



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI  
TANSZÉK

## Dolgozat címe

*Témavezető:*

Témavezető Tamás  
egyetemi tanársegéd

*Szerző:*

Hallgató Hanga  
programtervező informatikus BSc

*Budapest, 2022*

Az eredeti szakdolgozati / diplomamunka témabejelentő helye.

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>3</b>
1.1. Statikus kódelemzés illetve refaktorálás . . . . .	3
1.2. RefactorErl . . . . .	4
1.3. Fejlesztő környezeti nyelvi támogatás . . . . .	4
1.4. Erlang LS . . . . .	5
1.5. Visual Studio Code . . . . .	5
1.6. Feladat . . . . .	5
<b>2. Felhasználói dokumentáció</b>	<b>6</b>
2.0.1. Rendszerkövetelmények . . . . .	6
2.0.2. Telepítés . . . . .	6
2.0.3. . . . .	6
<b>3. Fejlesztői dokumentáció</b>	<b>7</b>
3.1. Tételek, definíciók, megjegyzések . . . . .	7
3.1.1. Egyenletek, matematika . . . . .	8
3.2. Forráskódok . . . . .	9
3.2.1. Algoritmusok . . . . .	10
<b>4. Összegzés</b>	<b>11</b>
<b>A. Szimulációs eredmények</b>	<b>12</b>
<b>Irodalomjegyzék</b>	<b>14</b>
<b>Ábrajegyzék</b>	<b>15</b>
<b>Táblázatjegyzék</b>	<b>16</b>
<b>Algoritmusjegyzék</b>	<b>17</b>



# 1. fejezet

## Bevezetés

### 1.1. Statikus kódelemzés, illetve refaktorálás

Mi az a kód elemzés? Nem a programozó feladata a kódot megérteni? Ezen kérdések mind felmerülhetnek bennünk. A kódelemző egy olyan szoftver, ami képes a forráskód elemzésére, annak futtatása nélkül. Ugyan egy kicsi projektben egy elemző program használata még nem feltétlen szükséges, azonban egy ipari méretű szoftver esetében a programozók feladatát nagyban megkönnyítik.

Egy statikus elemző szoftver segítségével rengeteg információt tudhatunk meg még a fejlesztés során, mint például: *kód-metrikákat* és *függőségi információkat*, melyek olyan kódrészek tekintében felettébb fontosak, amiket nem mi írtunk.

Szintén nagy méretű projekteknél fontos a refaktoráló, azaz kód átalakító eszközök használata is, hiszen manuálisan egy függvény vagy egy változó átnevezése igen nagy kihívást jelent. Azonban nem csak ilyen méretű szoftvereknél hasznos, amit esetlegesen több ember is használt, hanem egy projekt fejlesztésében, fejlődésében is jelentősége van egy ilyen, refaktoráló eszköznek. Gondoljunk bele: elkezdünk egy projektet, (ami lehet egy független szoftver, vagy akár egy új funkció egy már meglévő projektben) először egy prototípust készítünk belőle, majd az alapján folytatjuk a fejlesztést, jellemzően már egy tervet követve, azonban a legalaposabb tervezés ellenére is előfordulhat, hogy egy kódot át kell alakítani.

## 1.2. RefactorErl

A RefactorErl az ELTE Informatikai Karán futó kutatás-fejlesztési projekt melynek fő célja az, hogy Erlang forrásokhoz statikus elemző szoftvert készítsen. Az eszközzel az elemzés mellett kódátalakítást, refaktorálást is elvégezhetünk. Főbb funkcionálisáiként kiemelném a: függőségi gráf vizualizációját, illetve a szematikus lekérdező nyelvét, amelyen keresztül többek között olyan információk érhetőek el, mint nem használt makró definíciók, változó lehetséges értékei vagy biztonsági sérülékenységek[1]. Az eszköznek jelenleg több felhasználói felülete van, amelyből kiemelném a webes felületet és a Erlang Shell-en keresztül elérhető parancssori felületét.

## 1.3. Fejlesztő környezeti nyelvi támogatás

Napjainkban nagyon elterjedtek az integrált fejlesztői környezetek, avagy IDE<sup>1</sup>-k. Az ilyen fejlesztői környezetben egy szoftveren belül tudjuk szerkeszteni a forráskódot, konzolt kezelni, verziót kezelni, s a kódukról diagnosztikákat is kapunk, szinte valós időben, amennyiben az adott programozási nyelvhez rendelkezésre áll ilyen IDE-kompatibilis rendszer. (Például: *Visual Studio*, *IntelliJ*, *Emacs*)

A nyelvi támogatás nem jelent mást, mint egy a fejlesztői környezetbe integrált szoftvert, ami segíti a programozó munkáját a hibák és figyelmeztetések vizualizációjában, kód-kiegészítési javaslatokat is tesz, illetve a dokumentáció egy részét is elérhetővé teszi a szerkesztőn belül (Például: az adott függvény paraméterezése, leírása)

Sok esetben az ilyen nyelvi támogató rendszerek kifejezetten egy rendszerhez készültek, egy másik szerkesztőben pedig nem használhatóak, pedig a forrást ugyanúgy kell elemezni és a fejlesztőnek is közel azonos diagnosztikákat, visszajelzéseket kell nyújtani. Erre ad megoldást a Microsoft Language Server Protocolja (LSP), ami egy fejlesztői környezet független nyelvi kiszolgáló protokollt jellemez. Ennek az előnye, hogy a kiszolgálót csak egyszer kell megírni, és onnantól kezdve könnyedén integrálható, bármely kliensbe (amennyiben az megvalósítja a protokollt)

---

<sup>1</sup>*Integrated Development Environment*

## 1.4. Erlang LS

Az Erlang Language Server (röviden: Erlang LS) a nyelvi kiszolgáló az Erlang programozási nyelvhez. *Roberto Aloi* vezetésével indult el a nyílt forráskódú projekt, aminek mára már jelentős karbantartói és fejlesztői csapata van, a világ minden pontjáról, Magyarországról is. Az Erlang LS alapvetően egy szerkesztő független megvalósítás, azonban a csapat a Visual Studio Code-hoz fejlesztett egy interfész kiegészítőt is, ami segítségével az LS, szépen betud épülni a szerkesztőbe. Az elemző eszköz elérhető továbbá például Emacs, IntelliJ és még sok más IDE-ben is. Az Erlang LS támogatja a DAP<sup>2</sup>-ot és a legfrissebb kiadása már a RefactorErl diagnosztikák integrációját is támogatja.

## 1.5. Visual Studio Code

A Visual Studio Code manapság az egyik legelterjedtebb kódszerkesztő szoftver, ami köszönhető nyílt forrásának, illetve annak elérhető macOS, Linux és Windows operációs rendszereken, sőt akár (béta verzióban) az interneten is. Az eszköz beépített grafikus verzió kezelést és integrált terminált is biztosít, a plugin<sup>3</sup> architektúrája végett, pedig szinte a kedvünkre bővíthető és személyre szabható<sup>4</sup>.

## 1.6. Feladat

---

<sup>2</sup>Debugger Adapter Protocol

<sup>3</sup>kiegészítő, beépülő modul

<sup>4</sup>Ezen kiegészítőket az alkalmazáson belül elérhető Marketplaceből tudjuk letölteni

## 2. fejezet

# Felhasználói dokumentáció

### 2.0.1. Rendszerkövetelmények

### 2.0.2. Telepítés

Erlang/OTP telepítése

Visual Studio Code telepítése

Erlang LS bővített változatának telepítése

RefactorErl telepítése

...

RefactorErl Visualiser telepítése

### 2.0.3.



## 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 3.1. egyenlet:

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

## 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 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.

## 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] .



## Ábrák jegyzéke

## Táblázatok jegyzéke

# Algoritmusjegyzék

1.	A general interval B&B algorithm . . . . .	10
----	--------------------------------------------	----

# Forráskódjegyzék

3.1. Hello World in C++ . . . . .	9
3.2. Hello World in C# . . . . .	9