

# Teendők listája

<input type="checkbox"/> Milyen tanszék? . . . . .	1
<input type="checkbox"/> "Témebejelentő" header? . . . . .	1
<input type="checkbox"/> Közérthető leírás alatt ezt érti, vagy ez már túl specifikus? . . . . .	3
<input type="checkbox"/> Irodalomjegyzékben jó a citation? . . . . .	4
<input type="checkbox"/> Telefonra ha meg lesz akkor az jöhet ide . . . . .	5
<input type="checkbox"/> Content elrendezése nagy page break nélkül . . . . .	6
<input type="checkbox"/> Hmm ide kéne még valami . . . . .	12
<input type="checkbox"/> Kéne am egy összefoglaló . . . . .	21

Milyen  
tanszék?



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

MÉDIA- ÉS OKTATÁSinFORMATIKAI

TANSZÉK

## Időpont foglaló webes alkalmazás

*Témavezető:*

Dr. Menyhárt László Gábor  
adjunktus

*Szerző:*

Andi Péter  
programtervező informatikus BSc

*Budapest, 2021*

"Témebejelentő"  
header?

A szakdolgozat célja egy időpont foglaló webes alkalmazás létrehozása. Az alkalmazásban vállalkozók (pl.: edzők, magán tanárok) szabad időpontokat hirdethetnek, melyeket ügyfelek lefoglalhatnak. Ez az alkalmazás lehetővé teszi, hogy az egyéni-, kis- és középvállalkozók egyszerűen tudják egyeztetni ügyfeleikkel a munkáikat. Továbbá, a szoftver számon tartja a múltbeli foglaltidőpontokat, melyek így lekérdezhetők, így például a vállalkozó számlázás során egyszerűen meg tudja állapítani, hogy az adott hónapra hány alkalmat vett igénybe egy kliens.

A program két különálló részből áll, egy webes frontendből, amit Javascript-el és hasonló modern technológiákkal valósítok meg és egy backend API-ból melyet C# ASP.NET-ben kivitelezek. A frontend a backenddel http requestekkel kommunikál, a backend pedig egy adatbázist használ az adatok tárolására. A dolgozatomban rámutatok ennek az architektúrának az előnyeire és hátrányaira egy monolitikus MVC alapú webes alkalmazással szemben.

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>3</b>
1.1. Motiváció . . . . .	3
1.2. Megvalósítandó alkalmazás leírása . . . . .	3
1.3. Kedvhozó az architektúrához . . . . .	4
<b>2. Felhasználói dokumentáció</b>	<b>5</b>
2.1. Rendszerkövetelmények . . . . .	5
2.2. Telepítés . . . . .	5
2.2.1. Telepítés Dockerrel . . . . .	5
2.2.2. Telepítés Docker nélkül . . . . .	8
2.3. Funkciók leírása . . . . .	8
2.4. Használat . . . . .	10
2.4.1. Ügyfeleknek . . . . .	10
2.4.2. Vállalkozóknak . . . . .	10
2.5. Felsorolások . . . . .	10
2.5.1. Szoros térközű felsorolások . . . . .	11
2.6. Képek, ábrák . . . . .	12
2.6.1. Képek szegélyezése . . . . .	12
2.6.2. Képek csoportosítása . . . . .	13
2.7. Táblázatok . . . . .	14
2.7.1. Sorok és oszlopok egyesítése . . . . .	14
2.7.2. Több oldalra átnyúló táblázatok . . . . .	14
<b>3. Fejlesztői dokumentáció</b>	<b>17</b>
3.1. Tételek, definíciók, megjegyzések . . . . .	17
3.1.1. Egyenletek, matematika . . . . .	18

3.2. Forráskódok . . . . .	19
3.2.1. Algoritmusok . . . . .	20
4. Összegzés	21
A. Szimulációs eredmények	22
Irodalomjegyzék	24
Ábrajegyzék	25
Táblázatjegyzék	26
Forráskódjegyzék	27

# 1. fejezet

## Bevezetés

### 1.1. Motiváció

Szakdolgozatom célja egy időpont foglaló webes alkalmazás létrehozása. A motivációt unokatestvérem adta, aki személyi edzőként dolgozik. A munkájához elengedhetetlen, hogy időpontot egyeztessen ügyfeivel. Ezt üzenetváltásokkal tette, viszont, ha valaki lemondott egy időpontot, akkor utána arra a szabad időpontra más ügyfelet körülményes volt találni a platform miatt. Arról nem is beszélve, hogy hónap végén a számlakiállításához így nem volt egy konkrét listája, amit egyszerűen be tudott volna vinni a számlázó rendszerébe.

Én programozásban mindig is webes alkalmazások fejlesztését élveztem a legjobban, így amikor felvetette az ötletet, hogy lehetne egy időpont foglaló alkalmazást csinálni, le is csaptam rá. Ezzel nem csak az egész eddigi összes webes tudásomat tesztelhetem és fejleszthetem, hanem segíthetek is unokatestvéremnek, aki nagyon sokat segített rajtam is.

### 1.2. Megvalósítandó alkalmazás leírása

Közérthető leírás alatt ezt érti, vagy ez már túl specifikus?

Az alkalmazásnak két fő felhasználói köre van, a vállalkozók és az ügyfelek. Az ügyfelek tudnak a vállalkozók között böngészni, egyes vállalkozók időpontjait megnézni, szűrni és lefoglalni. Megnézhetik a lefoglalt időpontjaikat, melyeket lemondhatnak.

A vállalkozók létrehozhatnak kategóriákat (pl.: személyi edzés, angol korrepetálás), melynek megadhatnak árat, maximum résztvevő számot és hogy publikus-e az esemény, vagy csak megadott ügyfelek láthatják. Ez azért fontos, mert például unokatestvérem hétvégére csak családtagoknak vagy közeli ismerősöknek tartott edzéseket, az alkalmazásban ezért kell tudni szabályozni a láthatóságát a kategóriáknak. A vállalkozók időpont hirdetésnél választhatnak egy kategóriát és kezdő és vég időpontot, esetleg módosíthatják a résztvevő limitet. A kategóriákat, időpontokat és vállalkozói profilt lehet szerkeszteni. A vállalkozó le tudja kérdezni, kategóriákra és időtartamra szűrhetően, hogy egy ügyfél melyik kategóriából hány időpontot foglalt, ezek mennyibe kerültek összesen és generálhat egy pdf formátumú számlát.

### 1.3. Kedvhozó az architektúrához

A dolgozatomban nem csak a programra koncentráltam, hanem, hogy a mögöttes architektúra és kód minőségi és bővíthető legyen.

Irodalomjegyzékben jó a citation?

A backendem Uncle Bob Clean Architecture[1] elvén alapuló objektum orientált kód. Ezzel moduláris, elkülönített hatáskörű osztályokból áll a REST API-om, mellyel a Dependency Inversion Principle miatt egyszerűen és hatékonyan unit- és integrációs tesztelhető az alkalmazás.

A frontendemen React.js-t<sup>1</sup> használom Typescript-el, e miatt erős fordítási idejű garanciát kapok, hogy a kódom helyes. Továbbá a Typescript erős típusrendszere miatt a megjelenítés mögött funkcionális paradigmájú kód van. Ez azt jelenti, hogy nincs destruktív értékadás, összeg típusokkal és egy saját aszinkron Result monád típus miatt nem kivételeket kezelek, hanem típus szintű konstrukciókkal garantálom, hogy minden hiba megfelelően le legyen kezelve és programozói hibából ne lehessen inkonzisztens állapotban levő adathoz hozzáférni.

---

<sup>1</sup>React.js - <https://reactjs.org/>

## 2. fejezet

# Felhasználói dokumentáció

### 2.1. Rendszerkövetelmények

Szerver oldalon: Windows 10 vagy Linux operációs rendszer, 2GB RAM, legalább 5GB tárhely az adatoknak, port nyitási lehetőség.

Telefonra ha  
meg lesz akkor  
az jöhet ide

Kliens oldalon: Legalább Chrome 90, Firefox 88, Edge 90, ezek mind asztali számítógépen, legalább 1280x720-as képernyő felbontással.

### 2.2. Telepítés

Az alkalmazást legegyszerűbben Docker<sup>2</sup> segítségével lehet telepíteni. A Docker egy konténerizációs technológia, amely megkönnyíti az alkalmazások kihelyezését. Az alkalmazások 'konténerekbe' csomagolódnak minden függőségükkel, ezen konténereket utána egy egységként lehet futtatni, nem kell a felhasználó rendszerére egyesével a program futásához felállítani a környezetet.

#### 2.2.1. Telepítés Dockerrel

A Dockeres telepítéshez szükséges a Docker<sup>3</sup> és Docker Compose<sup>4</sup> telepítése.

A fő mappában megtalálható *docker-compse.yml* fájlban találhatók meg a konténerek konfigurációi. Három konténerből áll, egy MariaDB <sup>5</sup> adatbázisból, a

<sup>2</sup><https://www.docker.com/>

<sup>3</sup><https://www.docker.com/get-started>

<sup>4</sup><https://docs.docker.com/compose/install/>


<sup>5</sup><https://mariadb.org/>



backend REST API-ból és a frontendből. A yml fájlban a következő konfigurációs lehetőségek a 2.1 táblázatban találhatók.

A Docker konténereket úgy találták ki, hogy eldobhatóak legyenek, azaz ki lehessen őket törölni és újra futtatni és ugyan úgy működjenek. Az adatok perzisztálását így *volume*-okkal oldhatjuk meg, jelen esetben az adatbázis és a profilképeket a host gép lokális mappáiban tároljuk.

Content  
elrendezése  
nagy page  
break nélkül



Konfigurációs változó	Alap érték	Megjegyzés
db		
MYSQL_ROOT_PASSWORD	kebab	MariaDB adatbázis root felhasználójának jelszava
volumes	./db_data:/var/lib/mysql	MariaDB adatbázis perzisztens tárolása a lokális db_data mappában
ports	3306:3306	A konténer 3306-as portja a hoston az 3306-es portra forwardolása
backend		
IWA_CorsAllowUrls	http://127.0.0.1:8100	Vesszővel elválasztva az engedélyezett publikus frontend url-ek. Pl.: <a href="https://andipeter.me">https://andipeter.me</a>
IWA_MYSQL_HOST	db	MariaDB adatbázis host neve
IWA_MYSQL_PORT	3306	MariaDB adatbázis portja
IWA_MYSQL_DB	iwa	MariaDB adatbázison belül használandó adatbázis
IWA_MYSQL_USER	root	MariaDB adatbázis felhasználója
IWA_MYSQL_PASS	kebab	MariaDB adatbázis felhasználójának jelszava
volumes	./avatars:/app/AvatarData	A profilképek perzisztens tárolása a lokális avatars mappában
ports	5000:80	A konténer 80-as portja a hoston az 5000-es portra forwardolása
frontend		
API_URL	http://127.0.0.1:5000	A backend publikus elérési url-je. Pl.: <a href="https://andipeter.me/api/">https://andipeter.me/api/</a>
ports	8100:80	A konténer 80-as portja a hoston az 8100-es portra forwardolása

2.1. táblázat. Konfigurációs változók beállításai

Az alkalmazást ez után a *docker-compose up* paranccsal indíthatjuk el. Első futtatásra ez eltarthat pár percre, mert a Dockernek le kell töltenie a megfelelő alap konténereket az internetről és utána létre kell hozni ezeket a konténereket a forráskódból.

További docker-compose parancsok a docker-compose dokumentációjábanban<sup>6</sup> találhatók.

### 2.2.2. Telepítés Docker nélkül

Az alkalmazás futtatásához szükség lesz egy MariaDB szerverre és azon belül egy *iwa* nevű adatbázisra. Az alkalmazás Linux és Windows rendszereken is futhat, ehhez a megfelelő backend fájl futtatása szükséges.

A backend futtatása parancssorból az *IWA\_Backend.API* futtatható fájlal lehet. A konfigurációja az *appsettings.json* fájlban található, kitöltése a 2.1 táblázat alapján történik. A backend így a host 80-as portján fog futni. Ezt a `-urls=http://localhost:5001/` konzoli argumentummal lehet megváltoztatni, ebben az esetben az 5001-es porton futna az alkalmazás.

A frontend statikus HTML, JS és CSS fájlokból áll, ezt például Apache<sup>7</sup> vagy Nginx<sup>8</sup> szerverekkel, vagy más hasonló webhost szolgáltatásokkal lehet kitelepíteni. A frontend konfigurációja a mappájában a *config.js* fájlban történik, kitöltési útmutató a 2.1 táblázatban található.

## 2.3. Funkciók leírása

Az alkalmazásban lehet regisztrálni ügyfél vagy vállalkozóként. Az oldalt lehet bejelentkezve vagy bejelentkezés nélkül böngészni.

A vállalkozók létrehozhatnak kategóriákat. A kategória effektíve egy időpont típus, például személyi edzés. A kategóriák megegyeszerűsítik az új időpontok létrehozását, mert a különböző időpontok közti azonos adatokat enkapszulálják, az időpontnál így csak az időpont specifikus adatokat kell megadni. Egy kategóriának lehet egy leírása, ára, ajánlott max résztvevő száma és láthatósága. Az ajánlott max

---

<sup>6</sup><https://docs.docker.com/compose/reference/>

<sup>7</sup><https://httpd.apache.org/>

<sup>8</sup><https://www.nginx.com/>

résztevőszám azt jelenti, hogy egy új időpont létrehozásánál alapból ez a szám lesz a max résztvevők mezőben, viszont ettől el lehet térni időpontról időpontra, például egy csoportos edzésre a Margit szigeten többen jöhetnek mint a Hősök tereire.

Egy kategória láthatósága a következőt jelenti. Ha nyílt egy esemény, akkor bárki láthatja, bárki jelentkezhet rá. Ha egy esemény nem nyílt, akkor csak azok az emberek láthatják és jelentkezhetnek rá, akik engedélyezett résztvevőként fel lettek véve a kategóriára. Ennek az a szerepe, hogy például egy Családi edzésre hétvégén ne tudjon mindenki jelentkezni, csak az előre felvett családtagok. Vagy például egy kedvezményes árazású időpontnak más lehet a kategóriája.

Kategóriákat nem lehet törölni, abból az okból, hogy akkor az összes hozzá tartozó időpont is törlődne, ezzel múltbéli időpontok adatai elvesznének.

Új időpont hirdetésénél az időponthoz kell választani egy kategóriát, kezdő és vég időpontot. Opcionálisan meg lehet változtatni a max résztvevő számot. Van lehetőség alapból felvenni ügyfeleket az időpontra, például ha a vállalkozó már előre leegyeztetett egy időpontot de még nem írta ki az alkalmazáson, akkor az ügyfélnek nem kell bejelentkeznie és lefoglalni az időpontot.

Időpont szerkesztésnél a vállalkozónak van lehetősége változtatni egy időpont összes értékén. A kategórián például azért változtathat, mert Angol óra helyett Német órát tartott az ügyfélnek, vagy Páros edzés helyett Személyi Edzést, mert közbe jött valami. Lehet az időpontra jelentkezett felhasználókat is módosítani, lejelentkeztetni és felvenni ügyfeleket, akár az időpont után is. Például valaki lemondott egy edzést és beugrott helyette valaki más, a nap végén pedig így korrektül tudja adminisztrálni ezt a vállalkozó.

A számlázás funkciónál a vállalkozók adott felhasználók lefoglalt időpontjaiból tudnak számlát generálni egy időszakra, például Április 1 és 30 között. Ez a számla jelenlegi formájában nem minősül NAV által elfogadott számlának, viszont a vállalkozónak nagyon jó segítség, hogy a saját számlázó szoftverébe (pl.: számlázz.hu) miről írjon számlát. Az alkalmazásba azért se került online fizetési lehetőség vagy számlázz.hu integráció, mert a valóságban az időpontokon kívül mást is tartalmazni szokott a számla (pl.: edzőterem bérlet, edzésterv) és ezekre akkor ezen felül egy külön számlát kéne kiállítania a vállalkozónak.

## 2.4. Használat

### 2.4.1. Ügyfeleknek

### 2.4.2. Vállalkozóknak

Lorem ipsum dolor sit amet  $\mathbb{N}$ , consectetur adipiscing elit. Duis nibh leo, dapibus in elementum nec, aliquet id sem. Suspendisse potenti. Nullam sit amet consectetur nibh. Donec scelerisque varius turpis at tincidunt. Cras a diam in mauris viverra vehicula. Vivamus mi odio, fermentum vel arcu efficitur, lacinia viverra nibh. Aliquam aliquam ante mi, vel pretium arcu dapibus eu. Nulla finibus ante vel arcu tincidunt, ut consectetur ligula finibus. Mauris mollis lectus sed ipsum bibendum, ac ultrices erat dictum. Suspendisse faucibus euismod lacinia  $\mathbb{Z}$ .

## 2.5. Felsorolások

Etiam vel odio ante. Etiam pulvinar nibh quis massa auctor congue. Pellentesque quis odio vitae sapien molestie vestibulum sit amet et quam. Pellentesque vel dui eget enim hendrerit finibus at sit amet libero. Quisque sollicitudin ultrices enim, nec porta magna imperdiet vitae. Cras condimentum nunc dui, eget molestie nunc accumsan vel.

- Fusce in aliquet neque, in pretium sem.
- Donec tincidunt tellus id lectus pretium fringilla.
- Nunc faucibus, erat pretium tempus tempor, tortor mi fringilla neque, ac congue ex dui vitae mauris.

Donec dapibus sodales ante, at scelerisque nunc laoreet sit amet. Mauris porttitor tincidunt neque, vel ullamcorper neque pulvinar et. Integer eu lorem euismod, faucibus lectus sed, accumsan felis. Nunc ornare mi at augue vulputate, eu venenatis magna mollis. Nunc sed posuere dui, et varius nulla. Sed mollis nibh augue, eget scelerisque eros ornare nec.

1. Donec pretium et quam a cursus. Ut sollicitudin tempus urna et mollis.

2. Aliquam et aliquam turpis, sed fermentum mauris. Nulla eget ex diam.

3. Donec eget tellus pharetra, semper neque eget, rutrum diam Step 1.

Praesent porta, metus eget eleifend consequat, eros ligula eleifend ex, a pellentesque mi est vitae urna. Vivamus turpis nunc, iaculis non leo eget, mattis vulputate tellus. Maecenas rutrum eros sem, pharetra interdum nulla porttitor sit amet. In vitae viverra ante. Maecenas sit amet placerat orci, sed tincidunt velit. Vivamus mattis, enim vel suscipit elementum, quam odio venenatis elit<sup>9</sup>, et mollis nulla nunc a risus. Praesent purus magna, tristique sed lacus sit amet, convallis malesuada magna.

**Vestibulum venenatis** malesuada enim, ac auctor erat vestibulum et. Phasellus id purus a leo suscipit accumsan.

**Orci varius natoque** penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam interdum rhoncus nisl, vel pharetra arcu euismod sagittis. Vestibulum ac turpis auctor, viverra turpis at, tempus tellus.

**Morbi dignissim** erat ut rutrum aliquet. Nulla eu rutrum urna. Integer non urna at mauris scelerisque rutrum sed non turpis.

### 2.5.1. Szoros térközű felsorolások

Phasellus ultricies, sapien sit amet ultricies placerat, velit purus viverra ligula, id consequat ipsum odio imperdiet enim:

1. Maecenas eget lobortis leo.
2. Donec eget libero enim.
3. In eu eros a eros lacinia maximus ullamcorper eget augue.

In quis turpis metus. Proin maximus nibh et massa eleifend, a feugiat augue porta. Sed eget est purus. Duis in placerat leo. Donec pharetra eros nec enim convallis:

- Pellentesque odio lacus.
- Maximus ut nisl auctor.

---

<sup>9</sup>Phasellus faucibus varius purus, nec tristique enim porta vitae.

- Sagittis vulputate lorem.

Hmm ide kéne  
még valami

- 

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed lorem libero, dignissim vitae gravida a, ornare vitae est.

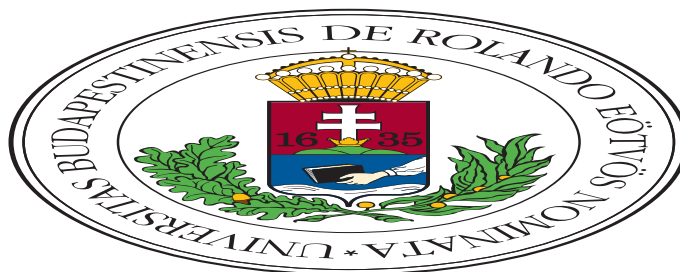
**Cras maximus** massa commodo pellentesque viverra.

**Morbi sit** amet ante risus. Aliquam nec sollicitudin mauris

**Ut aliquam rhoncus sapien** luctus viverra arcu iaculis posuere

## 2.6. Képek, ábrák

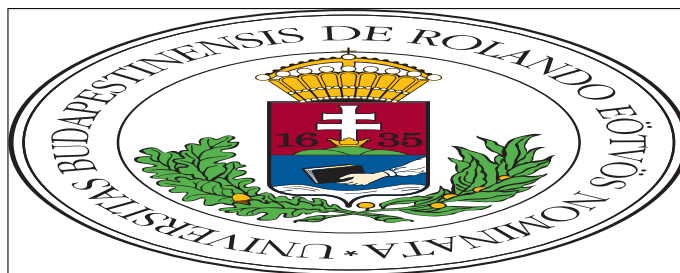
Aliquam vehicula luctus mi a pretium. Nulla quam neque, maximus nec velit in, aliquam mollis tortor. Aliquam erat volutpat. Curabitur vitae laoreet turpis. Integer id diam ligula. Nulla sodales purus id mi consequat, eu venenatis odio pharetra. Cras a arcu quam. Suspendisse augue risus, pulvinar a turpis et, commodo aliquet turpis. Nulla aliquam scelerisque mi eget pharetra. Mauris sed posuere elit, ac lobortis metus. Proin lacinia sit amet diam sed auctor. Nam viverra orci id sapien sollicitudin, a aliquam lacus suscipit, Figure 2.1:



2.1. ábra. Quisque ac tincidunt leo

### 2.6.1. Képek szegélyezése

Ut aliquet nec neque eget fermentum. Cras volutpat tellus sed placerat elementum. Quisque neque dui, consectetur nec finibus eget, blandit id purus. Nam eget ipsum non nunc placerat interdum.



2.2. ábra. Quisque ac tincidunt leo

## 2.6.2. Képek csoportosítása

In non ipsum fermentum urna feugiat rutrum a at odio. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nulla tincidunt mattis nisl id suscipit. Sed bibendum ac felis sed volutpat. Nam pharetra nisi nec facilisis faucibus. Aenean tristique nec libero non commodo. Nulla egestas laoreet tempus. Nunc eu aliquet nulla, quis vehicula dui. Proin ac risus sodales, gravida nisi vitae, efficitur neque, Figure 2.3:



(a) Vestibulum quis mattis urna


(b) Donec hendrerit quis dui sit amet  
venenatis

2.3. ábra. Aenean porttitor mi volutpat massa gravida

Nam et nunc eget elit tincidunt sollicitudin. Quisque ligula ipsum, tempor vitae tortor ut, commodo rhoncus diam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Phasellus vehicula quam dui, eu convallis metus porta ac.



## 2.7. Táblázatok

Nam magna ex, euismod nec interdum sed, sagittis nec leo. Nam blandit massa bibendum mattis tristique. Phasellus tortor ligula, sodales a consectetur vitae, placerat vitae dolor. Aenean consequat in quam ac mollis.

Phasellus tortor	Aenean consequat
<i>Sed malesuada</i>	Aliquam aliquam velit in convallis ultrices.
<i>Purus sagittis</i>	Quisque lobortis eros vitae urna lacinia euismod.
<i>Pellentesque</i>	Curabitur ac lacus pellentesque, eleifend sem ut, placerat enim. Ut auctor tempor odio ut dapibus.

2.2. táblázat. Maecenas tincidunt non justo quis accumsan

### 2.7.1. Sorok és oszlopok egyesítése

Mauris a dapibus lectus. Vestibulum commodo nibh ante, ut maximus magna eleifend vel. Integer vehicula elit non lacus lacinia, vitae porttitor dolor ultrices. Vivamus gravida faucibus efficitur. Ut non erat quis arcu vehicula lacinia. Nulla felis mauris, laoreet sed malesuada in, euismod et lacus. Aenean at finibus ipsum. Pellentesque dignissim elit sit amet lacus congue vulputate.

Quisque	Suspendisse		Aliquam		Vivamus	
	Proin	Nunc	Proin	Nunc	Proin	Nunc
Leo	2,80 MB	100%	232 KB	8,09%	248 KB	8,64%
Vel	9,60 MB	100%	564 KB	5,74%	292 KB	2,97%
Auge	78,2 MB	100%	52,3 MB	66,88%	3,22 MB	4,12%

2.3. táblázat. Vivamus ac arcu fringilla, fermentum neque sed, interdum erat.

Mauris bibendum mauris vitae enim mollis, et eleifend turpis aliquet.

### 2.7.2. Több oldalra átnyúló táblázatok

Nunc porta placerat leo, sit amet porttitor dui porta molestie. Aliquam at fermentum mi. Maecenas vitae lorem at leo tincidunt volutpat at nec tortor. Vivamus

semper lacus eu diam laoreet congue. Vivamus in ipsum risus. Nulla ullamcorper finibus mauris non aliquet. Vivamus elementum rhoncus ex ut porttitor.

Praesent aliquam mauris enim	
<i>Suspendisse potenti</i>	<i>Lorem ipsum dolor sit amet</i>
<i>Praesent</i>	Nulla ultrices et libero sit amet fringilla. Nunc scelerisque ante tempus sapien placerat convallis.
<i>Luctus</i>	Integer hendrerit erat massa, non hendrerit risus convallis at. Curabitur ultrices, justo in imperdiet condimentum, neque tortor luctus enim, luctus posuere massa erat vitae nibh.
<i>Egestas</i>	Duis fermentum feugiat augue in blandit. Mauris a tempor felis. Pellentesque ultricies tristique dignissim. Pellentesque aliquam semper tristique. Nam nec egetas dolor. Vestibulum id elit quis enim fringilla tempor eu a mauris. Aliquam vitae lacus tellus. Phasellus mauris lectus, aliquam id leo eget, auctor dapibus magna. Fusce lacinia felis ac elit luctus luctus.
<i>Dignissim</i>	Praesent aliquam mauris enim, vestibulum posuere massa facilisis in. Suspendisse potenti. Nam quam purus, rutrum eu augue ut, varius vehicula tellus. Fusce dui diam, aliquet sit amet eros at, sollicitudin facilisis quam. Phasellus tempor metus vel augue gravida pretium. Proin aliquam aliquam blandit. Nulla id tempus mi. Fusce in aliquam tortor.
<i>Pellentesque</i>	Donec felis nibh, imperdiet a arcu non, vehicula gravida nibh. Quisque interdum sapien eu massa commodo, ac elementum felis faucibus.

<i>Suspendisse potenti</i>	<i>Lorem ipsum dolor sit amet</i>
<i>Molestie</i>	Cras ullamcorper tellus et auctor ultricies. Maecenas tincidunt euismod lectus nec venenatis. Suspendisse potenti. Pellentesque pretium nunc ut euismod cursus. Nam venenatis condimentum quam. Curabitur suscipit efficitur aliquet. Interdum et malesuada fames ac ante ipsum primis in faucibus.
<i>Vivamus semper</i>	In purus purus, faucibus eu libero vulputate, tristique sodales nunc. Nulla ut gravida dolor. Fusce vel pellentesque mi, vel efficitur eros. Nunc vitae elit tellus. Sed vestibulum auctor consequat.
<i>Condimentum</i>	Nulla scelerisque, leo et facilisis pretium, risus enim cursus turpis, eu suscipit ipsum ipsum in mauris. Praesent eget pulvinar ipsum, suscipit interdum nunc. Nam varius massa ut justo ullamcorper sollicitudin. Vivamus facilisis suscipit neque, eu fermentum risus. Ut at mi mauris.

2.4. táblázat. Praesent ullamcorper consequat tellus ut eleifend

## 3. fejezet

# Fejlesztői dokumentáció

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis nibh leo, dapibus in elementum nec, aliquet id sem. Suspendisse potenti. Nullam sit amet consectetur nibh. Donec scelerisque varius turpis at tincidunt.

### 3.1. Tételek, definíciók, megjegyzések

**1. Definíció.** *Mauris tristique sollicitudin ultrices. Etiam tristique quam sit amet metus dictum imperdiet. Nunc id lorem sed nisl pulvinar aliquet vitae quis arcu. Morbi iaculis eleifend porttitor.*

Maecenas rutrum eros sem, pharetra interdum nulla porttitor sit amet. In vitae viverra ante. Maecenas sit amet placerat orci, sed tincidunt velit. Vivamus mattis, enim vel suscipit elementum, quam odio venenatis elit, et mollis nulla nunc a risus. Praesent purus magna, tristique sed lacus sit amet, convallis malesuada magna. Phasellus faucibus varius purus, nec tristique enim porta vitae.

**1. Tétel.** *Nulla finibus ante vel arcu tincidunt, ut consectetur ligula finibus. Mauris mollis lectus sed ipsum bibendum, ac ultrices erat dictum. Suspendisse faucibus euismod lacinia. Etiam vel odio ante.*

*Bizonyítás.* Etiam pulvinar nibh quis massa auctor congue. Pellentesque quis odio vitae sapien molestie vestibulum sit amet et quam. Pellentesque vel dui eget enim hendrerit finibus at sit amet libero. Quisque sollicitudin ultrices enim, nec porta magna imperdiet vitae. Cras condimentum nunc dui. □

Donec dapibus sodales ante, at scelerisque nunc laoreet sit amet. Mauris porttitor tincidunt neque, vel ullamcorper neque pulvinar et. Integer eu lorem euismod, faucibus lectus sed, accumsan felis.

**Emlékeztető.** *Nunc ornare mi at augue vulputate, eu venenatis magna mollis. Nunc sed posuere dui, et varius nulla. Sed mollis nibh augue, eget scelerisque eros ornare nec. Praesent porta, metus eget eleifend consequat, eros ligula eleifend ex, a pellentesque mi est vitae urna. Vivamus turpis nunc, iaculis non leo eget, mattis vulputate tellus.*

Fusce in aliquet neque, in pretium sem. Donec tincidunt tellus id lectus pretium fringilla. Nunc faucibus, erat pretium tempus tempor, tortor mi fringilla neque, ac congue ex dui vitae mauris. Donec pretium et quam a cursus.

**Megjegyzés.** *Aliquam vehicula luctus mi a pretium. Nulla quam neque, maximus nec velit in, aliquam mollis tortor. Aliquam erat volutpat. Curabitur vitae laoreet turpis. Integer id diam ligula.*

Ut sollicitudin tempus urna et mollis. Aliquam et aliquam turpis, sed fermentum mauris. Nulla eget ex diam. Donec eget tellus pharetra, semper neque eget, rutrum diam.

### 3.1.1. Egyenletek, matematika

Duis suscipit ipsum nec urna blandit,  $2 + 2 = 4$  pellentesque vehicula quam fringilla. Vivamus euismod, lectus sit amet euismod viverra, dolor metus consequat sapien, ut hendrerit nisl nulla id nisi. Nam in leo eu quam sollicitudin semper a quis velit.

$$a^2 + b^2 = c^2$$

Phasellus mollis, elit sed convallis feugiat, dolor quam dapibus nibh, suscipit consectetur lacus risus quis sem. Vivamus scelerisque porta odio, vitae euismod dolor accumsan ut.

In mathematica, identitatem Euleri (equation est scriptor vti etiam notum) sit aequalitatem Equation 3.1:

$$e^{i \times \pi} + 1 = 0 \tag{3.1}$$

## 3.2. Forráskódok

Nulla sodales purus id mi consequat, eu venenatis odio pharetra. Cras a arcu quam. Suspendisse augue risus, pulvinar a turpis et, commodo aliquet turpis. Nulla aliquam scelerisque mi eget pharetra. Mauris sed posuere elit, ac lobortis metus. Proin lacinia sit amet diam sed auctor. Nam viverra orci id sapien sollicitudin, a aliquam lacus suscipit. Quisque ac tincidunt leo Code 3.1 and 3.2:

```
1 #include <stdio>
2
3 int main()
4 {
5     int c;
6     std::cout << "Hello World!" << std::endl;
7
8     std::cout << "Press any key to exit." << std::endl;
9     std::cin >> c;
10
11     return 0;
12 }
```

3.1. forráskód. Hello World in C++

```
1 using System;
2 namespace HelloWorld
3 {
4     class Hello
5     {
6         static void Main()
7         {
8             Console.WriteLine("Hello World!");
9
10            Console.WriteLine("Press any key to exit.");
11            Console.ReadKey();
12        }
13    }
14 }
```

3.2. forráskód. Hello World in C#

### 3.2.1. Algoritmusok

A general Interval Branch and Bound algorithm is shown in Algorithm 1. One of the following selection rules is applied in Step 3.

Példa forrása: Acta Cybernetica (ez egy link).

---

**Algoritmus 1** A general interval B&B algorithm

---

**Funct** IBB( $S, f$ )

```

1: Set the working list  $\mathcal{L}_W := \{S\}$  and the final list  $\mathcal{L}_Q := \{\}$ 
2: while (  $\mathcal{L}_W \neq \emptyset$  ) do
3:   Select an interval  $X$  from  $\mathcal{L}_W$  Selection rule
4:   Compute  $lb f(X)$  Bounding rule
5:   if  $X$  cannot be eliminated then Elimination rule
6:     Divide  $X$  into  $X^j$ ,  $j = 1, \dots, p$ , subintervals Division rule
7:     for  $j = 1, \dots, p$  do
8:       if  $X^j$  satisfies the termination criterion then Termination rule
9:         Store  $X^j$  in  $\mathcal{L}_W$ 
10:      else
11:        Store  $X^j$  in  $\mathcal{L}_W$ 
12:      end if
13:    end for
14:  end if
15: end while
16: return  $\mathcal{L}_Q$ 

```

---

## 4. fejezet

# Összegzés

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu egestas mauris. Quisque nisl elit, varius in erat eu, dictum commodo lorem. Sed commodo libero et sem laoreet consectetur. Fusce ligula arcu, vestibulum et sodales vel, venenatis at velit. Aliquam erat volutpat. Proin condimentum accumsan velit id hendrerit. Cras egestas arcu quis felis placerat, ut sodales velit malesuada. Maecenas et turpis eu turpis placerat euismod. Maecenas a urna viverra, scelerisque nibh ut, malesuada ex.

Aliquam suscipit dignissim tempor. Praesent tortor libero, feugiat et tellus porttitor, malesuada eleifend felis. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam eleifend imperdiet lorem, sit amet imperdiet metus pellentesque vitae. Donec nec ligula urna. Aliquam bibendum tempor diam, sed lacinia eros dapibus id. Donec sed vehicula turpis. Aliquam hendrerit sed nulla vitae convallis. Etiam libero quam, pharetra ac est nec, sodales placerat augue. Praesent eu consequat purus.

Kéne am egy  
összefoglaló



## A. függelék

### Szimulációs eredmények

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque facilisis in nibh auctor molestie. Donec porta tortor mauris. Cras in lacus in purus ultricies blandit. Proin dolor erat, pulvinar posuere orci ac, eleifend ultrices libero. Donec elementum et elit a ullamcorper. Nunc tincidunt, lorem et consectetur tincidunt, ante sapien scelerisque neque, eu bibendum felis augue non est. Maecenas nibh arcu, ultrices et libero id, egestas tempus mauris. Etiam iaculis dui nec augue venenatis, fermentum posuere justo congue. Nullam sit amet porttitor sem, at porttitor augue. Proin bibendum justo at ornare efficitur. Donec tempor turpis ligula, vitae viverra felis finibus eu. Curabitur sed libero ac urna condimentum gravida. Donec tincidunt neque sit amet neque luctus auctor vel eget tortor. Integer dignissim, urna ut lobortis volutpat, justo nunc convallis diam, sit amet vulputate erat eros eu velit. Mauris porttitor dictum ante, commodo facilisis ex suscipit sed.

Sed egestas dapibus nisl, vitae fringilla justo. Donec eget condimentum lectus, molestie mattis nunc. Nulla ac faucibus dui. Nullam a congue erat. Ut accumsan sed sapien quis porttitor. Ut pellentesque, est ac posuere pulvinar, tortor mauris fermentum nulla, sit amet fringilla sapien sapien quis velit. Integer accumsan placerat lorem, eu aliquam urna consectetur eget. In ligula orci, dignissim sed consequat ac, porta at metus. Phasellus ipsum tellus, molestie ut lacus tempus, rutrum convallis elit. Suspendisse arcu orci, luctus vitae ultricies quis, bibendum sed elit. Vivamus at sem maximus leo placerat gravida semper vel mi. Etiam hendrerit sed massa ut lacinia. Morbi varius libero odio, sit amet auctor nunc interdum sit amet.

Aenean non mauris accumsan, rutrum nisi non, porttitor enim. Maecenas vel tortor ex. Proin vulputate tellus luctus egestas fermentum. In nec lobortis risus, sit amet tincidunt purus. Nam id turpis venenatis, vehicula nisl sed, ultricies nibh. Suspendisse in libero nec nisi tempor vestibulum. Integer eu dui congue enim venenatis lobortis. Donec sed elementum nunc. Nulla facilisi. Maecenas cursus id lorem et finibus. Sed fermentum molestie erat, nec tempor lorem facilisis cursus. In vel nulla id orci fringilla facilisis. Cras non bibendum odio, ac vestibulum ex. Donec turpis urna, tincidunt ut mi eu, finibus facilisis lorem. Praesent posuere nisl nec dui accumsan, sed interdum odio malesuada.

# Irodalomjegyzék

- [1] Robert C. Martin. *The Clean Architecture*. Aug. 2012. URL: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html> (visited on 04/14/2021).

# Ábrák jegyzéke

2.1. Quisque ac tincidunt leo . . . . .	12
2.2. Quisque ac tincidunt leo . . . . .	13
2.3. Aenean porttitor mi volutpat massa gravida . . . . .	13

# Táblázatok jegyzéke

2.1. Konfigurációs változók beállításai . . . . .	7
2.2. Maecenas tincidunt non justo quis accumsan . . . . .	14
2.3. Rövid cím a táblázatjegyzékbe . . . . .	14
2.4. Praesent ullamcorper consequat tellus ut eleifend . . . . .	16

# Forráskódjegyzék

3.1. Hello World in C++ . . . . .	19
3.2. Hello World in C# . . . . .	19