

Kodolányi János Egyetem

Üzemmérnök-informatikus szak

Tárgy: Rendszermodellezés, Adatvizualizáció, Emberi viselkedés és kommunikáció tárgycsoport

Tanév: 2023/2024 II. félév

Név: Kozma Viktor

NEPTUN Kód: NU6XEK

Konzulens: Dr. Pitlik László

**Szoftverfejlesztési döntéstámogatás mesterséges intelligenciával**

Tartalomjegyzék

[1. Kivonat 3](#_Toc164159249)

[2. Abstract 3](#_Toc164159250)

[3. Bevezetés 4](#_Toc164159251)

[3.1. Célok 4](#_Toc164159252)

[3.2. Feladatok 4](#_Toc164159253)

[3.3. Célcsoportok 5](#_Toc164159254)

[3.4. Hasznosság 5](#_Toc164159255)

3.5. Motiváció

[4. Szakirodalmi/saját előzmények 5](#_Toc164159256)

[5. Adatok és módszerek 5](#_Toc164159257)

[5.1. OAM. 5.1.1.Objektumok 11](#_Toc164159258)

[5.1.2. Attribútumok 11](#_Toc164159259)

[. 5.2. Eelemzések 5.2.1. … 5.2.1.1.Elemzés 1. 11](#_Toc164159260)

[5.2.1.2. Eredmény 1. 12](#_Toc164159261)

5.2.2…. 5.2.2.1[. Elemzés 2. 12](#_Toc164159262)

5.2.2.2.[. Eredmény 2. 14](#_Toc164159263)

5.2.3…..5.2.3.1.[. Elemzés 3. 15](#_Toc164159264)

[5.2.3.2.. Eredmény 3. 16](#_Toc164159265)

[6. ChatGPT vélemény 17](#_Toc164159266)

[7. Vita 19](#_Toc164159267)

[8. Következtetések 19](#_Toc164159268)

[9. Jövőkép 19](#_Toc164159269)

[10. Mellékletek 19](#_Toc164159270)

[18.1. Rövidítések jegyzéke 19](#_Toc164159271)

[18.2. Referenciák 20](#_Toc164159272)

[18.3. Ábrák 21](#_Toc164159273)

# Kivonat

Jelen dolgozat célja a "legjobb" programnyelv meghatározása a [REST API](#_Rövidítések_jegyzéke) szolgáltatások fejlesztésének kontextusában. A kiindulópont az volt, hogy leendő szakdolgozatomban egy ilyen (automatizált szoftver-evaluációs) szolgáltatás elkészítését tűztem ki célul, így ezen dolgozat témája kvázi kutatómunka/teszteset a feladat elvégzéséhez. Felmerült a kérdés, hogy azonosíthatók-e és megfogalmazhatók-e objektív kritériumok/értékelési módszerek, amelyek alapján eldönthető, mely programnyelv a legalkalmasabb adott célra. A dolgozat célja tehát a programozási nyelvek összehasonlítása azok tulajdonságai és népszerűségük alapján a [REST API](#_Rövidítések_jegyzéke) szolgáltatások fejlesztésének kontextusában.

A cél elérése érdekében a Gemini [ChatGPT](#_Rövidítések_jegyzéke) (vö. Google mesterséges intelligencia) segítségét vettem igénybe egy adathalmaz létrehozásában a programozási nyelvek releváns tulajdonságairól (Xi). Ezeket az adatokat összevetettem a Google Trends-ben szereplő népszerűségi mutatókkal (Y). A cél egy "termelési függvény" felállítása volt, amellyel becsülhető a népszerűség a programozási nyelvek leíró adatai alapján.

A valós Google Trend-alapú érdeklődési adatokra vonatkozó becslési hibák elemzése alapján következtetéseket lehet levonni a programozási nyelvek iránti érdeklődés túlértékeltségéről/alulértékeltségéről/normaszerűségéről, illetve a népszerűségük valósághűségéről (validitásáról).

Kulcsszavak: [REST API](#_Rövidítések_jegyzéke), Gemini, [ChatGPT](#_Rövidítések_jegyzéke), programozási nyelvek, népszerűség, érdeklődés, Google Trends, objektív kritériumok, termelési függvény, alul/túlértékeltség, normaszerűség, [OAM](#_Rövidítések_jegyzéke), [COCO](#_Rövidítések_jegyzéke)

# Abstract

In my thesis, I define the "best" programming language in the context of developing [REST API](#_Rövidítések_jegyzéke) services. The starting point was to set the creation of such a service as the goal for my future thesis, making the topic of this thesis essentially a research task for accomplishing the assignment. The question arose whether objective criteria could be identified and formulated to determine which programming language is most suitable. The aim of the thesis is to compare programming languages based on their characteristics and popularity in the context of developing [REST API](#_Rövidítések_jegyzéke) services.

To achieve this goal, I utilized the assistance of Gemini [ChatGPT](#_Rövidítések_jegyzéke) in creating a dataset about the relevant properties of programming languages. I compared this data with popularity metrics from Google Trends. The goal was to establish a "production function" to estimate popularity based on the descriptive data of programming languages.

Analysis of estimation errors allows conclusions to be drawn about the overvaluation, undervaluation, and realistic popularity of programming languages.

Keywords: [REST API](#_Rövidítések_jegyzéke), Gemini, [ChatGPT](#_Rövidítések_jegyzéke), programming languages, popularity, Google Trends, objective criteria, production function, overvaluation, [OAM](#_Rövidítések_jegyzéke), [COCO](#_Rövidítések_jegyzéke)

# Bevezetés

Az informatika fejlődésével szinte évente hallani új programnyelvek születéséről. A bőség zavara, hogy vajon melyik programnyelvet alkalmazzuk egy adott feladat elvégzéséhez. Ezen dolgozat meghatározza és összehasonlítja a [REST API](#_Rövidítések_jegyzéke) fejlesztésre alkalmas (vö. … fejezetcímre utalás, ahol kiderül, mik is ezek az elvárások) programnyelveket azok releváns tulajdonságai alapján, illetve összeveti azok népszerűségi adataival (vö. Google Trends).

## Célok

A feladatok elvégzésével meghatározni későbbi szakdolgozatomban alkalmazandó programozási nyelv kiválasztását (vö. döntéstámogatás). Nem világos, le lesz-e programozva maga a jelen esetben éppen nyelvkiválasztási folyamat a jövőben a szerző által, vagy sem (azaz szabad préda a dolgozat és az xls mások programozási feladatéhsége esetére)?

Továbbá, hogy bemutassam ennek a feladatmegoldásnak a folyamatát esetleges további felhasználásra.

## Feladatok

Első feladatként az adatvagyont kellett létrehozni, tehát összegyűjteni azon releváns információkat, amelyeket felhasználhatok az összehasonlításban. Szükség volt a programnyelvek leíró tulajdonságaira, illetve ezek népszerűségi adataira. Az összegyűjtött adatokból létrehoztam egy objektum-attribútum mátrixot ([OAM](#_Rövidítések_jegyzéke)). Következő lépésként az [OAM](#_Rövidítések_jegyzéke) adatait rangsoroltam (fejezetre utalás, ahol az irányok kifejtése megtörténik), majd elemeztem a [COCO STD](#_Rövidítések_jegyzéke) online robot, árelőny-termelési függvény generátor segítségével, illetve az anti-diszkriminatív módszert alkalmazó [COCO Y0](#_Rövidítések_jegyzéke) online robottal. Végül elkészítettem ezt a dolgozatot, amely részletesen bemutatja az elemzés reprodukálható folyamatát.

## Célcsoportok

Jelen dolgozatomat ajánlom minden olyan leendő kollégának, aki saját projektje kapcsán keres megfelelő döntéstámogató módszert a megfelelő eszközök kiválasztásában. Nem kizárólag programnyelvek kiválasztásában, de az itt bemutatott (pl. anti-diszkriminatív) módszerekkel az élet bármely területén objektív döntéseket hozhatunk.

## Hasznosság

Dolgozatom elsődleges hasznossága a lelendő szakdolgozatom elkészítéséhez szükséges kutatómunka, melynek eredményeit felhasználva tudom elérni a kívánt célt (vö. diploma megszerzése).

A hasznosság közgazdasági értelmezése az információs többletérték fogalmához kapcsolódik: vagyis kik miért miből mennyit hajlandók fizetni ezen fejlesztésért és ezen összeg fedezi-e a fejlesztés költségeit?

# Szakirodalmi/saját előzmények

Dolgozatom elkészítésében korábbi szintén a szakdolgozattal kapcsolatos témában készült [OAM](#_Rövidítések_jegyzéke) elemzés segített, amelyben digitális aláírópadokat hasonlítottam össze technikai tulajdonságaik alapján (vö. [sinature\_pads\_v2.xlsx](#_Referenciák)). Az aláírópadok közül kerestem a műszakilag legjobb terméket, illetve ellenőriztem, hogy az eszközök árazása vajon korrekt módon történt. A konkrét REST API programnyelvek témában sok összehasonlító cikk született már, de ahogy évente jelennek meg újabb nyelvek és keretrendszerek érdemes mindig újra elvégezni az összehasonlítást. Dolgozatom egyik kiinduló pontja volt a [datarundown.com](https://datarundown.com/) cikke a „[The 15 Best Languages for REST API: A Complete Exploration](https://datarundown.com/rest-api/)” címmel.

# Adatok és módszerek

A szükséges adatvagyon létrehozásához meg kellett határozni kifejezetten a célfeladat alapján vizsgálandó programnyelveket (vö. objektumok) és tulajdonságokat, jellemzőket (vö. attribútumok). Tehát a [REST API](#_Rövidítések_jegyzéke) fejlesztésnél releváns tulajdonságokra szükséges koncentrálni, ezért ebben segítségül hívtam a Google Gemini [ChatGPT](#_Rövidítések_jegyzéke) szolgáltatását. Azért ezt a [ChatGPT](#_Rövidítések_jegyzéke) szolgáltatást választottam, mert egyrészt ingyenes, másrészt mögötte van a Google hatalmas keresőmotorja is, így az internetes tartalomban is tud keresni.

Számos tényezőt kell figyelembe venni a legmegfelelőbb programnyelv kiválasztásakor egy [REST API](#_Rövidítések_jegyzéke) szolgáltatás fejlesztéséhez. A [ChatGPT](#_Rövidítések_jegyzéke) által felsorolt tényezőkön (vö. <rest_api_programnyelvek.xlsx> „ChatGPT” munkalap) kívül leendő szakdolgozatom témájával összefüggésben bővíttettem a [.NET DLL](#_Rövidítések_jegyzéke) integrálási lehetőséggel. A kapott táblázatban (1. ábra) (nyers adatok) szöveges értékelések szerepeltek, melyeket valahogy osztályozni/sorszámozni szükséges. Erre minden egyes tulajdonságnál egyedi osztályozási/rangsorolási rendszert kellett alkalmazni.

A következőkben felsorolom a figyelembe vett tulajdonságokat (attribútumokat).

* **Teljesítmény**: A nyelv általános teljesítménye, beleértve a sebességet és a feldolgozási kapacitást. Minél nagyobb az érték annál jobb.
* **Memóriahasználat**: A nyelv memóriahasználata, befolyásolva a futtatási környezet és a projekt mérete által. Minél alacsonyabb, annál jobb.
* **Fejlesztési sebesség**: A nyelvvel való fejlesztés gyorsasága, beleértve a kódolási és tesztelési folyamatokat. Minél nagyobb, annál jobb.
* **Tanulási görbe**: A nyelv elsajátításának nehézségi foka, kezdő és haladó programozók számára. Minél kisebb, annál jobb.
* **Ismert nyelvek**: A nyelv elterjedtsége és népszerűsége a programozók körében. Ezt a tulajdonságot kivettem az összehasonlításból, mivel helyette a Google Trends népszerűségi adatokat használtam fel.
* **Szaktudás**: Bizonyos nyelvekhez specifikus szaktudás szükséges. Minél kisebb, annál jobb.
* **Közösségi támogatás**: A nyelvhez kapcsolódó online fórumok, dokumentációk és aktív közösség mértéke. Minél nagyobb, annál jobb.
* **Operációs rendszer**: Az operációs rendszerek, amelyeken a nyelv futtatható. Minél több, annál jobb.
* **REST API keretrendszerek száma**: A nyelvhez elérhető [REST API](#_Rövidítések_jegyzéke) keretrendszerek száma és sokfélesége. Minél több, annál jobb.
* **Kódolási stílus**: A nyelvhez jellemző kódolási stílus és paradigma (pl. objektumorientált, funkcionális). Minél több, annál jobb.
* **Tesztelhetőség**: A nyelvben elérhető tesztelési keretrendszerek és a kód tesztelésének egyszerűsége. Minél nagyobb, annál jobb.
* **Dokumentáció**: A nyelvhez kapcsolódó dokumentáció minősége és elérhetősége. Minél több, annál jobb.
* **.NET DLL integráció**: A nyelv képessége [.NET DLL](#_Rövidítések_jegyzéke) könyvtárak integrálására és használatára. Minél nagyobb, annál jobb.

A képen szöveg, képernyőkép, sor, Betűtípus látható

Automatikusan generált leírás

*1. ábra: Gemini nyers adat  
Forrás:* [*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„Nyers adat Gemini” munkalap*

Az osztályozást/sorszámozást követően (Xi) kaptam egy, már csak számokból álló elemezhető táblázatot (2. ábra).

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

*2. ábra: Végleges adatok  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„Végleges adatok” munkalap A1-M11 tartománya*

Jelmagyarázat: mértékegységek?

Ezek után össze kellett gyűjteni a programozási nyelvekre vonatkozó népszerűségi adatokat is a Google Trends segítségével. A népszerűségi adatok lekérdezésénél a számításba vehető nyelveket a „[REST API](#_Rövidítések_jegyzéke)” kifejezéssel együttesen kérdeztem le, tehát a találatok kereső kifejezése minden esetben a „[REST API](#_Rövidítések_jegyzéke)” kifejezés, plusz a nyelv megnevezése volt (pl. „[REST API](#_Rövidítések_jegyzéke) Python”). Összesen 10db nyelvet vizsgáltam, melyek a legrelevánsabbak egy [REST API](#_Rövidítések_jegyzéke) fejlesztési projektben. A Google Trends által nyújtott keresési adatok az elmúlt egy évre vonatkozóan a 3. és 4. ábrán láthatók.

A képen szöveg, sor, Diagram, diagram látható

Automatikusan generált leírás

*3. ábra: Google Trends adatok 1.  
Forrás:* [*https://trends.google.com/trends/explore?q=REST%20API%20JAVA,REST%20API%20Python,REST%20API%20JavaScript,REST%20API%20C%23,REST%20API%20Go&hl=hu*](https://trends.google.com/trends/explore?q=REST%20API%20JAVA,REST%20API%20Python,REST%20API%20JavaScript,REST%20API%20C%23,REST%20API%20Go&hl=hu)

Jelmagyarázat: mértékegységek?

A képen szöveg, sor, Betűtípus, Diagram látható

Automatikusan generált leírás

*4. ábra: Google Trends adatok 2.  
Forrás: <https://trends.google.com/trends/explore?q=REST%20API%20PHP,REST%20API%20Ruby,REST%20API%20Swift,REST%20API%20Kotlin,REST%20API%20Rust&hl=hu>*

Jelmagyarázat: mértékegységek?

A Google Trends adatait exportáltam egy CSV fájlba, amit betöltöttem egy EXCELL táblába a további utófeldolgozásra, ahol az éves adatokat átlagoltam (5. ábra).

A képen szöveg, képernyőkép, Párhuzamos látható

Automatikusan generált leírás

*5. ábra: Google Trends CSV  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„Google Trends Internet” munkalap*

Jelmagyarázat: mértékegységek?

A kapott átlagokat felvittem a nyelvek mellé (Y0), majd a táblázat adatait rangsoroltam, hogy az így kapott objektum-attribútum mátrixot (vö: tanulási minta) fel tudjam használni a [COCO Y0](#_Rövidítések_jegyzéke) és [COCO STD](#_Rövidítések_jegyzéke) online robot használatával.

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

*6. ábra: OAM rangsorolt  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„OAM (rangsorolt)” munkalap*

Jelmagyarázat: mértékegységek?

Y0 elvileg konstans, COCO STD Y-ja lehet más-más értékekkel feltöltve, de az Y0 is fut eltérő értékekkel…

# Objektumok

Az objektumok az elemzésben figyelembe vett programnyelvek, összesen 10db nyelv:

1. …
2. …

# Attribútumok

Az attribútumok / tulajdonságok az elemzésben figyelembe vett szempontok(… db):

1. Teljesítmény,
2. Memóriahasználat,
3. Fejlesztési sebesség,
4. Tanulási görbe,
5. Szaktudás,
6. Közösségi támogatás,
7. Operációs rendszer,
8. [REST API](#_Rövidítések_jegyzéke) keretrendszerek száma,
9. Kódolási stílus,
10. Tesztelhetőség,
11. Dokumentáció,
12. [.NET DLL](#_Rövidítések_jegyzéke) integráció,
13. Népszerűségi adatok (Y)

Mindenhol: mértékegység, irány, értelmezési intervallum (max-min-szórás-módusz-medián-…)

# Elemzés 1.

Miután elkészült az objektum-attribútum mátrix ([OAM](#_Rövidítések_jegyzéke)) a végleges adatokkal, itt még nem alkalmaztam a rangsorolást, csupán egy naiv (nem optimalizáló) számítást végeztem annak kiderítésére, vajon melyik nyelv kapná a legtöbb „jóság pontot”. Azaz nyelvenként összegeztem a kapott jóság pontokat 7. ábra (N oszlop), majd rangsoroltam az így kapott eredményt (O oszlop). Az eredményeket színskálával jelöltem, ahol minél zöldebb annál jobb a helyezés.

A képen szöveg, képernyőkép, szám, Párhuzamos látható

Automatikusan generált leírás

*7. ábra: OAM (naív)  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„OAM (naív)” munkalap*

Jelmagyarázat: mértékegységek? Színkódok jelentése?

# Eredmény 1.

A naív megközelítésnél a színskálákat elemezve azt láthatjuk, hogy a dobogós helyezésben a Java és Python harcol az 1. és 3. helyezésért, a PHP stabilan 2. helyezett a népszerűségi adatokkal összevetve. A sereghajtóknál látható szintén helycsere a Ruby és Rust között.

# Elemzés 2.

A következő elemzésem kapcsán a [COCO Y0](#_Rövidítések_jegyzéke) anti-diszkriminatív módszerét alkalmazva kerestem az „ideális” jelöltet. Vagyis lehet-e minden objektum másképpen egyforma? Bemeneti adatként elkészítettem egy módosított [OAM](#_Rövidítések_jegyzéke) adathalmazt, ahol alapul vettem az objektumok tulajdonságait (Xi) jellemző rangsorokat, viszont az Y0 értékként mindenkinek megelőlegeztem 1000 „jóság” pontot. Az így kapott adatokat (vö. 8. ábra) már be lehetett másolni a <https://miau.my-x.hu/myx-free/coco/beker_y0.php> linken elérhető online elemző szolgáltatásba.

*Az anti-diszkriminációs számítások-- más néven az ideálkereső, mesterséges fogalom-alkotó modell jelölése --, ahol minden X esetén az idealitás irányába ható irány megadása után optimalizálás keretében keressük az átlagtól leginkább eltérő objektumot úgy, hogy az optimalizálás célja mindvégig az objektumok azonosságának kikényszeríteni akarása (vö. lehet-e minden objektum másként egyforma – azaz létezik-e az adott fogalom ábrázolási skálája: vö. gázelegy-összetétel idealitás-index). Y0 modell esetén a lépcsők nagysága minimum 1 egység, vagyis a következő lépcső értéke legyen minimum 1 egységgel kisebb, mint az azt megelőző. Az Y vektor ez esetben konstans pl.: 1000 egység. (Tapasztalati érték: Y értéke legyen min. 2 nagyságrenddel nagyobb, mint az objektumok száma – a jelenleg alkalmazott LP-motor specialitásait figyelembe véve.)*

*Forrás:* [*http://www.combustioninstitute.hu/tdk\_dij/2021/PitlikMarcell\_TDK2020.pdf*](http://www.combustioninstitute.hu/tdk_dij/2021/PitlikMarcell_TDK2020.pdf) *(30. oldal)*

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

*8. ábra: OAM (Y0)  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„OAM (Y0)” munkalap A1-N11 tartomány*

# Eredmény 2.

Az online elemzőt futtatva megkaptam a 9. ábrán látható eredményt, ahonnan a becslés oszlopot átmásoltam az eredeti táblázat „P” oszlopába (vö. 10. ábra).

A képen képernyőkép, sor, Párhuzamos, szám látható

Automatikusan generált leírás

*9. ábra: COCO (Y0)  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„COCO (Y0)” munkalap A43-Q53 tartomány*

A képen szöveg, képernyőkép, szám, Párhuzamos látható

Automatikusan generált leírás

*10. ábra: OAM (Y0) eredmény  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„OAM (Y0)” munkalap A1-S11 tartomány*

Ellenőrzésként elvégeztem az inverz becslést is, ez látható a 10. ábra „R” oszlopban. Az eredményeket szintén színekkel jelöltem, ahol a minél zöldebb annál jobb. Jól látható, hogy az elemző szolgáltatás szerint a két legmegfelelőbb nyelv ez esetben a Python és C#, majd ezt követi a Java.

# Elemzés 3.

Utolsó elemzésem kapcsán kíváncsi voltam rá, hogy a Google Trends népszerűségi adatai miként korrelálnak a becsült adatokkal. Vagyis kialakítom a népszerűség termelési függvényét a [COCO STD](#_Rövidítések_jegyzéke) online termelési függvény generátorral és ennek alapján megtudom, hogy valóban a legjobb tulajdonságokkal rendelkező nyelv a népszerűbb.

Bemeneti adatként elkészítettem egy módosított [OAM](#_Rövidítések_jegyzéke) adathalmazt (vö. 11. ábra), ahol alapul vettem az objektumok tulajdonságait (Xi) jellemző rangsorokat, viszont az Y0 értékként a Google Trends népszerűségi adatokat használtam. Az érdeklődési potenciált kifejező pontszám (N oszlop), a nyers Google Trends adatokból a következő képlettel vezethető le: INT(M(oszlop)\*1000+1000). Az így kapott adatokat (vö. 8. ábra) már be lehetett másolni a <https://miau.my-x.hu/myx-free/coco/beker_std.php> linken elérhető online elemző szolgáltatásba.

*A COCO-STD modelljében rejlő lehetőség, hogy valós Y-változót (pl. árat) az X-változók lépcsős függvényeként közelítjük, vagyis minden X változóhoz meghatározunk egy-egy lépcsős függvényt pl. ár/érték-arány-elemzés érdekében, ahol az X-tömb csak pozitív egész számokat, azaz rangsor-számokat tartalmaz. A COCO-STD modell esetében a lépcsőkre vonatkozó megkötése: a következő lépcső legyen kisebb egyenlő, mit az azt megelőző. Az Y (oszlop) vektor az attribútumokat (O sor vektorok) leíró paraméterek közül az egyetlen kitüntetett (pl.: ár).*

*Forrás:* [*http://www.combustioninstitute.hu/tdk\_dij/2021/PitlikMarcell\_TDK2020.pdf*](http://www.combustioninstitute.hu/tdk_dij/2021/PitlikMarcell_TDK2020.pdf)

*(30. oldal)*

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

*11. ábra: OAM (STD)  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„OAM (STD)” munkalap A1-N11 tartomány*

# Eredmény 3.

Az online elemzőt futtatva megkaptam a 12. ábrán látható segédtáblákat, ahonnan a becslés oszlopot átmásoltam az eredeti táblázat „O” oszlopába (vö. 13. ábra).  
 A képen képernyőkép, szöveg, szám, sor látható

Automatikusan generált leírás

*12. ábra: COCO (STD)  
Forrás:*[*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„COCO (STD)” munkalap A105-Q115 tartomány*

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

*13. ábra: OAM (STD) eredmények  
Forrás:* [*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„OAM (STD)” munkalap*

A kapott eredményeket színskálával jelöltem (vö. 13. ábra), ahol a minél zöldebb annál jobb. A segédtáblázatoknak (12. ábra becslés oszlop) hála az O oszlopban készült egy becslés, amely megadja a programozási nyelvek vélhetően valós értékét a korábban osztályozott tulajdonságaik alapján. Az N oszlop (Y0) eredeti Google Trends adatok és az O oszlop becsült adatok között ellenőriztem a korrelációt. A korreláció jelzi két tetszőleges érték közötti lineáris kapcsolat nagyságát és irányát (avagy ezek egymáshoz való viszonyát). A korrelációs együttható értéke -1 és +1 között változik. Esetünkben ez az érték +0.946184 , amely majdnem tökéletes pozitív korreláció, tehát a két változó értékei együtt emelkednek és süllyednek. Az eredményekből látható, hogy a népszerűségi adatok és a becslés alapján a dobogós helyezetek azonos nyelvek között oszlik meg, csak a sorrend más. A népszerűségi adatok alapján a Java még mindig kicsit túlértékelt és a PHP -t pedig kezdik elfeledni, ez nyilván köszönhető a Java mögött álló Oracle cégnek és kampányának.

# ChatGPT vélemény

Végezetül kíváncsi voltam, hogy a ChatGPT által szolgáltatott adatok alapján az AI mit javasol, melyik nyelvet használjam a projektemben. Így megkérdeztem tőle pontosan specifikálva a saját szempontjaimat (vö. 14. ábra).

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

*14. ábra: ChatGPT vélemény  
Forrás:* [*rest\_api\_programnyelvek.xlsx*](rest_api_programnyelvek.xlsx) *„ChatGPT” munkalap*

A válasz, miszerint a C# nyelv a megfelelő érthető volt, hiszen a .NET DLL integrálhatóság fontos szempont, illetve a Windows operációs rendszeren való futtatás lehetősége erősen a Microsoft irányába tereli a választást. Ettől függetlenül a korábbi anti-diszkriminatív elemzésem is a C# és Python nyelvet hozta ki a dobogó élére.

# Vita

Végső soron a legmegfelelőbb nyelv kiválasztása a projekt specifikációjától, a csapat preferenciáitól és a platformtól függ. Jelen dolgozatban specifikusan a saját szakdolgozatomban érintett szempontokat vettem figyelembe, ezért más projekteknél értelemszerűen más tulajdonságok is számba vehetők. Ettől eltekintve maga a módszer alkalmas más fejlesztési projektek esetében a megfelelő eszközök kiválasztására.

# Következtetések

Az elemzés megmutatta, hogy a trendek még mindig képesek felülbírálni az objektív tényeket. A népszerűségi adatok alapján a Java kicsit túl van értékelve a kevesebb marketinggel rendelkező Python vagy PHP nyelvvel szemben. Kezdő programozóként számomra egyértelműen kiderült, hogy a Python nyelv lesz a befutó, hiszen az én projektem és tudásom szempontjából ideális jelölt.

# Jövőkép

Dolgozatom elemzését felhasználva immár tudom, hogy leendő szakdolgozatom kapcsán mely programnyelvet célszerű alkalmaznom. Ezen információk használatával további elemzéseket is el tudok majd végezni célom elérése érdekében.

# Mellékletek

## Rövidítések jegyzéke

COCO (Component based Object Comparison for Objectivity): hasonlóságelemzés. Optimalizálás keretében minden tulajdonság esetén minden rangsorszámhoz hozzárendelünk egy konkrét értéket (lépcsőt), amennyivel az adott tulajdonság hozzájárul a függő változó értékéhez.

OAM: Objektum Attribútum Mátrix, azaz egyfajta tanulási minta.

COCO STD: a COCO modul alapváltozata, amely objektumok ár-teljesítmény számításra képes, Y-ként minden esetben szükséges valós árat megadni.

COCO Y0: a COCO modul Y0 változata, amely teljesítmény elemzésre használható, minél nagyobb a kapott pont, annál jobb a teljesítmény. Az Y mindig fiktív és azonos.

REST API (Representational State Transfer Application Programming Interface.): Egy állapotmentes (stateless) szoftver architektúra megoldás, ahol a meglévő webes standardokat használjuk ki az alkalmazás-alkalmazás kommunikációban a szerveren vagy kliensen tárolt objektumok állapotának átvitelére. Tipikusan a HTTP szabványban található szabályokat követi: HTTP státuszkódok, HTTP igék és HTTP fejléc kezelés.

ChatGPT: A ChatGPT a Generative Pre-trained Transformer angol rövidítése, ami egy nagy nyelvi modellt takar, amelyet az OpenAI fejlesztett ki, és képes emberi szintű szöveget generálni és párbeszédet folytatni.

.NET DLL (Dynamic Link Library): a .NET Framework keretrendszerben használt fájlformátum. A DLL fájlok olyan modulokat tartalmaznak, amelyekben kód és adat található, és amelyeket több program is használhat egyszerre.

## Referenciák

Kozma Viktor (2023.09.11) Signature pads OAM XLSX Forrás: <https://miau.my-x.hu/bprof/2023/osz/rapid/signature_pads_v2.xlsx>

Kozma Viktor (2024.04.15) REST API programnyelvek OAM XLSX Forrás: <rest_api_programnyelvek.xlsx>

Dr. Pitlik László. (dátum nélk.). COCO online elemző szolgáltatás weblapja, Forrás: <https://miau.my-x.hu/myx-free/coco>

REST API a szoftver infrastruktúra nyelve (dátum nélk.) Forrás: <https://www.opengov.hu/tervezesi-elvek/restapi.html>

Pitlik Marcell. (2020.). Térfogati égés optikai vizsgálata és mesterséges intelligencia alapú elemzése. Tudományos Diákköri Konferencia Budapest, 2020. Letöltés dátuma: 2024.04.13, Forrás: <http://www.combustioninstitute.hu/tdk_dij/2021/PitlikMarcell_TDK2020.pdf>

Google Trends (dátum nélk.) Forrás: <https://trends.google.com/trends?geo=&hl=hu>

Eric J. The 15 Best Languages for REST API: A Complete Exploration (dátum nélk.) Forrás: <https://datarundown.com/rest-api/>

## Ábrák

1. ábra: Gemini nyers adat
2. ábra: Végleges adatok
3. ábra: Google Trends adatok 1.
4. ábra: Google Trends adatok 2.
5. ábra: Google Trends CSV
6. ábra: OAM rangsorolt
7. ábra: OAM (naív)
8. ábra: OAM (Y0)
9. ábra: COCO (Y0)
10. ábra: OAM (Y0) eredmény
11. ábra: OAM (STD)
12. ábra: COCO (STD)
13. ábra: OAM (STD) eredmények
14. ábra: ChatGPT vélemény