájától függően a döntési szempontok is sokfélék lehetnek: futási idő/komplexitás, tárhely komplexitás, támogatottság/dokumentáció mértéke stb. Lehetőleg próbálj egy problémát, tervezési döntést minél több szempontból megvizsgálni elemezni.

## 2.3. Hivatkozások, irodalomjegyzékek, szoftverek

- Felhasználhatsz adott esetben máshonnan ábrákat és szövegrészeket, de azok **forrását mindenképpen jelöld!** Értelem szerint szövegnél, ha szó szerint idézel, akkor tedd is idézőjelbe: szövegnél mindig az adott mondathoz/mondatrészhez, ahova vonatkozik az idézet. Jelölni kell azt az irodalomjegyzék tételt is, ahonnan hivatkozod a bölcsességet (használd a kereszt-hivatkozásokat, azok itt is működnek).
- Irodalomjegyzékhez kb. mindegy, hogy milyen hivatalos (standard) formátumot használsz (pl. [Latexnél is több van](#), pl. ACM, apalike, plain stb.), de legyen egységes és következetes. Személy szerint talán a [szám] típusút preferáljuk, mint a grafomán [sok hosszú vezetéknev évszámmal] típusú, de igazából mindegy, hogy mit használsz.
  - Hogy ne csak ködszurkálás legyen ez a pont, megadtunk egy-egy lehetséges példát szabvány [1], cikk [2] és weboldal [3] hivatkozásra.
- Bár sok kiváló magyar koponya van és a magyar puma mindig ugrásra készen áll, a szakma nyelve azért mégis az angol, tehát célszerű lehetőleg nagyobb részben angol nyelvű szakirodalmat olvasni, hivatkozni stb.
- Ha használsz bármilyen programnyelvet, szoftvert, 3. féltől származó programcsomagot stb., akkor azt verziószámmal és weblinkkel (jelölve a legutolsó elérést) hivatkozod be, akár irodalomjegyzék tételként, akár lábjegyzetként (vagy ha nagyon indokolt, akkor sok szoftver tétel esetében lehet ezeknek külön „irodalomjegyzék” vagy alfejezet a releváns helyen).
- Ha konzulensed megszórt sok cikkel, amit érdemes elolvasnod, akkor érdemes megnézni ezt a vonatkozó cikket [4], hogy ezt hogyan lehet hatékonyan tenni.

## 2.4. Ábrák és táblázatok

- Ábráknak és táblázatoknak mindig legyen száma és mindig a számuk alapján hivatkozzál rájuk a folyószövegből (képzeld azt, hogy az olvasó egy preprocessor: ami nincsen belinkelve a főszövegből azt nem is látja, számára nem létezik). Sok szívástól megkíméled magad, ha ezt kereszt-hivatkozással oldod meg, és nem szimplán begépeled az ábra számát (utóbbi esetben úgyis el fog csúszni). Mindig legyen képaláírás és abból egyértelmű legyen, hogy mire vonatkozik (pl. Scenario 3 simulations: execution times, XYZ framework components,  $A_M$  auxiliary graph of FSM  $M$  stb.). Táblázatokhoz ugyanez a megjegyzés. Fontos, hogy az ábra a szövegnek megfelelő helyén legyen (igen, ez a végén szerkesztési szívás, mind Word-ben, mind Latex-ben, de hát nem adják ingyen azt a diplomát). Ha nem saját ábrát / táblázatot használ, akkor mindenképpen jelöld meg a forrásukat akár ott közvetlenül zárójelben/lábjegyzetben, akár irodalomjegyzéki hivatkozásként.
- Ha grafikont használ, hogy futási eredményeket mutass be, akkor a tengelyeknek mindig legyen felirata és mértékegysége. Több algoritmus futási eredményeit ábrázolhatod ugyanazon ábrán, ha a skálák azonosak. Ekkor az egyes algoritmusok futását más-más színnel és stílussal jelölheted (pl. szaggatott, pontozott vonalak, mérési pontokon háromszögek, négyzetek, körök stb., gondolj a fekete-fehér nyomtatásra is), és feltétlenül jelöld az ábrán belül jelkulccsal, hogy melyik melyikre vonatkozik. Nagyon elszálló függvények, vagy egymástól távol lévő külön mérési vonalak esetén megfontolandó a logaritmus skála használata is (lásd 1. ábra).



1. ábra: 1. szcenárió szimulációinak végrehajtási ideje<sup>7</sup>

## 2.5. Jelölések, rövidítések, definíciók, érthetőség

- Úgy kell írni a szöveget, hogy „anyám is megértse”. Nyilván nem egyiptológusok, filozófusok vagy bányászok a célközönség, de egy megfelelő alap műszaki (informatikai/mérnöki) háttérrel rendelkező embernek meg kell értenie csak a dolgotból magából, hogy pontosan mit csináltál<sup>8</sup>.
- Ha bármilyen rövidítést használsz, akkor old fel a rövidítést az első előfordulásánál. Még boldogabb lesz a konzulensed, ha ezen felül lesz egy rövidítések feloldása fejezet (javasolt doksi elején), ahol benne van az összes rövidítés abc rendben (Latexnél van is támogatás erre, ami még az előfordulásokat is indexeli ld. pl. `\index{DAG -- Directed Acyclic Graph}`). Ráadásul ezzel együtt 1 oldallal hosszabb lesz a munkád és szebben is fog kinézni.
- Ha bármi jelölést használsz a szövegben (pl. programváltozóra, definíció adott elemére stb.), akkor azt különítsd el más stílussal (pl. dőlt betű) és azt a jelölést használd rá következetesen a szövegben (futóváltozókat persze újra lehet használni ismét futóváltozóként). Ha sok ilyen jelölés lesz, akkor megfontolandó esetleg egy jelölésrendszereket összefoglaló lapot bevezetned (javasolt a doksi elején).
- Az általad használt fogalmakat definiáld, még ha alapfogalom is (pl. „Model-based testing” téma esetén: irányított gráf, erősen összefüggő és egyéb használt tulajdonságok, Véges Automata és alaptulajdonságok definíciója). Viszont fontos, hogy csak a használt fogalmakat definiáld (tehát előbbi téma esetén ne írd gráfelmélet alapjai c. könyvet).
- A fentieket úgy is össze lehetne foglalni, hogy a dokumentum megálljon a saját lábán, egy műszaki beállítottságú személy megértse a háttérrel utána olvasás nélkül is.

## 2.6. Formázás

- Már munka közben is legyen mindenképpen egy világos struktúra: számozott alfejezetekkel, rendes stílusokkal, formázással (sorkizárt, lehetőleg 1.5 sorközzel), de ahogyan közeledsz a vége felé mindenképpen át kell nézned, hogy mennyire „pofás” egyben az egész:
  - A főfejezetek új oldalon kezdődjenek, az alfejezetek ne nagyon kezdődjenek az oldal alja felé (legalább 3 sor kövesse).
  - Legyen oldalszámozás arab számokkal.
  - Ne legyenek a folyószövegben tördelési hibák, pl. fattyúsor (amikor egy új bekezdés egy oldal utolsó sorában végződik, vagy amikor egy bekezdés utolsó sora már új oldalra kerül) kerülendő. Szerencsére ezekben mind segít a sablon fájl alkalmazása.

<sup>7</sup> Kép forrása: Gábor Árpád Németh and Zoltán Pap. The incremental maintenance of transition tour. Fundamenta Informaticae, 129(3):279–300, July 2014. DOI: 10.3233/FI-2014-972.

<sup>8</sup> Ok, lehet anyukád is informatikus és villamosmérnök, mint pl. a dokumentum egyik szerzőjénél, de azért talán értjük a mondanivaló lényegét...

- A képek helyzetére figyelj a folyószövegben. Hivatkozáshoz képest ne nagyon csúszanak el (Latexnél ez szívás lehet és „játszadozást” igényel az elhelyezést jelző kapcsolókkal, pl.  $[h]$ ,  $[!h]$  és társai, lehet, hogy át is kell méretezned), méretük legyen megfelelő (szövegek rajta kellően olvasható méretben).
- Ismét megemlítjük, hogy a hivatkozások helyességét ellenőrizd, frissítsd, amennyiben ez automatikusan nem történik meg!

## 2.7. BSc Szakdolgozat specifikus megjegyzések

### 2.7.1. Ideidő

- „Én” vagy „Mi”... vagyis milyen személyben írjuk a szakdolgozatot / diplomamunkát? **Szakdolgozat** esetén, mivel egy szoftvert készíttetek el, ezért javasolt az „én” (E/1) használata, mivel ti csináljátok és a konzulensetek maximum tanácsaival, útmutatásaival látja el az előrehaladásotokat.

### 2.7.2. Irodalomjegyzék

- **Szakdolgozat** esetén: mindenképpen érdemes bújni az angol és/vagy magyar szakirodalmat a szoftvered elkészítésének témájában. Arra figyelj, hogy lehetőleg ne legyen 5 évnél régebbi az az irodalom, amit majd idézel. Érdemes továbbá használni a programnyelvek, környezetek, programcsomagok stb. hivatalos dokumentációit is, ezeket feltétlenül lehet használni és hivatkozni is rájuk. Fórumbejegyzéseket is olvashatsz, nagy hasznodra fog válni biztosan a Stackoverflow vagy a Github fórumainak egy-egy bejegyzése, ezekre hivatkozzál bátran. Ezeknél azonban vedd figyelembe, hogy akár az adott fórumkérdés feltevője is törölheti a bejegyzést, így feltétlenül írd oda az irodalomjegyzékes bejegyzésed mellé, hogy mikor tekintetted meg utoljára az adott oldalt („Utolsó megtekintés dátuma” szöveget és utána a dátumot helyezd el a bejegyzésben).

## 2.8. MSc Diplomamunka specifikus megjegyzések

### 2.8.1. Ideidő

- „Én” vagy „Mi”... vagyis milyen személyben írjuk a szakdolgozatot / diplomamunkát? **Diplomamunka** esetén, mivel már egy tudományos tekintetben is értékelhető dolgozat készül el, ezért van lehetőségetek választani az „én” (E/1) és a „mi” (T/1) között. Ez akár attól is függhet, hogy a konzulensed mennyire vett részt a veled közös munkában, vagy akár irodalmi stílus megfontolásból is használhatod a többes szám első személyű megfogalmazásokat. Így akár a dolgozat olvasóját is be tudod vonni az együtt gondolkodásba, személyesebbé tudod tenni a munkádat ezzel. A legfontosabb viszont az, hogy következetesen használjátok az E/1-et vagy T/1-et az egész műben.

### 2.8.2. Irodalomjegyzék

- **Diplomamunka** esetén: ne felejtsetd el, hogy szakszöveget írsz. Tehát ne fórumbeszélgetéseket / blogokat vagy akár ppt előadásokat hivatkozzál be<sup>9</sup>, a Wikipédia sem javasolt, sőt, inkább tiltott. Viszont egy könyv vagy lektorált konferenciacikk / folyóiratcikk már jó lehet, ha meg ráadásul 5 évnél nem régebbi, akkor kifejezetten state-of-the-art irodalmat ismerőnek tűnhetsz majd (ez utóbbi inkább kutatási jellegű témáknál lehet indokolt). A Google / Stackoverflow / Github fórumbejegyzés linkek itt már nem megfelelőek (nem lektoráltak vagy ellenőrzöttek, törölhetőnek/változhatnak az idők során stb.), ha mindenképpen ilyet szeretnének használni, akkor tegyük inkább lábjegyzetbe.

---

<sup>9</sup> Fun fact: Volt egy olyan permutációs problémáról szóló matekos szakcikk [5], hogy egy problémára alsó korlátot egy animevel foglalkozó fórum anoním kommentelője oldott meg. Mivel nem tudták kinyomozni, hogy ki volt az illető ezért végül úgy hivatkozták be, hogy egy anoním szerző a 4chanfórumról [6] (a sztorid amúgy ld. itt: <https://www.wired.com/story/how-an-anonymous-4chan-post-helped-solve-a-25-year-old-math-puzzle/>), de azért érezzük, hogy nem ez a mainstream...

### 3. A munka ütemezése

- Hivatalos határidőkre te figyelj!
- Készíts magadnak ütemtervet és próbáld tartani! Vannak szuper szoftverprojekt támogató programok, például a ClickUp<sup>10</sup>. Ez szakdolgozat esetén mindenképpen hasznodra válhat a szoftvered elkészítéséhez, de ha a diplomamunkás kutatásodhoz készítesz valamilyen szoftvert, ott is érdemes lehet az alkalmazása.
- Ha bármi nagyobb időtartam kiesés van a félév során, amit előre láatsz, feltétlenül jelezd a konzulensnek!
- Érdemes a félév elején egyeztetni a tervet az előrehaladásról, utána pedig időről-időre jelentkezni a konzulensnél, amikor érdemi előrehaladás történt.
  - Tehát még egyszer: akkor jelentkezz, ha van érdemi haladás, vagy kérdésed van, ekkor viszont mindenképpen: erre vannak a konzultációk.
- Tekints a konzulensre, mint egy autonóm, jó szándékú, de korlátozott erőforrásra, ami eléréséhez versengsz a többi hallgatóval. Ebből ugyanis a következők vezethetők le:
  - Ha nem utolsó pillanatban kezded a munkát, hanem előbb jelentkezel, akkor jó eséllyel többet tud segíteni.
  - A témavezető az idejét a hallgatók között annak arányában osztja be, hogy a többi hallgatóhoz viszonyítva mennyire érdemes. Szép munkára, nagyobb haladásra, ahol úgy érzi, hogy a korábbi kommentjei hatással voltak a munkára szívesebben szán ő is több időt.

---

<sup>10</sup> <https://clickup.com/>



## 4. Mi a nyavalyát fog nézni a bíráló?

- A hivatalos értékelési szempontokat megtalálod az [ELTE IK vonatkozó oldalán](#) a „Diplomamunka bírálata” menüpont alatt. Amit kiemelnénk mindenképpen:
  - Legyenek egyértelműek a célok.
  - Legyen egyértelmű a háttér, amire építesz (mennyire jól írtad azt le, vonatkozó irodalmak ismerete és hivatkozása stb.)
  - Megfelelő módszereket választottál-e a feladat megoldásához, alternatívákat pro-kontra érvekkel megvizsgáltad-e stb.
  - Milyen a dolgozat felépítése (logikus, áttekinthető szerkezet, ábrákkal-táblázatokkal-példákkal megfelelően illusztrált), nyelvezetileg (érthető, szabatos megfogalmazások, helyesírási és gépelési hibáktól mentes) rendben van-e stb.
  - Elért eredmények ismertetése, végkövetkeztetések helyessége, továbblépési lehetőségek, amivel esetleg a kutatócsoport másik tagja fog tovább foglalkozni vagy – BSc szakdolgozat esetén – esetleg én magam az MSc.-s tanulmányaim során ...stb.

## 5. Szakdolgozat / Diplomamunka szerkezete

- Nézd meg mindenképpen a „Diplomamunka formai követelményei” részt [itt](#) (MSc. esetén) vagy a „Szakdolgozat formai követelményei” részt [itt](#) (BSc. esetén) de az itt linkelt sablon fájlok úgy vannak kialakítva (BSc. docx / [BSc Latex](#) / MSc.), hogy minden formai követelménynek megfeleljenek és tartalmazza a kötelező fejezeteket. Az alábbi szösszenet csak a hivatalos formátumú leírás elkészítéséhez szeretne nyújtani egy-két támpontot.
- Élesen különüljenek el a már korábban meglévő eredmények a saját munkádtól. Szempont ezzel kapcsolatban, hogy a dolgozat legalább felének, de inkább legalább 60%-ának a saját munkádnak, új eredményeidnek a bemutatása kell, hogy legyen (tehát nem irodalomkutatás). A tématerülettől függően ezek aránya változhat, de első körben arany szabályként erre kb. 2/3 arányt javasolunk.
- Fontos, hogy minden egyes főfejezet egy új oldalon kezdődjön!
- Amikor úgy érzed, hogy készen vagy, akkor pedig ne felejtse el frissíteni a tartalomjegyzéket, keresztshivatózásokat, ábrajegyzéket stb., hogy minden megfelelő-e.

### 5.1. A dolgozat eleji fejezetek

#### Abstract MadLibs!!

This paper presents a \_\_\_\_\_ method for \_\_\_\_\_  
(synonym for new) (sciencey verb)  
the \_\_\_\_\_. Using \_\_\_\_\_, the  
(noun few people have heard of) (something you didn't invent)  
\_\_\_\_\_ was measured to be \_\_\_\_\_ +/- \_\_\_\_\_  
(property) (number) (number)  
\_\_\_\_\_. Results show \_\_\_\_\_ agreement with  
(units) (sexy adjective)  
theoretical predictions and significant improvement over  
previous efforts by \_\_\_\_\_, et al. The work presented  
(Loser)  
here has profound implications for future studies of  
\_\_\_\_\_ and may one day help solve the problem of  
(buzzword)  
\_\_\_\_\_.  
(supreme sociological concern)

Keywords: \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_  
(buzzword) (buzzword) (buzzword)

JORGE CHAM © 2009

WWW.PHDCOMICS.COM

2. ábra: Példa „abstract”<sup>12</sup> :)

#### 5.1.1. Abstract / Kivonat

- Lásd :
  - Ez nagyságrendileg kb. fél oldal legyen.
  - Legyen benne, hogy mi a tématerület, mit csináltál, azt hogyan csináltad és mik az elért eredmények.

<sup>12</sup> Kép forrása: <https://phdcomics.com/comics/archive.php?comid=1121>

### 5.1.2. Tartalomjegyzék, rövidítés-, jelölésjegyzék

- Fűszerezd a szerkezetet tartalomjegyzékkel, szükség esetén rövidítés- vagy jelölésjegyzékkel. Előbbi, értelem szerint, a dokumentum elejére kerüljön, utóbbiak javasolt helye szintén ott vagy esetleg függelékben.

### 5.1.3. Bevezetés

- Ezeket a főbb pontokat tartalmazza:
  - Tématerület ismertetése, a végén leírva, hogy a te témád ehhez hogyan kapcsolódik.
  - A témaválasztás indoklása, mi a motiváció? Mi a szakdolgozatod célja, amit el szeretnél érni vele.
  - Érdemes még a bevezetés utolsó bekezdésében a főbb fejezetekről egy pár soros ismertetőt írni, amiből látszik a szakdolgozatod szerkezete.
- Ennek a fejezetnek a célja, hogy a témában nem jártas személy is képen legyen vele, hogy miről lesz szó a továbbiakban, és megérthesse, hogy mivel foglalkoztál.
- Fontos, hogy a bevezető fejezet legyen az első számozott fejezet. Az ezen fejezet előtti részek (abstract és kivonat, tartalomjegyzék, rövidítésjegyzék, jelölésjegyzék stb.) ne kapjanak számozást!

## 5.2. A BSc Szakdolgozat fejezetei

- Szakdolgozat esetén mindenképpen valamilyen szoftver implementálását kell bemutatni!

### 5.2.1. Felhasználói dokumentáció

- Itt be kell mutatnod az alkalmazásod olyan szinten, hogy a felhasználó használni tudja. Itt lehet elhelyezni képernyőképeket, amelyek segítik a program használatát. Le kell írnod milyen gépigény szükséges, hogyan kell telepíteni és elindítani. Gondolkodj úgy, hogy ha te lennél a felhasználó, aki először találkozik veled, akkor tudnád-e használni ezek alapján a programodat. Érdemes esetleg megkérni egy barátot, hogy tényleg tudja-e használni a programot az általad leírtak alapján. Itt lehet use case-eket írni, hogy bemutasd az egyes funkciókat.

### 5.2.2. Fejlesztői dokumentáció

- Itt olyan szemszögből kell megmutatni a programodat, mintha egy cégnél dolgoznál és a többi fejlesztőnek írnád, ami által megismerhetik az alkalmazás minden fontosabb részletét.
- Architektúra, tervezési döntések indoklással: a döntések meghozatala előtt ismertesd a piacon elérhető megoldásokat, amelyek alternatívaként tudnak szolgálni a te egyedi megoldásodhoz. Világíts rá az alternatív megoldások előnyeire és hátrányaira. Érvelj amellett, hogy az általad választott programozási nyelv / környezet / platform és végső soron a saját szoftvered miért jobb, mint a bemutatott alternatívák.
- Ne tegyetek a dolgozatba óriási forráskódokat, senkinek a fejében nincsen compiler. Szorítkozzatok arra, hogy ha kódot akartok ismertetni, akkor csak a lényeges részeket mutassátok meg és magyarázzátok is el, hogy mi történik ott. Triviális kódsorokat felesleges beletenni a dolgozatba.
- Itt mutasd be a programod rétegeit: adatbázist, táblákat, osztályokat, modulokat, fontosabb függvényeket, algoritmusokat, felhasználói eseteket, hálózati kommunikációt, fejlesztői környezetet. Itt szerepeljenek osztálydiagramok, UML ábrák, de itt lehet beszélni az üzemeltetésről is. Milyen időzített folyamatok vannak, hol van a logolás, használ-e felhő rendszereket stb.
- Az ember hajlamos megismételni ugyanazt, ami kb. a felhasználói dokumentációban szerepelt. Azonban itt ügyljetek arra, hogy ennek a résznek a programozói fogásokról, az egyes felhasznált eszközök szakmai ismertetőjéről kell szólnia. Plusz az elkészített programotok olyan fogásairól, amit kb. úgy mutattok be, mint ha a tényleges fejlesztést utána majd át akarnátok adni egy másik programozónak, és akkor őt ezzel a fejezettel képbe tudnátok hozni:
  - Elmondjátok milyen fizikai eszközöket használtok, azokkal hogyan kommunikáltok,
  - hogy történik az adatok fogadása, továbbítása az egyes eszközök, modulok, web esetén a szerver és kliens oldal között,
  - milyen adatbázist használtok, hogy néz ki az adatszerkezetetek,
  - ha használtok fájlkezelést valamire, azt is írjátok le,

- ha használtak valamilyen olyan programcsomagot, ami nem triviális (értsd: nem ciklusok és elágazások), hanem például egy Arduino irányítására alkalmas Java-s osztálykönyvtárt kezeltek, akkor mutassátok be röviden, és térjete ki azokra a függvényekre, amelyeket itt konkrétan használtak, hogy azok mit is csinálnak,
- mutassátok be, hogyan reagál a program az egyes felhasználói eseményekre stb.
- Tesztelés módja, futási eredmények riportjával és azok kiértékelésével<sup>13</sup>
  - Minden osztályodra legyen tesztelés is. Egység tesztek, integrációs tesztek, rendszer és elfogadási tesztek. Ha nem tudsz automatikus tesztet írni a programodra, akkor csinálj teszt jegyzőkönyvet. Ebben leírod mi a funkció, amit tesztelsz és annak mi az elvárt kimenete, de nagyon sok tesztelő eszköz elérhető manapság. Olyanok is vannak, amelyek felületet képesek tesztelni. Nézz utána!
  - A programod felületének manuális tesztelésénél gondoldj a User eXperience<sup>14</sup> (felhasználói élmény<sup>15</sup>) irányelveire és aszerint gondold át, hogy minden a megfelelő helyen van-e, logikus-e a használhatósága, gondoldj az akadálymentesítésre is stb.

### 5.3. MSc Diplomamunka fejezetei

- Diplomamunka esetén már nem elég egy szoftver készítése, hanem valamilyen tudományos témát kell feldolgozni.
- Mivel itt már rendkívül szerte ágazóak a témák, ezért itt már nem tudunk olyan konkrét tanácsokat adni, mint a BSc-s szakdolgozat esetén. A szakterület irodalmának áttekintése mindenképpen szükséges: a vonatkozó definíciók, alapok ismertetése már követheti is a bevezetés fejezetet. Utána pedig következhet a saját munka bemutatása felhasználói / fejlesztői / kutatói szemszögből ismertetve. Ezt érdemes átbeszélni a témavezetőddel!

### 5.4. A dolgozat végi fejezetek

#### 5.4.1. Összefoglalás (kb. 1 oldal terjedelem)

- Nagyon hasznos tud lenni, ha egy oldalban össze tudod foglalni, hogy miről is szól a dolgozatod, mik az eredményeid. Ez annyiban más, mint a bevezetés, hogy itt feltételezheted, hogy már elolvasták a dolgozatod. Itt újra mutasd be röviden a problémát, amivel foglalkoztál, vezesd végig tömören, hogy mit és hogyan használtál és végül mutasd be, hogy mire jutottál.

#### 5.4.2. Továbbfejlesztési lehetőségek (amennyiben értelmes, kb. 1 oldal terjedelem)

- Ahogy Mark Zuckerberg mondta, egy program sosincs készen, sosem készül el, csak jön a határidő, amire teljesíteni kell valamilyen működő megoldást. Ebbe a fejezetbe írd le, milyen további funkciókat tudsz elképzelni az alkalmazásodhoz. Mit lehetne még hozzáfőjlesztetni, mi az, amit esetleg máshogy is meg lehetett volna valósítani.

#### 5.4.3. Irodalomjegyzék

- A hivatkozásokra példát ennek a dokumentumnak a végén találsz.

#### 5.4.4. Függelékek

- A program használatával, futásával kapcsolatos nagyon részletes infóknak lehet, hogy nem a főszövegben, hanem a dolgozat végén egy appendix / függelék fejezetben van helye. Helyzettől függ.

<sup>13</sup> ELTE IK BSc szakdoga esetén ez is a „Fejlesztői dokumentáció” fejezet része

<sup>14</sup> <https://www.springboard.com/blog/design/ux-design-principles/>

<sup>15</sup> <https://www.newconcept.hu/blog/felhasznaloi-elmany-tervezes-ux-design> és <https://kreativwebdesigntanfolyam.hu/webdesign-blog/mi-kulonbseg-ui-es-ux-kozott>

## 6. Leadás előtti ellenőrzőlista

- ✓ Nevem, neptun-kódom rajta van az első oldalon
- ✓ Témavezetőm adatai (név, tanszék, beosztás) helyes
- ✓ A védés év szerepel a dolgozaton (decemberi leadásnál januárban van a védés)
- ✓ A margók, papírméret helyesen vannak beállítva
- ✓ A szöveg, kép nem lóg bele a margóba egyik oldalon sem
- ✓ A képeken, grafikonon olvasható méretűek a szövegek nyomtatási méretben is
- ✓ Minden képnek, táblázatnak, kódrészletnek van száma és képaláírása és be van hivatkozva a folyó szövegből
- ✓ Minden főfejezet új oldalon kezdődik
- ✓ A szöveg összességében is pofás, nincsenek sehol sem fattyúsorok és árvasorok
- ✓ Nem maradt „TODO” szöveg a dolgozatban
- ✓ A tartalomjegyzék, ábrajegyzék, irodalomjegyzék és egyéb jegyzékek adatai a legfrissebbek

A szöveget helyesírásellenőrzővel is átnéztem helyesírási és gépelési hibák után kutatva.

## 7. Irodalomjegyzék

- [1] RFC 2549: IP over Avian Carriers with Quality of Service. <https://tools.ietf.org/html/rfc2549> Accessed: 2021-04-22.
- [2] A. Paradise, A. Obertas, A. O'Grady, M. Young: The Long Night: Modeling the Climate of Westeros, *Popular Physics*, 52 (12): 1-11. 2019.
- [3] Bullshit Generator. <http://bullshitgenerator.net/> Accessed: 2021-04-22.
- [4] S. Keshav: How to Read a Paper <https://web.stanford.edu/class/ee384m/Handouts/HowtoReadPaper.pdf>
- [5] Engen, Michael; Vatter, Vincent (2021), "Containing all permutations", *American Mathematical Monthly*, **128** (1): 4-24, [doi:10.1080/00029890.2021.1835384](https://doi.org/10.1080/00029890.2021.1835384)
- [6] Anonymous 4chan poster; Houston, Robin; Pantone, Jay; Vatter, Vince: A lower bound on the length of the shortest superpattern. 2018. <https://oeis.org/A180632/a180632.pdf>