



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI  
TANSZÉK

# Ekvivalens Python forráskód-párok generálása

*Témavezető:*

Szalontai Balázs  
doktorandusz

*Szerző:*

Verebics Peter  
programtervező informatikus BSc

*Budapest, 2024*

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>3</b>
<b>2. Felhasználói dokumentáció</b>	<b>4</b>
2.1. Futtatási környezet . . . . .	4
2.2. Adatbázis beállítása . . . . .	4
2.3. Adathalmazt generáló CLI . . . . .	5
2.3.1. Alkalmazás használata . . . . .	5
2.4. Átalakításokat szemléltető GUI . . . . .	5
2.5. Képek, ábrák . . . . .	5
2.5.1. Képek szegélyezése . . . . .	6
2.5.2. Képek csoportosítása . . . . .	6
2.6. Táblázatok . . . . .	7
2.6.1. Sorok és oszlopok egyesítése . . . . .	7
2.6.2. Több oldalra átnyúló táblázatok . . . . .	8
<b>3. Fejlesztői dokumentáció</b>	<b>10</b>
3.1. Csomagok . . . . .	10
3.1.1. <i>app</i> csomag . . . . .	10
3.2. Forráskódok . . . . .	11
3.2.1. Algoritmusok . . . . .	12
<b>4. Összegzés</b>	<b>13</b>
<b>Köszönetnyilvánítás</b>	<b>14</b>
<b>A. Szimulációs eredmények</b>	<b>15</b>
<b>Irodalomjegyzék</b>	<b>17</b>
<b>Ábrajegyzék</b>	<b>17</b>

Táblázatjegyzék	18
Algoritmusjegyzék	19
Forráskódjegyzék	20

# 1. fejezet

## Bevezetés

Egyre elterjedtebbek a forráskódokat mély tanuláson alapuló módszerekkel refaktoráló szoftverek. Ezek a szoftverek sokszor helytelenül refaktorálják a bemeneti forráskódot, azaz változtatnak a kód jelentésén. A forráskódok ekvivalenciájának vizsgálata fontos feladat, ugyanis ha lenne lehetőség egy forráskód pár ekvivalenciájának megállapítására, akkor a refaktoráló szoftverek által adott hibás megoldásokat ki lehetne szűrni.

Az ekvivalencia meghatározásának feladata megoldható mély tanuláson alapuló módszer segítségével, ami a párba állított forráskódokról eldönti, hogy azok ekvivalensek-e. Egy ilyen háló tanításához szükség van egy adathalmazra, ami forráskód párokat tartalmaz felcímkézve azzal, hogy ekvivalensek-e. A szakdolgozatomban Python kódok ekvivalenciáját eldöntő, mélytanuló háló számára generálok adathalmazt. A bemutatott szoftver segítségével lehetőség nyílik az adathalmaz előállítására és az átalakítások szemléltetésére.

A szakdolgozatomban az adathalmaz generálására használt, Python kódokat átalakító szoftvert mutatom be. A szoftver célja minél több ekvivalens átalakítás megvalósítása, az átalakítások szemléltetéséhez grafikus felhasználói felület is készül. Az átalakítások lehetnek egyszerűbbek (pl. azonosítók átnevezése), vagy összetettebbek (pl. a listát előállító for-ciklus "comprehension" kifejezéssé alakítása). Ezek az átalakítások a forráskód absztrakt szintaxisfájának módosításával valósíthatók meg.

## 2. fejezet

# Felhasználói dokumentáció

A szoftvernek két felhasználói felülete van: egy parancssoros felhasználói felület (CLI - command line interface) az adathalmaz generálásához és egy grafikus felhasználói felület (GUI - graphical user interface) az átalakítások szemléltetéséhez és az adathalmaz böngészéséhez. Mindkét felhasználói felület nyelve angol.

### 2.1. Futtatási környezet

Az alkalmazás egy Python 3.10-es vagy újabb verziójú Python interpreterrel futtatható. A futtatáshoz az alkalmazás függőségeit installálni kell a pip csomagkezelővel. A függőségek a projekt mappában a *requirements.txt* fájlban találhatók, és a következő paranccsal installálhatók:

```
1 $ pip install -r requirements.txt
```

### 2.2. Adatbázis beállítása

Az alkalmazás működéséhez szükség van egy elérhető *mongodb* adatbázisra. Ezt a projekt *config* könyvtárában található *default.ini* konfigurációs fájlban lehet beállítani.

Három paraméter szükséges:

- *host* - host neve (URL), pl.: localhost
- *port* - port száma (egész), pl.: 27017
- *database* - adatbázis neve

## 2.3. Adathalmazt generáló CLI

Ezzel a CLI alkalmazással van lehetőségünk egy adathalmaz generálására.

### 2.3.1. Alkalmazás használata

Az alkalmazás a projekt könyvtárból indítható a következő paranccsal:

```
1 $ python -m source.persistor <mode> <path>
```

Két paraméter szükséges:

1. *mode* - az adatok forrásának típusa, lehetséges értékek:
  - *csv* - csv fájlból olvassa a forrásfájlok tartalmát
  - *dir* - könyvtárból rekurzívan olvassa a forrásfájlokat
2. *path* - az adatok forrásának elérési útvonala, csv olvasása esetén a forráskódok csak az első oszlopban lehetnek

```
reading csv, be patient this might take a while...
-----[0]
No changes to save.
-----[1]
Fixed 13 errors.
1 file reformatted
1 file reformatted
1 file reformatted
Inserted 3 changes on hash f3842737e0fe9141df61f299c99e65d9b20a41ae3c76644ef663483aec8aeb31.
-----[2]
Fixed 12 errors.
1 file reformatted
1 file reformatted
1 file reformatted
Inserted 3 changes on hash bdd72cdcf458519e9e88afa499c0e6363e427fa7ea8c9fe5484a426babad40c1.
-----[3]
```

2.1. ábra. A program a kiírja a feldolgozott fájlal kapcsolatos adatokat.

## 2.4. Átalakításokat szemléltető GUI

Ez a GUI alkalmazás szemlélteti az átalakításokat. Kipróbálhatunk vele egy vagy több átalakítást, vizualizálhatjuk kódok absztrakt szintaxis fájt, és az adatbázisba bekerült átalakítások eredményét is megnézhetjük.

## 2.5. 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

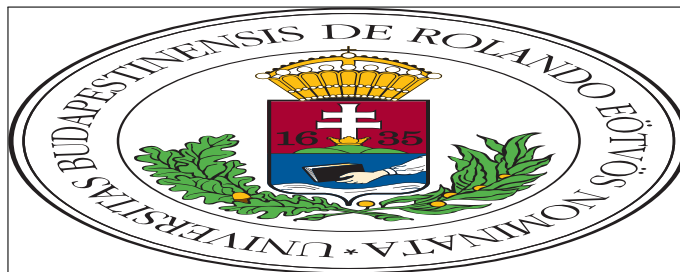
id diam ligula. Nulla sodales purus id mi consequat, eu venenatis odio pharetra. Cras a arcu quam. Suspendisse augue risus, pulvinar a turpis et, commodo aliquet turpis. Nulla aliquam scelerisque mi eget pharetra. Mauris sed posuere elit, ac lobortis metus. Proin lacinia sit amet diam sed auctor. Nam viverra orci id sapien sollicitudin, a aliquam lacus suscipit, 2.2. ábra:



2.2. ábra. Quisque ac tincidunt leo

### 2.5.1. 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.



2.3. ábra. Quisque ac tincidunt leo

### 2.5.2. 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, 2.4. ábra:



(a) Vestibulum quis mattis urna



(b) Donec hendrerit quis dui sit amet venenatis

## 2.4. 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.

## 2.6. 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.

2.1. táblázat. Maecenas tincidunt non justo quis accumsan

## 2.6.1. 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%

2.2. táblázat. Vivamus ac arcu fringilla, fermentum neque sed, interdum erat. Mauris bibendum mauris vitae enim mollis, et eleifend turpis aliquet.

### 2.6.2. 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>Suspendisse potenti</i>	<i>Lorem ipsum dolor sit amet</i>
<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>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.

2.3. táblázat. Praesent ullamcorper consequat tellus ut eleifend

## 3. fejezet

# Fejlesztői dokumentáció

A szoftver forráskódja több csomagban és modulban található. A forráskód jelentős része öt fő csomagba van szervezve.

Csomag	Rövid leírás
<i>app</i>	GUI alkalmazás csomagja
<i>client</i>	adatbázis kliens
<i>model</i>	adatok modellezése és mentése
<i>tests</i>	egység és egyéb tesztek
<i>transformations</i>	átalakítások forráskódja és API az átalakításokhoz

3.1. táblázat. A szoftver fő csomagjai

### 3.1. Csomagok

Minden csomag a szoftver egy jól elkülöníthető részét vagy funkcióját valósítja meg. A fő csomagok (a *tests* csomag kivételével) nem tartalmazzak "futtatható" fájlokat, a belépési pontok külön modulokba vannak szervezve.

#### 3.1.1. *app* csomag

Az *app* csomag feladata az átalakításokat szemléltető grafikus felhasználói felületű alkalmazás megvalósítása. Az alkalmazás architektúrája MVC (Model-View-Controller) szerű.

Az alkalmazásban két féle model különböztethető meg: állapot és adatelérési modellek. Az alkalmazás állapotának modelje az *app.appstate* modulban található. Az adatelérési modellek nem az alkalmazás csomagjában vannak definiálva mivel a szoftver más rétegeinek is szüksége van ezekre.

A felhasználói felület (View-réteg) kódja a *app.views* csomagban található, a Python-ban alaphól megtalálható *tkinter* könyvtárt használja, amit az erre építő *ttkbootstrap* könyvtárral egészít ki.

A kontrollerek feladata a kommunikáció a modellek és nézetek között. Csak a felhasználói felület két fő nézete a *RefactorTab* és a *DatabaseTab* rendelkeznek saját kontrollerekkel.

A model és nézet szétválasztása érdekében az állapot modelje az *observer* tervezési mintát használja a nézetek frissítésére.

## 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 3.1. és 3.2. forráskód:

```
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

Az 1. algoritmus egy általános elágazás és korlátozás algoritmust (*Branch and Bound algorithm*) mutat be. A 3. lépésben egy megfelelő kiválasztási szabályt kell alkalmazni. Példa forrása: Acta Cybernetica (ez egy hiperlink).

---

#### 1. algoritmus 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 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.

# Köszönetnyilvánítás

Amennyiben a szakdolgozati / diplomamunka projekted pénzügyi támogatást kapott egy projektből vagy az egyetemtől, jellemzően kötelező feltüntetni a dolgozatban is. A dolgozat elkészítéséhez segítséget nyújtó oktatók, hallgatótársak, kollégák felé is nyilvánítható külön köszönet.

## 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.

# Ábrák jegyzéke

2.1. A program a kiírja a feldolgozott fájlal kapcsolatos adatokat. . . . .	5
2.2. Quisque ac tincidunt leo . . . . .	6
2.3. Quisque ac tincidunt leo . . . . .	6
2.4. Aenean porttitor mi volutpat massa gravida . . . . .	7

# Táblázatok jegyzéke

2.1. Maecenas tincidunt non justo quis accumsan . . . . .	7
2.2. Rövid cím a táblázatjegyzékbe . . . . .	8
2.3. Praesent ullamcorper consequat tellus ut eleifend . . . . .	9
3.1. A szoftver fő csomagjai . . . . .	10

# Algoritmusjegyzék

1.	A general interval B&B algorithm . . . . .	12
----	--	----

# Forráskódjegyzék

3.1. Hello World in C++ . . . . .	11
3.2. Hello World in C# . . . . .	11