



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

PROGRAMOZÁSI NYELVEK ÉS FORDÍTÓPROGRAMOK

TANSZÉK

Táblázatkezelő szoftver implementálása Haskell nyelven

Supervisor:

Dr. Kaposi Ambrus

egyetemi docens

Author:

Széles Márk

programtervező informatikus BSc

Budapest, 2021

This page should be the original Thesis Topic Declaration.

Tartalomjegyzék

1	Bevezetés	3
2	Felhasználói dokumentáció	5
2.1	Felsorolások	5
2.1.1	Szoros térközű felsorolások	6
2.2	Képek, ábrák	7
2.2.1	Képek szegélyezése	7
2.2.2	Képek csoportosítása	8
2.3	Táblázatok	9
2.3.1	Sorok és oszlopok egyesítése	9
2.3.2	Több oldalra átnyúló táblázatok	10
3	Fejlesztői dokumentáció	12
3.1	A fejlesztői dokumentáció felépítése	12
3.2	A szoftver felépítése	12
3.2.1	Felhasznált technológiák összefoglalása	12
3.2.2	A globális állapot	13
3.2.3	Programkomponensek és modulszerkezet	14
3.3	A program komponenseinek részletes leírása	15
3.3.1	A főprogram	15
4	Összegzés	16
A	Szimulációs eredmények	17
	Bibliography	19
	List of Figures	20

List of Tables	21
List of Codes	22

fejezet 1

Bevezetés

Ha indokolni szeretnénk egy új táblázatkezelő szoftver elkészítésének létjogosultságát, két kérdésre kell választ adnunk:

1. Milyen funkciókat kell ellátnia egy táblázatkezelő szoftvernek?
2. Mi az, ami hiányzik a jelenleg elterjedt szoftverekből? (Pl. Microsoft Excel)

Az első kérdésre talán az a legegyszerűbb válasz, hogy egy táblázatkezelő lehetőséget ad adatok tárolására és a bevitt adataink alapján újabb adatok kiszámítására. Ez a valóságban számtalan alkalmazási lehetőséget jelent. Az Excel-lel például lehet színes, táblázatos formájú órarendet készíteni, egy gyakorlati csoport eredményeit számontartani, családi költségvetést vezetni, stb.

Egy táblázatkezelőben minden cella tartalma egy funkcionális program. Egy cellába írhatunk egy egyszerű kifejezést (adat), vagy egy összetettebb programot, ami korábbi adatok függvényében számít ki egy új adatot. A táblázatkezelő tehát nem más, mint egy könnyen használható interfész a háttérben meghúzódó funkcionális nyelvhez. Ennek a nyelvnek az intuitív használatát számos funkció segíti. Lehetővé válik az összetett program komponensekre bontása, és az egyes komponensek eredményeinek hatékony vizualizációja.

Ha tekintjük napjaink legnépszerűbb táblázatkezelő szoftverét, az Excel-t, azt láthatjuk, hogy a fent leírt feladatot kiválóan ellátja. Bővelkedik megjelenítéssel kapcsolatos opciókban, a felhasználói felület használata intuitív, az elérhető dokumentáció közérthető. Fő hiányosságát nem is ebben látom, hanem az általa használt programozási nyelvben. Az Excel-ben a szoftver saját programozási nyelvét használhatjuk, aminek bővítésére a VBA programnyelv használatával van lehetőség. (*refe-*

rencia?) Ez azonban nem a legkényelmesebb megoldás, nehézkes egy összetettebb számítási funkciót hozzáadni az eszköztárunkhoz.

Ezen probléma megoldására teszek kísérletet dolgozatomban. Egy olyan táblázatkezelő szoftvert készítettem el, aminek a celláiba – a táblázatkezelő funkciók megfelelő ellátása érdekében kissé kiegészített – Haskell nyelven lehet programokat írni. Így a felhasználó rendelkezésére áll egy általános célú programnyelv teljes eszköztára.

fejezet 2

Felhasználói dokumentáció

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} .

2.1 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 elit¹, 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.

2.1.1 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.

¹Phasellus faucibus varius purus, nec tristique enim porta vitae.

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.
- Sagittis vulputate lorem.

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

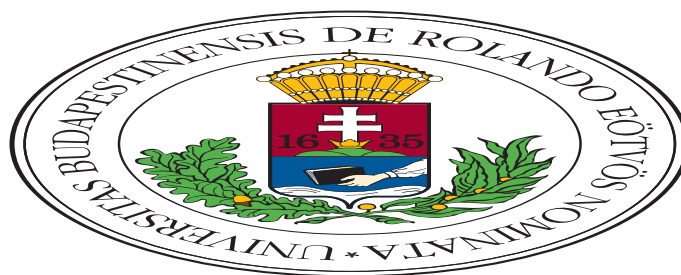
Cras maximus massa commodo pellentesque viverra.

Morbi sit amet ante risus. Aliquam nec sollicitudin mauris

Ut aliquam rhoncus sapien luctus viverra arcu iaculis posuere

2.2 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:

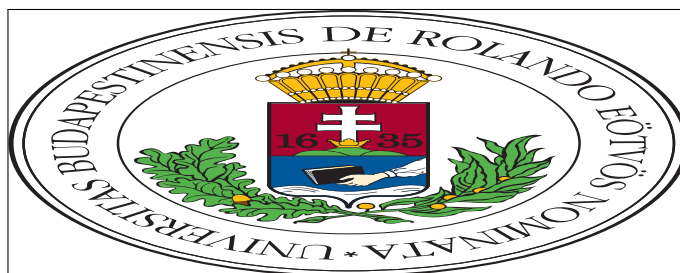


ábra 2.1: Quisque ac tincidunt leo

2.2.1 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.



ábra 2.2: Quisque ac tincidunt leo

2.2.2 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

ábra 2.3: 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.

2.3 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.

?tablename? 2.1: Maecenas tincidunt non justo quis accumsan

2.3.1 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%

?tablename? 2.2: Vivamus ac arcu fringilla, fermentum neque sed, interdum erat.

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

2.3.2 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 egestas 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.

?tablename? 2.3: Praesent ullamcorper consequat tellus ut eleifend

fejezet 3

Fejlesztői dokumentáció

3.1 A fejlesztői dokumentáció felépítése

A fejlesztői dokumentáció három nagy részből áll. Az 3.2 részben ismertetésre kerülnek a szoftver készítése során felhasznált technológiák, valamint nagy vonalakban a szoftver logikai felépítése. (Milyen programkomponensek vannak, milyen feladatokat látnak el, hogyan kapcsolódnak egymáshoz.) A 3.3 rész tartalmazza az egyes komponensek részletesebb leírását. Minden komponens esetén ismertetésre kerül a más komponensek felé nyilvánosságra hozott interfész, valamint a komponensek működési elve, beleértve a használt típusok leírását és a fontosabb algoritmusok működési elvét. A 3.4 rész tartalmazza a tesztelési eljárás leírását és a tesztelés eredményeit.

3.2 A szoftver felépítése

3.2.1 Felhasznált technológiák összefoglalása

Az alkalmazás Haskell nyelven íródott. A grafikus megjelenítés *GTK+* alapú, a *gtk2hs* csomag által biztosított bindingokat használtam a grafikus felület kezeléséhez. Ez a csomag a *GTK+* osztályhierarchiáját Haskell típusosztályok hierarchiájaként adja vissza. Az egyes osztályok metódusainak a típusosztályok definíciójában szereplő függvények felelnek meg. A *GTK+* típusai foreign pointerok segítségével vannak megvalósítva, és *IO*-ban használhatók.

Az alkalmazás a *GTK+* logikájának megfelelően eseményvezérelt. A felhasználó akciói eseményeket váltanak ki, amelyek hatására handlerek futnak le. A handlerek minden esetben *IO* akciók, amelyek valamilyen módon módosítják a globális állapotot (lásd 3.2.1. Globális állapot). A szoftver fejlesztése során fontos volt, hogy minél kevesebb legyen az tisztátalan (impure), *IO*-n belül elvégzett számítás. Igyekeztem a program logikájának minél nagyobb részét egy tiszta, nem monadikus környezetben megvalósítani. Így a számítások helyessége könnyebben tesztelhető/verifikálható, a handlerek már keveset számolnak az *IO*-ban.

Az alkalmazás a parseolási feladatokhoz a *parsec* csomagot, a gráfok kezeléséhez az *fgl* csomagot, a ghci futtatásához pedig a *ghcid* csomagot használja. A modell adatainak könnyebb kezeléséhez a *microlens-platform* csomagot használtam, ami a jól ismert *lens* csomag egy kevesebb funkciót és kevesebb függőséget tartalmazó változata. A függőségek pontos listája elérhető a Felhasználói dokumentációban, illetve az egyes programkomponensek részletes leírásakor is említésre kerülnek a fontosabb felhasznált csomagok.

3.2.2 A globális állapot

Az alkalmazás fő felépítését egy az FP Complete blogján megjelent cikk (**IDE KÉNE EGY REFERENCIA**) inspirálta. Az alkalmazás a globális (olvasható) állapotot a *ReaderT* monád transzformer segítségével valósítja meg, az alkalmazás vezérlése így egy *ReaderT Env IO* környezetben történik, ahol *Env* a globális állapotot leíró adattípus. Fontos megjegyezni, hogy bár az *Env* típus komponensei az inicializálás után sosem módosulhatnak, a mögöttes állapot még változhat, hiszen a komponensek módosítható referenciák. Ez nagyon hasonló a Java nyelvben használható konstans referencia koncepciójához: a referencia nem változhat, de a referált adat igen.

A fentebb referált cikk által inspirálva a (GUI komponensein kívüli) globális állapot egy *StateT* transzformer helyett módosítható referenciákkal (*IORef* és *MVar*) kezeltetik. Ugyanis hiába tiszta, ha globálisan használjuk a *StateT*-t, valójában – a programlogika szintjén – ugyanúgy egy globális, módosítható állapotot vezetünk be. Szintén egy szempont, hogy a *GTK+* alapú *GUI* miatt eleve szerepelnek módosítható referenciák (foreign pointer) a globális állapotban, így ez a probléma sem-

miképpen nem kerülhető el teljesen. Egy további érv a globális *StateT* ellen, hogy egy nagyobb monad stack szükségszerűen bonyolítja a programot. A *ReaderT IO* ellenben még kifejezetten könnyen kezelhető. A cikk konkurenciához köthető problémákat is említ a *StateT*-vel kapcsolatban. Ez a szoftver jelenlegi verziójában még nem olyan jelentős (lévén a mostani implementáció nagyon kis mértékben épít konkurenciára). Azonban a jövőre nézve mindenképpen előnyös, ha a szoftvert könnyen lehet a konkurrens paradigma szerint bővíteni.

Ezen bevezető után tekintsük a globális állapot definícióját! Az alábbi típusdefiníció az *App.Types* modulban található:

```
1 data Env = Env { evalControl  :: EvalControl
2                  , gui         :: Gui
3                  , state       :: IORef Spreadsheet
4                  , file        :: IORef (Maybe File)
5                  } deriving Eq
```

Code 3.1: Az Env típus

Az *evalControl* mező tartalmazza a kifejezések ghci-ban való kiértékelés hez szükséges erőforrásokat. A *gui* mező tartalmazza a GUI komponenseit. A *state* mező egy módosítható referencia, ami a számológépet reprezentáló, *Spreadsheet* típusú adatot referálja. A *file* adattag tartalmazza az éppen a táblázatkezelőbe betöltött fájl fontosabb adatait, amennyiben be van töltve egy fájl. A mezők részletes leírása (típusdefinícióval együtt) szerepelni fog a komponenseket használó modulok részletes leírásában.

3.2.3 Programkomponensek és modulszerkezet (nem teljes!!!!!!!!!!)

Az alábbiakban röviden összefoglalom a szoftver moduljainak fő feladatát:

- **Main** – főprogram
- **App** – az alkalmazás fő logikája, eseménykezelés
 - **App.CreateEnv** – a globális állapot inicializálása, GUI funkcionalitás nélkül

- **App.RunApp** – a főprogram definiálása, a main loop terminálásakor végrehajtandó IO akciók megadása
- **App.Setup** – funkcionalitás hozzárendelése a GUI-hoz
- **App.Types** – a globális állapothoz tartozó típusdefiníciók
- **Eval** – kifejezések GHCi-ban történő kiértékelése
 - **Eval.EvalMain** – a tényleges kiértékelést végző szál főprogramja
 - **Eval.Ghci** – az App számára biztosított interfész a kiértékeléshez
- **Persistence** – az App számára biztosított interfész fájlok mentéséhez és betöltéséhez
- **Spreadsheet** – a számológéptábla reprezentációja és műveletei
 - **Spreadsheet.CodeGeneration** – kódgenerálás a kiértékeléshez, **FULL NEM ITT KÉNE LENNIE 44!4!!4**
 - **Spreadsheet.Interface** – a számológéptábla műveletei, amiket az App használhat
 - **Spreadsheet.Parser** – felhasználó által írt kód (string) reprezentációjának előállítás
 - **Spreadsheet.Types** – a számológéptábla és kapcsolódó kivételek típusdefiníciói

3.3 A program komponenseinek részletes leírása

3.3.1 A főprogram

A főprogram a

fejezet 4

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 portitor, 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.

?appendixname? A

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.

Bibliography

- [1] O. J. Dahl, E. W. Dijkstra és C. A. R. Hoare, szerk. *Structured Programming*. London, UK, UK: Academic Press Ltd., 1972. ISBN: 0-12-200550-3.
- [2] Thomas H. Cormen és tsai. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN: 0262033844, 9780262033848.
- [3] Glenn E. Krasner és Stephen T. Pope. “A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80”. *J. Object Oriented Program.* 1.3 (1988. aug.), old. 26–49. ISSN: 0896-8438. URL: <http://dl.acm.org/citation.cfm?id=50757.50759>.
- [4] E. Dijkstra. “Classics in Software Engineering”. Szerk. Edward Nash Yourdon. Upper Saddle River, NJ, USA: Yourdon Press, 1979. Fej. Go to Statement Considered Harmful, old. 27–33. ISBN: 0-917072-14-6. URL: <http://dl.acm.org/citation.cfm?id=1241515.1241518>.

?listfigurename?

2.1	Quisque ac tincidunt leo	7
2.2	Quisque ac tincidunt leo	8
2.3	Aenean porttitor mi volutpat massa gravida	8

?listtablename?

2.1	Maecenas tincidunt non justo quis accumsan	9
2.2	Rövid cím a táblázatjegyzékbe	9
2.3	Praesent ullamcorper consequat tellus ut eleifend	11

List of Codes

3.1	Az Env típus	14
-----	------------------------	----