МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

НА ТЕМУ

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма «**Аналітика даних»**

Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи АнД- 41

Бережний Д.А.
(прізвище та ініціали)

Керівник Гнатієнко Г. М.
(прізвище та ініціали)

к.т.н.
(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*
Протокол № 11 від 06.06.2022 р.
зав. кафедри_____ доц. Іларіонов О.Є.

Київ - 2022

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.
_____
"___"_____2021 р.

ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Бережному Дмитру Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи)
Система визначення оцінки наукової події в контексті інтересів організації

затверджена протоколом засідання кафедри від « » жовтня 2021 р. №

2. Термін здачі студентом закінченого проекту (роботи) 31 травня 2022 року

3. Вихідні дані до проекту (роботи)
Система визначення оцінки наукової події в контексті інтересів організації створюється для визначення статусу наукової події в контексті інтересів певної організації.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
Вибір та аналіз предметної області, дослідження видів наукових подій та їх характеристик, огляд методів визначення статусу періодичних видань у Scopus, WoS, CEUR, проєктування архітектури системи, розробка системи, вибір тестових задач, візуалізація результатів, тестування роботи системи

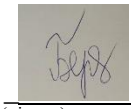
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
Огляд джерел інформації щодо наукових подій та періодичних видань, ілюстрація ситуації прийняття рішення, архітектура системи, програмна реалізація системи, презентація Power Point

6. Консультанти з випускної кваліфікаційної роботи із зазначенням розділів випускної кваліфікаційної роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Вибір та аналіз предметної області		
1	Постановка задачі визначення статусу наукової події		
2	Проєктування архітектури системи		
3	Програмна реалізація системи		
3	Вибір тестових задач, тестування роботи системи		

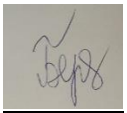
7. Дата видачі завдання 25 січня 2022 року

Керівник  / /
(підпис) (ПІБ)

Завдання прийняв до виконання  / Бережний Д.А. /
(підпис) (ПІБ)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1.	Вибір та аналіз предметної області	25.01-01.02	
2.	Постановка задачі визначення статусу наукової події	02.02-14.02	
3.	Проєктування архітектури системи	15.02-07.03	
4.	Програмна реалізація системи	08.03-25.04	
5.	Вибір тестових задач, тестування роботи системи	26.04-07.05	

Студент-дипломник  / Бережний Д.А. /
(підпис) (ПІБ)

Керівник випускної кваліфікаційної роботи  /
/

Анотація

Бережний Дмитро Андрійович виконав випускну кваліфікаційну роботу на тему «Система визначення оцінки наукової події в контексті інтересів організації» за спеціальністю 122 – «Комп'ютерні науки».

У випускній кваліфікаційній роботі проведено аналіз сучасних наукових подій, досліджено види наукових подій та їх характеристики, розроблено систему визначення оцінки наукової події в контексті інтересів організації, що визначає статус наукових подій в контексті інтересів певної організації.

Ключові слова: наукова подія, система, статус, парсинг, CEUR

Summary

The degree project: «The system of determining the assessment of a scientific event in the context of the interests of the organization» has completed by **Berezhnyi Dmytro** specialty 122 – «Computer Scienses».

In this graduation thesis the modern scientific events are analyzed, the types of scientific events and their characteristics are considered, the system is developed, that determines the status of a scientific event in the context of the interests of the organization.

Keywords: scientific event, system, status, evaluation, parsing, CEUR

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTML – HyperText Markup Language

CSV - Comma-Separated Values

JSON - JavaScript Object Notation

IT - Information Technology

1.1. WoS - Web of Science

БД – база даних

ФІТ – Факультет Інформаційних Технологій

SQL – Structured Query Language

XLS – розширення эксель файлу

RSS - Rich Site Summary

PDF - Portable Document Format

ЗМІСТ

ВСТУП

Актуальність теми. Наука – це певна сфера діяльності людини, головною метою якої є систематизація нових знань. Поняття "наука" включає в себе як діяльність, спрямовану на здобуття нового знання, так і результат цієї діяльності – суму здобутих наукових знань, що є основою наукового розуміння світу. Наразі наука розвивається великими кроками, оскільки в сучасному світі існує необхідність наукового підходу до всіх видів людської діяльності. Критичний аналіз фактів, збирання, оновлення інформації дозволяють будувати причинно-наслідкові зв’язки між явищами та прогнозувати їх перебіг. Однією з найважливіших проблем - є проблема визначення наукового рівня події у потужному потоці інформації. Кількість проведення різноманітних наукових заходів, публікацій, рецензій наукових робіт зростає з кожним днем, отже і зростає ймовірність появи неякісних, неактуальних, недоцільних статей, досліджень [4].

Необхідність використання системи визначення оцінки наукової події в контексті інтересів організації обумовлена такими факторами:

- велика кількість наукових робіт, що не відповідає стандартам якості, може негативно вплинути на рейтинг автора та прагнення решти науковців до забезпечення високої якості матеріалу. Це може призвести до стагнації, або навіть деградації навколишнього середовища, оскільки високий рівень якості проведення наукових заходів забезпечує розвиток науки. Якщо розвивається наука – розвиваються технології, а отже і розвивається повсякденне життя;
- за допомогою даної системи можна відслідковувати загальний рівень наукової компетентності певної організації.

Об’єктом дослідження є наукові події, представлені на веб-сайті http://ceur-ws.org/.

Предметом дослідження є методи та алгоритми парсингу інформації з веб-сайтів та документів .

Мета роботи: розробка системи визначення оцінки наукової події в контексті інтересів організації, в якій приймали участь науковці з факультету, для визначення статусу наукової події в контексті інтересів певної організації.

У розділі 1 було проведено аналітичний огляд наукометричих баз та метрик, за якими проводиться оцінка наукових матеріалів.

У розділі 2 даної випускної кваліфікаційної роботи було визначено програмний засіб для обробки інформації для створення системи визначення оцінки наукової події в контексті інтересів організації – веб-парсинг . Також, було розроблено діаграми різних типів для проектування розробки системи. Також була визначена мова програмування та інші програмні засоби для програмної реалізації системи.

У розділі 3 було детально описано код програми, інтерфейс програми та проведено тестування розробленої системи. Під час аналізу результатів було встановлено, що розроблена система працює коректно.

В подальшому робота може використовуватись для автоматизованого визначення рівня наукових подій та контролю рівня наукової компетентності факультетів, або університетів загалом, пов’язаних з ІТ-спеціальностями.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД

1.1. Вибір та аналіз предметної області

Наукова подія – це організований захід, де вчені або дослідники обговорюють, презентують, діляться між собою своїми науковими роботами. Проведення наукових подій є надзвичайно важливим для обміну інформацією, досвідом між вченими. Слід зазначити, що конкретних правил проведення наукових подій не існує, зазвичай певні норми проведення встановлюються оргкомітетами наукових заходів. Проте існують певні методичні рекомендації: наприклад у міжнародній конференції приймати участь має не менше п’яти країн з кількістю учасників – понад сто осіб. [1, 4]

Нині проводиться багато різноманітних наукових заходів: конгреси, симпозіуми, форуми, конференції, семінари, виставки тощо. Наукові події прийнято диференціювати за такими ознаками [1]:

1) За напрямком:
   1.1) науково-теоретичні;
   1.2) науково-технічні;
   1.3) науково-практичні.
2) За масштабом охоплення території:
   2.1) міжнародні;
   2.2) республіканські;
   2.3) міжрегіональні;
   2.4) регіональні.
3) За складом учасників:
   3.1) молодіжні;
   3.2) для зрілих вчених.
4) За періодичністю:
   4.1) одноразові;
   4.2) щорічні.
5) За тематикою проведення:

5.1) наукові події широкої тематики;

5.2) вузькоспеціалізовані наукові події.

Для оцінювання результативності наукової діяльності важливе місце відводиться наукометрії .

Наукометрія – галузь наукознавства, головною метою якої є вивчення еволюції науки за допомогою чисельних вимірювань наукової інформації, за певними наукометричними показниками [2, 6]. До основних таких показників відносяться:

- кількість публікацій;
- частота цитованості публікацій;
- імпакт-фактор наукового журналу, в якому вони публікуються;
- кількість отриманих грантів;
- активність у міжнародних наукових співробітництвах.

Головні задачі наукометрії вирішують спеціалізовані інститути та інформаційні служби, що створюють наукометричні бази, які ведуть дослідження про публікаційну активність науковців та цитованість авторів [3]. До найбільш відомих наукометричних баз відносяться Scopus та WoS [24, 25].

Scopus – це реферативна і бібліографічна база даних, що є інструментом для відстеження цитованості статей, що опубліковані у наукових виданнях Станом на січень 2017 року, Scopus містив понад 50 млн. записів. Scopus належить корпорації Elsevier, яка є розробником Scopus. Взагалі, Scopus індексує серіальні книжкові видання, матеріали конференцій та наукові журнали і являється найбільшою у світі БД, яка дає змогу відстежувати наукову цитованість, за яку відповідає індекс цитувань. Проте, слід зазначити, що повна база даних доступна лише за умови передплати.

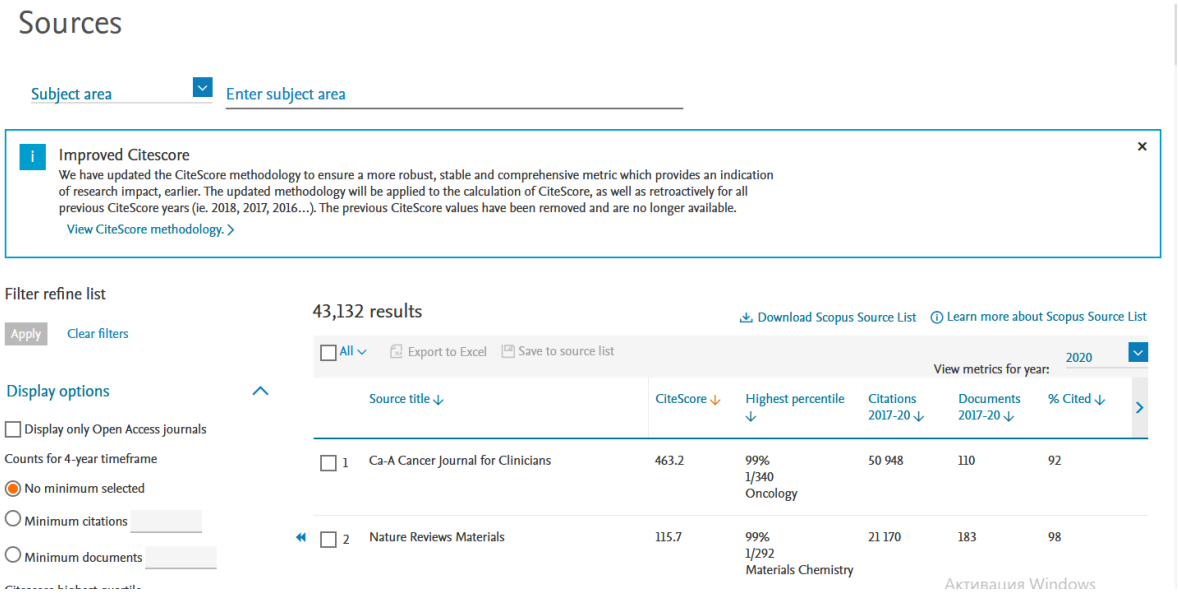


Рис.1.1. Стартова сторінка Scopus

Scopus класифікує научні матеріали за тематичними розділами. Основними серед них слід виділити наступні розділи:

- Комп'ютерні науки;
- Хімія;
- Математика;
- Енергетика;
- Виробництво;
- Астрономія і фізика;
- Медицина та стоматологія;
- Нейронауки;
- Психологія;
- Бізнес, менеджмент та бухгалтерський облік.

У Scopus також наявний інструмент Journal Analyzer, що дає можливість проводити широкий аналіз наукового рівня видань за чотирма показниками:

- загальна кількість статей, опублікована певним виданням протягом року;
- загальна кількість посилань на певне видання, наявне у інших виданнях;
- процент статей, що не були процитовані протягом року;

- тренд року – це відношення кількості посилань на видання до кількості статей, опублікованих у видані.

Для профілів авторів і установ на Scopus, які опублікували більше ніж однієї статті, створюються унікальні профілі з унікальними ідентифікаторами авторів або установ, відповідно.

Scopus індексує наукові джерела на різних мовах, проте необхідною умовою є наявність англомовної версії реферату. За останньою статистикою, географічне охоплення видавців розподілено наступним чином:

- Північна Америка – 36%;
- Південна Америка – 3%;
- Азійсько-Тихоокеанський регіон – 9%;
- Європа, Середній схід, Африка – 52%.

Індекс цитувань – це прийнята в науковому світі міра «значущості» праць якого-небудь ученого. Величина індексу визначається кількістю посилань на цю працю (або прізвище) в інших джерелах. Основним таким індексом являється коефіцієнт впливовості, або імпакт-фактор. В якості індекса цитувань Scopus використовує CiteScore.

Коефіцієнт впливовості - це коефіцієнт співідношення цитування наукових журналів. Для обчислення даного коефіцієнту використовується лише БД «Journal Citation Reports». Даний показник обчислюється за 3 роки, або за 5 років і його можна вважати, як усереднене співвідношення між кількості цитувань статей у журналі у поточному році, до загальної кількості статей, надрукованих у цьому ж журналі за попередні два роки. Наприклад, імпакт-фактор журналу за 2003 рік обчислюється так:

$2003_{\text{коефіцієнт впливовості}} = A/B$, де A - кількість цитувань статей надрукованих протягом 2001 — 2002 рр. в журналах за 2003 рік; B - загальна кількість «статей, на які можна посилатись» (зазвичай статті, доповіді з конференцій) надрукованих протягом 2001 — 2002 рр.

CiteScore - показник, що показує середньорічну кількість цитат останніх статей, опублікованих у цьому журналі. Тобто, індекс n-го року визначається,

як кількість цитувань, зроблених у цей рік і в три попередні, з документів, опублікованих у журналі за ці чотири роки, поділену на загальну кількість опублікованих документів того ж типу у базі даних протягом цього ж чотирирічного періоду:

$$\mathrm{CS}_y = \frac{\mathrm{Citations}_y + \mathrm{Citations}_{y-1} + \mathrm{Citations}_{y-2} + \mathrm{Citations}_{y-3}}{\mathrm{Publications}_y + \mathrm{Publications}_{y-1} + \mathrm{Publications}_{y-2} + \mathrm{Publications}_{y-3}}$$

Коефіцєнт **CiteScore** був створений задля конкуренції з імпакт-фактором. Таблиця порівняння CiteScore з коефіцєнтом впливовості:

Параметр	**Імпакт-фактор**	**CiteScore**
БД	JCR	Scopus
Кількість індексованих журналів	11000	22000
Доступ	Підписники	Вільний доступ
Оцінювання	Статті, огляди	Всі публікації

Табл.1.1. Порівняльна таблиця імпакт-фактору та CiteScore по певним параметрам

Ще однією відомою базою даних для зберігання наукових робіт являється Web of Science.

Web of Science (WoS) - це платформа, що зберігає бази наукової літератури і патентів. Основною частиною плафторми є БД Web of Science Core Collection WoS (CC), яка містить понад 18000 журналів, що розташовані у трьох ключових індексах наукової літератури:

- SCIE (Science Citation Index Expanded) – архів з 1900-го року, що містить 8300 журналів;
- SSCI (Social Science Citation Index) - архів з 1900-го року, що містить 2900 журналів;

- AHCI (Art and Humanities Citation Index) - архів з 1975-го року, що містить 1600 видань.

WoS має вбудовані інструменти для пошуку, аналізу та управління інформацією. Для визначення коефіцієнту впливовості у WoS застосовується ряд наукометричних показників, таких як: число публікацій, індекс Хірша та ін.). З точки зору повноти оцінки індекса цитування, до недоліків сервісу Web of Science можна віднести те, що статті враховуються переважно англійською мовою.

Індекс Хірша (h-індекс) – показник впливовості, який базується на кількості публікацій та їх цитуваннях. h-індекс науковця дорівнює h якщо він є автором h публікацій, кожна з яких була процитована щонайменше h разів. Наприклад, якщо науковець - автор 5 публікацій, 3 з яких процитовано по 3 рази, а інші 2 — по 1 разу, то його h-індекс дорівнює 3. Якщо науковець є автором 5 публікацій, 1 з яких процитована 100 разів, а інші 4 — по 1 разу, то його h-індекс дорівнює 1.

Після обрахунку індекса цитованості, наукові журнали ранжуються по 4 квартилям [7]: від $\mathbf{Q_1}$ (найвищий, до якого належать найавторитетніші журнали) до Q4 (найнижчий). Журнали Q1 мають 75,1-100% (процентиль «успіху»), Q2 – 50,1-75%, Q3 – 25,1-50% та Q4 – 0,1-25%. Система квартилів дозволяє найбільш об'єктивно оцінити рівень журналу незалежно від предметної області.

Слід зазначити, що для авторів є декілька способів підвищення наукометричних показників, таких як:

- друкування в індексованих за кордоном журналах;
- друкування з іноземними авторами;
- використання іноземних посилань;
- приділяння особоливої уваги щодо оформлення назви, анотацій, ключових слів, списків використаної літератури.

Для опублікування наукового заходу у таких наукометричних базах як Scopus або WoS і перетворення його у наукову подію, необхідно відповідати певним вимогам, які є передумовами визначення рейтингу наукової події :

- якість оформлення матеріалів наукового заходу;
- наповнення та наукова вага матеріалів наукового заходу;
- проміжок часу, протягом якого науковий захід перетворюється у наукову подію;
- мови наукового заходу, що безпосередньо впливають на міжнародне визнання;
- міра авторитетності наукового заходу: кількість цитувань учасників наукового заходу у WoS/Scopus, авторитетність редакційної колегії, кількість публікацій у DBLP, авторитетність членів Програмного комітету наукового заходу;

Усі передумови мають бути виконані. Також являється недопустимим крадіжка та плагіат наукових матеріалів, публікацій. Це відноситься до прямого порушення публікаційної етики.

Для відслідковування наукових подій, в якій брали науковці з факультету ФІТ, був обран сайт CEUR [9]. CEUR – це сайт, що надає безкоштовні матеріали відкритого доступу, що пов’язані з інформаційними технологіми (на відміну від Scopus та Web of Science (WoS), які публікують матеріали з різних галузей), для наукових конференцій та семінарів.

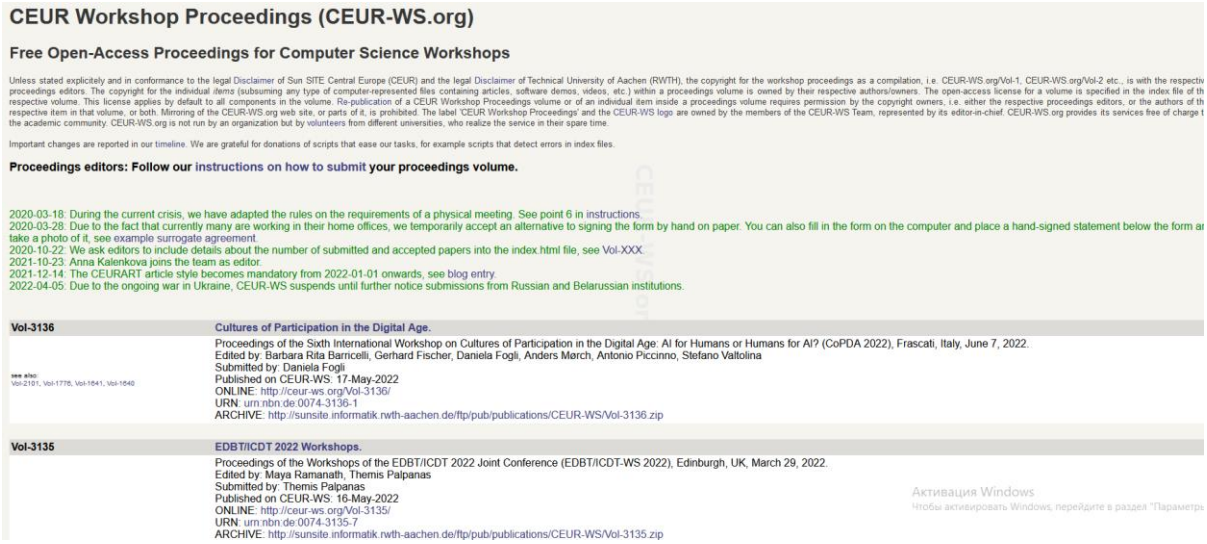
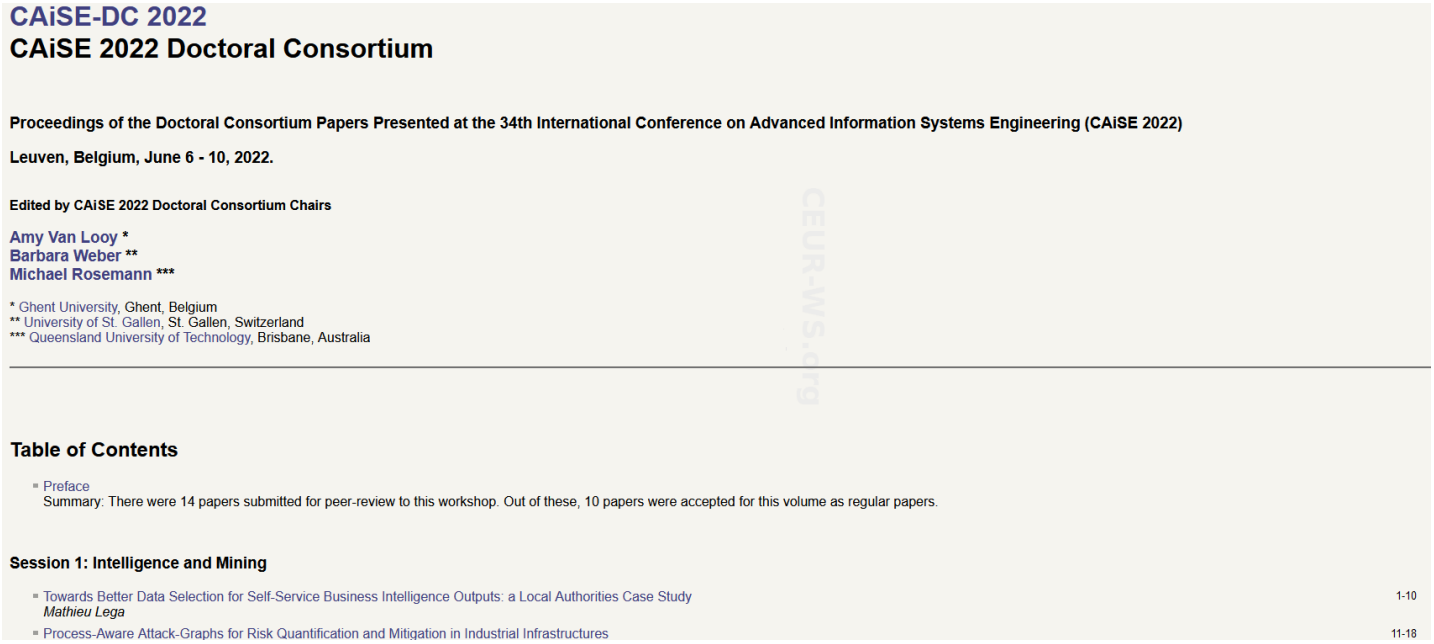


Рис.1.2. Стартова сторінка CEUR

На головній сторінці можна побачити список наукових подій, які позначаються “Vol-номер наукової події”. Про кожну подію наявна певна загальна інформація: дата проведення наукової події, хто приймає участь у даній події, дата завантаження даної події на CEUR, особи, які редагували статті. Також присутнє посилання на сторінку самої події, можливість завантаження архіву з матеріалами певної наукової події.



CAiSE-DC 2022

CAiSE 2022 Doctoral Consortium

Proceedings of the Doctoral Consortium Papers Presented at the 34th International Conference on Advanced Information Systems Engineering (CAiSE 2022)

Leuven, Belgium, June 6 - 10, 2022.

Edited by CAiSE 2022 Doctoral Consortium Chairs

Amy Van Looy *
Barbara Weber **
Michael Rosemann ***

* Ghent University, Ghent, Belgium
** University of St. Gallen, St. Gallen, Switzerland
*** Queensland University of Technology, Brisbane, Australia

CEUR-WS.org

Table of Contents

Рис.1.3. Окрема сторінка наукової події, представленої на CEUR

На сторінці самої події вже можна побачити більш детальну інформацію зокрема список статей і їх авторів; університети науковців, які проводили редагування статей.

Щоб перевірити відношення до інформатики, CEUR може попросити редакторів надати їм дані про кількість публікацій авторів і учасників наукової події у бібліографії DBLP. Як правило, CEUR вимагає , щоб у кожній статті був принаймні один автор із принаймні 5 роботами, зазначеними в DBLP [10].

DBLP – це бібліографія, що надає відкриту бібліографічну інформацію про основні журнали та матеріали з інформатики. Головна мета DBLP – це підтримання дослідників інформатики, надаючи безкоштовний доступ до посилань та метаданих на електронні видання публікацій. Станом на січень

2019-го, приблизно 4,4 мільйона публікацій проіндексовано у DBLP, авторами яких являються більше ніж 2,2 мільйона науковців.

Також DBLP надає на своєму сайті різну інформаційну статистику:

1) Діаграма, що показує розподіл типу публікацій:

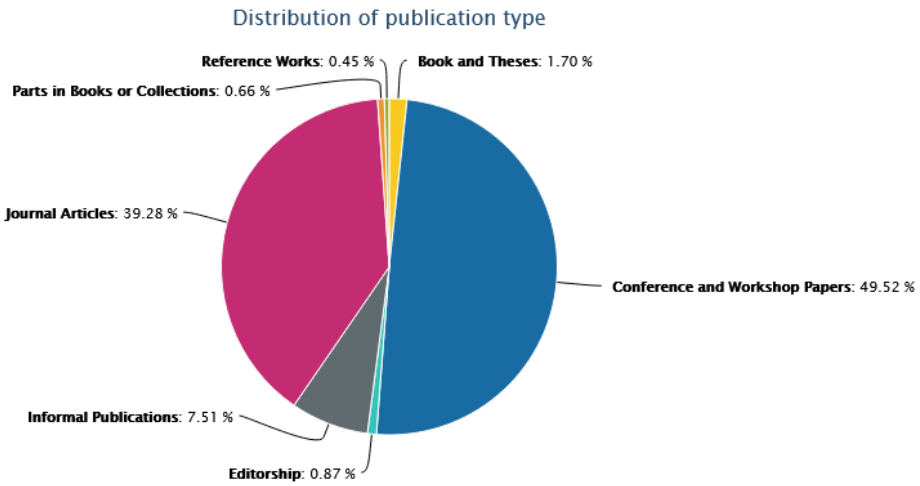


Рис. 1.4 – діаграма розподілу типу публікацій у DBLP

Як видно, найбільшу частку серед усіх публікацій, займають матеріали, що стосуються конференцій.

2) Діаграма, що показує кількість публікацій за рік:

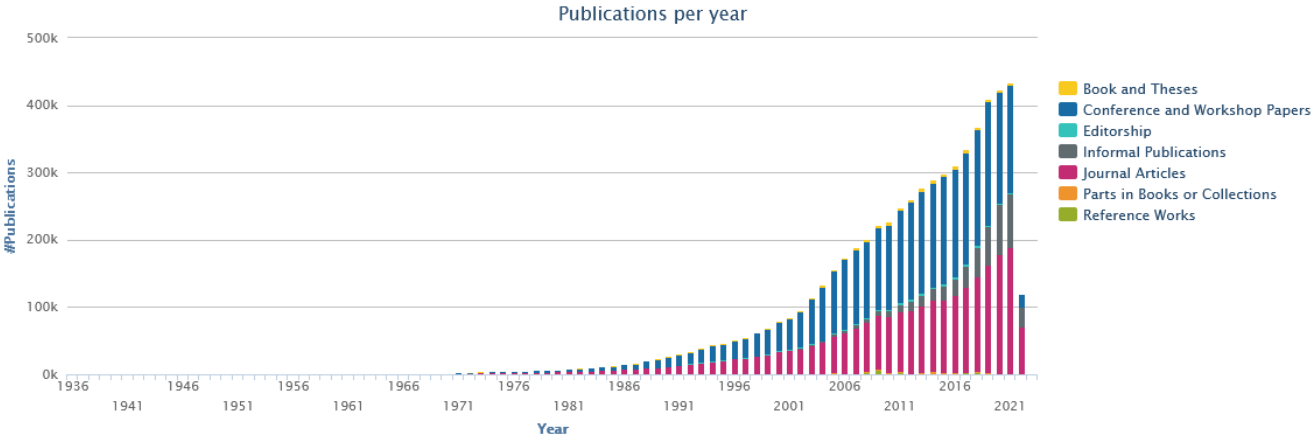


Рис.1.5. Діаграма кількості публікацій за рік у DBLP

Як видно, спостерігається значний спад кількості публікацій у 2021-ому році.

1.2. Підходи до визначення якості наукової події

Обрахунок імпакт-фактора наукового журналу досить схоже з визначенням ефективності інших наукових подій, проте серед наукових заходів існує багато різноманіття, а наукові події, як правило, не є регулярними [1].

Для того, щоб знайти адекватний інтегральний показник ефективності наукової події, необхідна побудова адекватної математичної моделі. Можна виділити три основні підходи до визначення якості наукової події:

1) Експертний підхід. Характеристики (атрибути) та інтервали зміни параметрів моделі встановлюються експертним шляхом.

2) Динамічний підхід. У цьому підході атрибути та інтервали значень характеристик наукової події змінюються по мірі наповнення бази даних. Слід зазначити, що інтегральний індекс якості проведення наукової події може бути визначений у різних шкалах: фіксованим числом, у вигляді інтервалу або у вигляді функції належності нечіткій множині. Також зазначені види інтегрального індексу ефективності проведення наукової події можуть бути виміряні у різних діапазонах.

3) Статичний підхід. Коли БД наукових подій є достатньо заповненою, то має місце використання статичного підходу для визначення ефективності результативності наукової події. У цьому разі значення параметрів математичної моделі та границі атрибутів наукової події залишаються константними при різних ситуаціях прийняття рішень.

1.3. Постановка задачі створення системи визначення оцінки наукової події в контексті інтересів організації

Для забезпечення роботи системи визначення оцінки наукової події в контексті інтересів організації необхідно, щоб система приймала на вхід список наукових подій з веб-сайту CEUR, а на виході формувала список наукових подій, де приймали участь викладачі факультету з обрахованою вагою (рис. 1.3).

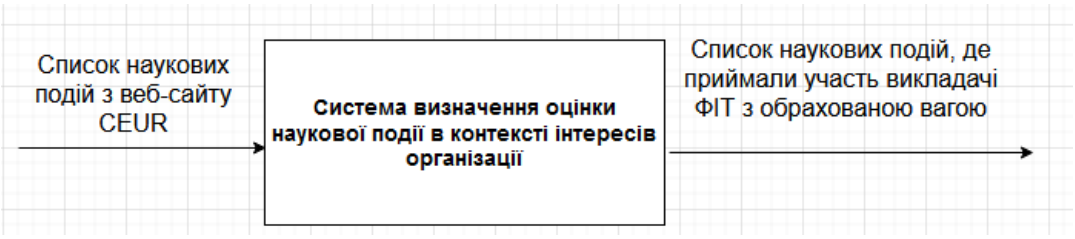


Рис 1.6. Чорна скриня системи визначення оцінки наукової події в контексті інтересів організації

Для створення математичної моделі системи визначення оцінки наукової події в контексті інтересів організації, було прийнято рішення керуватися експертним шляхом, тобто основні параметри моделі встановлюються певним експертом. Основними параметрами при створенні даної моделі було визначено:

- вага наукової статті, серед авторів якої фігурують викладачі ФІТ;
- вага цитувань викладачів ФІТ у матеріалах самої статті.

Було позначено вагу однієї наукової статті серед авторів яких фігурують викладачі ФІТ – w, кількість статей - q_w, вагу цитування, де фігурують викладачі ФІТ - v , к-ть цитувань – q_v, вагу наукової події - I. Тоді, математична модель ваги наукової події матиме такий вигляд:

$$I = q_w * w + q_v * v$$

Дана формула буде використовуватися для обрахунку ваги наукової події.

РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

2.1. Теоретичні відомості про веб-парсинг

Веб-парсинг — це процес автоматичного збору текстових даних з сайтів та їх структурування. Спеціальні програми або сервіси-парсери проходяться по сайту та збирають дані, які відповідають певним вимогам [11, 12, 13].

Наприклад, власник інтернет-магазину хоче швидко зібрати дані про магазини-конкуренти. Його цікавить інформація з карток товарів. Власник хочете зрозуміти, як їх заповнюють конкуренти, що вони роблять краще нього. Він визначає, яка інформація з сайтів йому потрібна і за допомогою певної програми парсить потрібну йому текстову інформацію у зручному вигляді. Це може бути назва товару, його ціна, категорія і опис. Далі структуровану інформацію досить зручно аналізувати і, наприклад, вирішити, яку ціну встановити для свого асортименту.

Взагалі, парсинг даних можна розділити на такі основні етапи:

- Збір контенту

Зазвичай в програму для парсинга завантажується код сторінки сайту. І з ним уже працює спеціальний скрипт – розбиває весь код на лексеми, аналізує, яка інформація потрібна користувачеві.

- Витяг інформації

Це найважливіший етап. У більшості випадків користувачеві не потрібна вся інформація зі сторінки. Наприклад, користувача цікавить лише певна категорія товарів. Парсер має знаходити в коді сторінки лише то місце, де вказана бажана категорія товару і витягне відповідну інформацію. Для витяга інформації можна використовувати регулярні вирази, але частіше використовуються спеціалізовані бібліотеки.

- Збереження результатів

Очевидно, що коли вся потрібна інформація витягнута, її потрібно зберегти. Зазвичай, дані оформлюються у вигляді таблиці, щоб було наочне уявлення. Можна також вносити дані у базу даних, як зручніше буде аналітику.

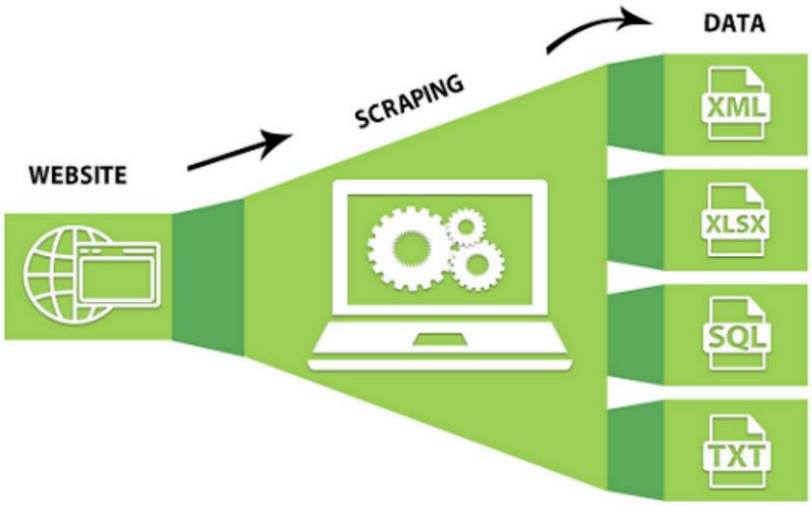


Рис.2.1. Схематичне зображення процесу парсингу даних

На рисунку 2.1. описується схематичне зображення процесу парсингу даних, що складається з витягування інформації з вебсторінки (website), обробки інформації (scraping) та представлення обробленої інформації у певному форматі.

Інформацію, вилучену внаслідок веб-скрепінгу, необхідно представити у зручному форматі для подальшого використання. Експортувати дані можна представити у багатьох форматах, серед яких слід виділити:

- CSV формат. Це файловий формат, що довзоляє зручно представити інформацію у вигляді таблиць, поля якої розмежовані комою.
- SQL формат, який використовується для роботи з реляційними базами даних. Зазвичай дані, які зберігаються у такому форматі призначені для подальшого експорту на інші веб-ресурси. Проте, зазвичай SQL формат згодом необхідно конвертувати у JSON або CSV формат.
- JSON формат – текстовий формат обробки даних, який розроблений на мові програмування JavaScript. Переважно даний формат використовується, коли існує необхідність передавання структурованої інформації через мережу.
- XLS формат – формат файлів Microsoft Excel.

- RSS формат – формат, що використовується для інформації, що має властивість часто змінюватися, наприклад нові записи у блозі.

Обраний формат для експорту даних у даній системі - CSV формат.

Слід зазначити, що парсинг може використовуватися як на благо, так і на шкоду. Парсинг дозволяє проаналізувати великі обсяги текстової інформації, використовуючи автоматизований процес. В той же час, за допомогою парсингу є ризик крадіжки унікального контенту з сайту, певної конфіденційної інформації, що може потрапити до рук шахраїв. Щоб запобігти цього, використовуються певні методи захисту сайтів від парсингу.

3) Розмежування прав доступу. Це найпростіший метод, що включає в собі приховання інформації про структуру сайту, наприклад зробити її доступною лише адміністратором сайту.
4) Створення чорного і білого списку користувачів. У чорному списку знаходяться користувачі, що порушили правила поведінки сайту, крадучи контент та ін. Білий список, відповідно, для добропорядних користувачів.
5) Використання методів захисту від роботів, таких як: капча, підтвердження реєстрації, тобто те, що зможе виконати людина, але не зможе виконати робот.

2.2. Аналіз структури веб-сторінки CEUR

Як було визначено в 1-ому розділі, об’єктом дослідження даного дипломного проекта є список наукових подій з сайту http://ceur-ws.org/ [9]. Оскільки витяг інформації з веб-сторінки є найважливішим етапом при побудові парсера даних, необхідно коректно витягнути потрібні користувачеві дані зі сторінки, потрібно розуміти як влаштовані веб-сайти. Майже всі сайти зверстані за допомогою мови тегів HTML. Більшість сучасних браузерів дає можливість подивитися на програмний код сторінки.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="Type=text/html;charset=utf-8">
<meta name="description" content="CEUR-WS.org provides free online scientific papers">
<meta name="keywords" content="open access, open archive, free scientific paper, workshop proceedings, online publishing, computer science, information systems" >

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="alexaVerifyID" content="NltdqtYOc70F5YTk10N8U9p0egM">
<META HTTP-EQUIV="expires" CONTENT="Wed, 26 Feb 1997 08:21:57 GMT">
<meta http-equiv="X-UA-Compatible" content="IE=8,9,10">
<LINK rel="stylesheet" type="text/css" href="./ceur-ws.css?version=2022-01">
<TITLE>CEUR-WS.org - CEUR Workshop Proceedings (free, open-access publishing, computer science/information systems)</TITLE>
<LINK REL="SHORTCUT ICON" HREF="ceur-ws.ico">
</HEAD>
<BODY>


<table border=0 cellpadding=0 cellspacing=5 width="97%">
<tr>
<td align=left valign=middle>
<div id="CEURWSLOGO"></div>
<!--<img alt="[25years CEUR-WS]" style="padding:4px; float:left;"  width="550" src="./CEUR-WS-logo-originals/2020/CEUR-WS-25anniversary.png"> -->
</td>
<td align=justify valign=middle>
<font face="ARIAL,HELVETICA,VERDANA" size =-2 color="#363636">
```

Рис.2.2. Програмний код сайту CEUR

Дослідивши програмний код сторінки, було визначено, що список наукових подій знаходиться в таблиці на головній сторінці сайту у відповідному тезі <table>:

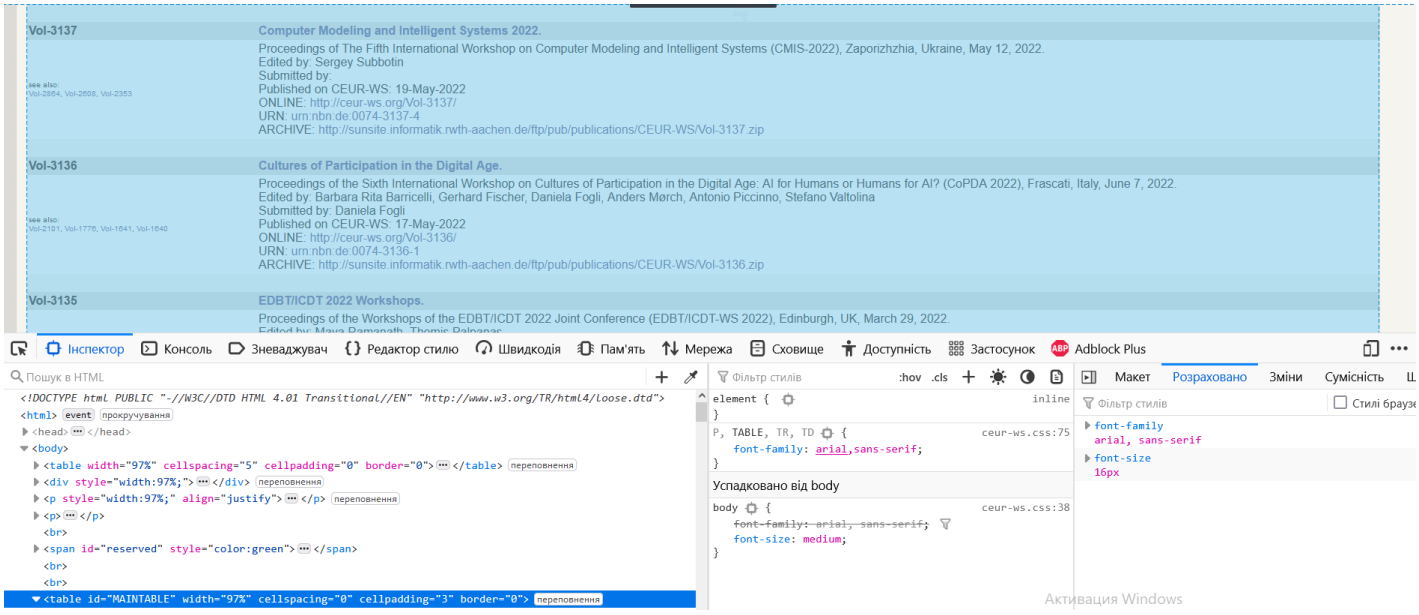


Рис.2.3. Процес дослідження програмного коду CEUR – таблиця наукових подій (виділена блакитним кольором область)

Щоб перейти на сторінку наукової події, потрібно звернутися до тегу посилання <a>.

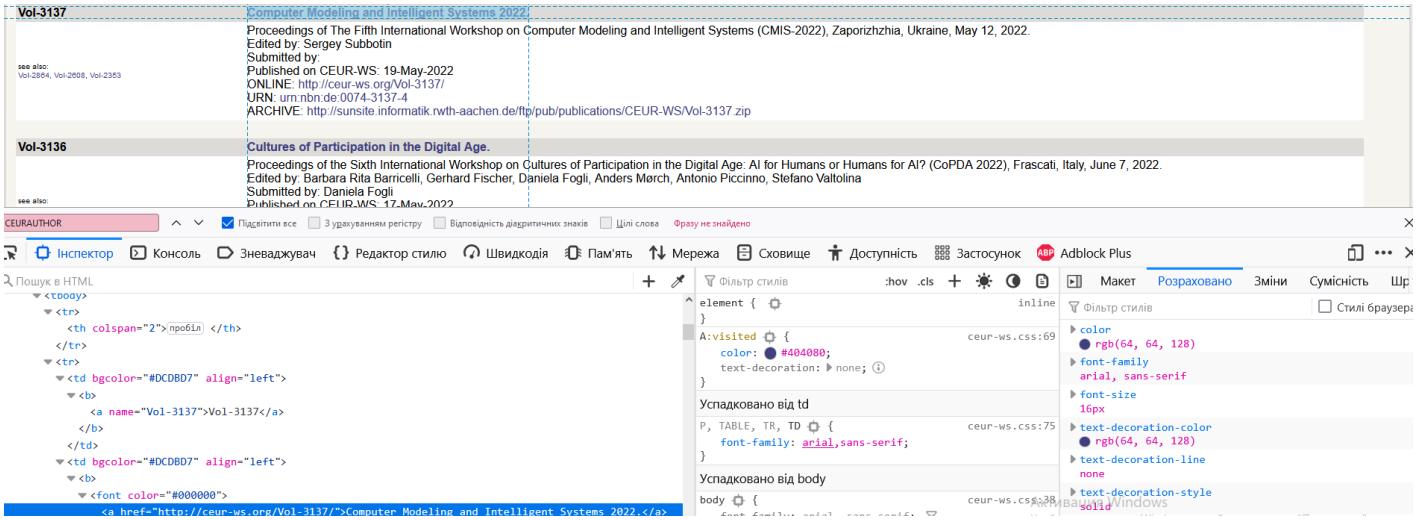


Рис.2.4. Процес дослідження програмного коду CEUR – посилання на окрему сторінку наукової подій (виділена блакитним кольором область.)

Автора статей містяться у тезі класу “ CEURAUTHOR” або класу “CEURAUTHORS”:

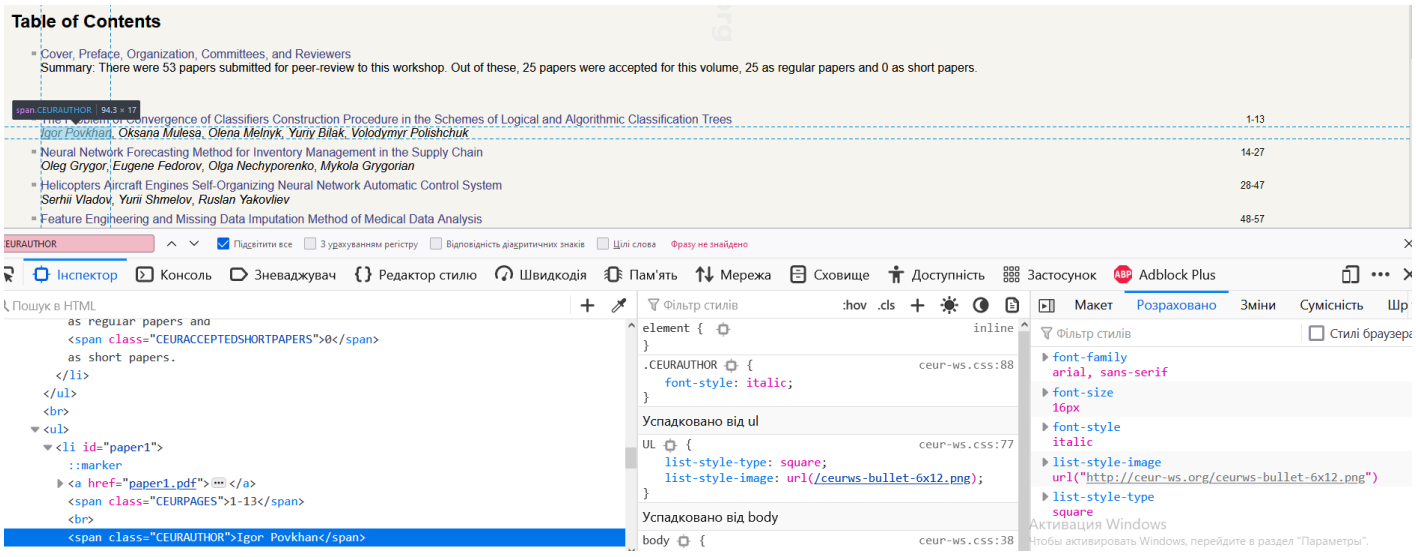


Рис.2.5. Процес дослідження програмного коду CEUR – автори наукових статей (виділена блакитним кольором область.)

Для відслідковування цитувань науковців, потрібно визначити, де міститься посилання на статтю. Було проаналізовано і визначено, що посилання на статті містяться у тезі класу “ CEURVOLTITLE”.

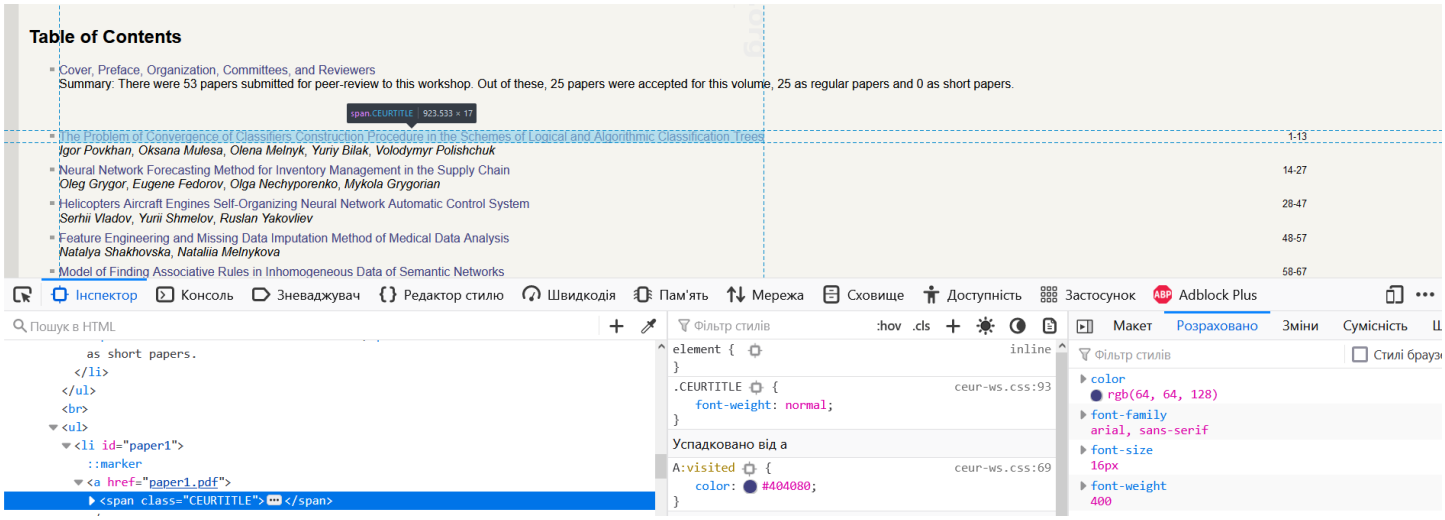


Рис.2.6. Процес дослідження програмного коду CEUR – посилання на наукову статтю (виділена блакитним кольором область.)

2.3. Проектування розробки програми

Для створення функціональної моделі, була використана діаграма IDEF0 [17].

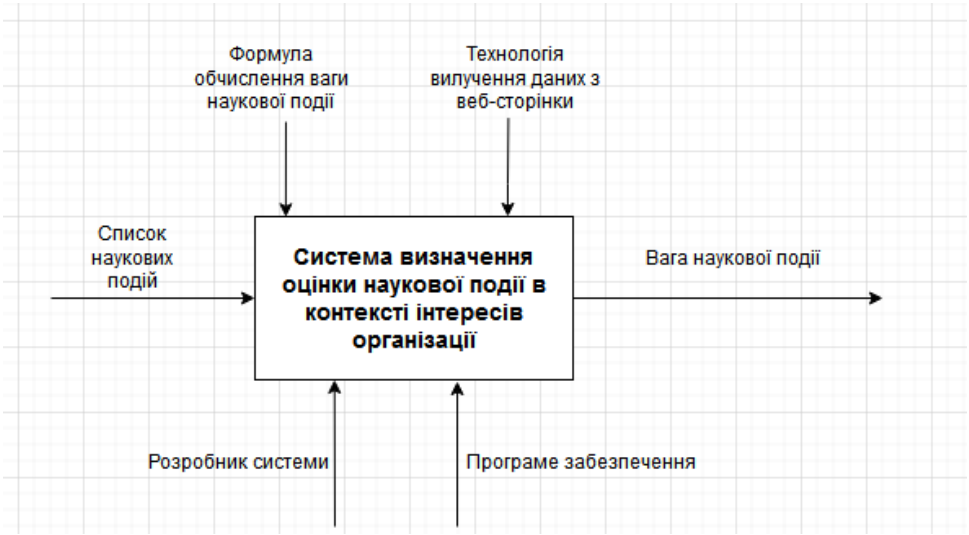


Рис.2.7. IDEF0 діаграма

Діаграма IDEF0 відображає головне завдання – систему визначення оцінки наукової події в контексті інтересів організації. На вході – список наукових подій, на виході – вага наукової події. Механізмами виступає сам розробник системи та програмне забезпечення, яке використовує розробник.

Створення системи відбувається завдяки формулі обчислення ваги наукової події та технологіям вилучення даних з веб-сторінок.

Для розробки алгоритму роботи програми була реалізована деталізована діаграма IDEF0 [17].

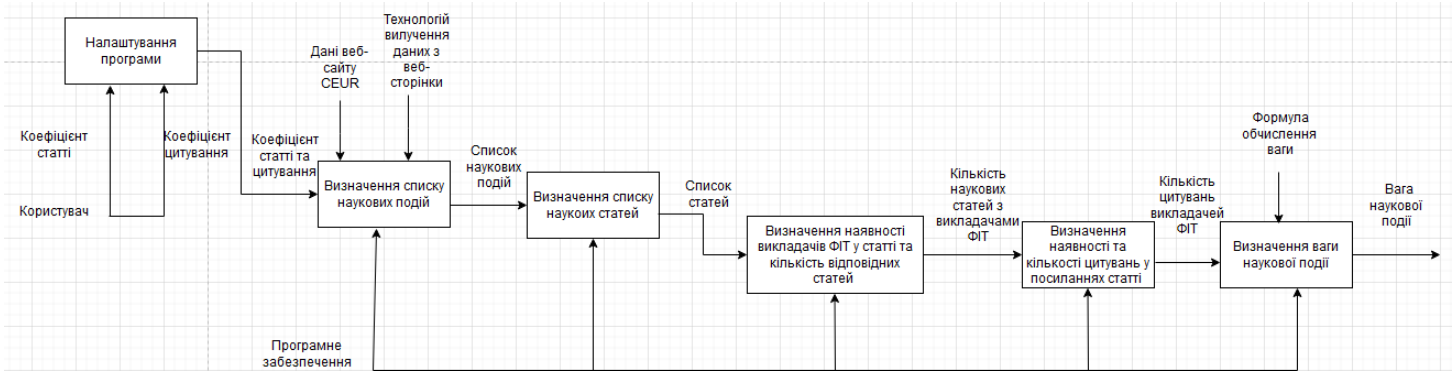


Рис.2.8. Деталізована IDEF0 діаграма

Відповідно даній діаграмі, першим кроком має бути визначення користувачем коефіцієнта наукової статті та цитування. Після цього, має відбуватися визначення списку наукових подій серед всіх даних, які надає веб-сайт CEUR за допомогою певних технологій і знань веб-парсингу, веб-скрепінгу. Далі необхідно визначити наукові статті і їх кількість, серед авторів яких наявні викладачі ФІТ. Якщо такі статті наяні, то ми переходимо до самих статтей і аналізуємо їх на предмет цитувань. Коли всі необхідні дані отримані, обчислюється вага наукової події. Слід зазначити, що всі кроки алгоритму після введення коефіцієнтів наукової статті та ваги, повинно виконувати розроблене програмне забезпечення у автоматизованому режимі.

Виходячи з поставлених задач, описаних у першому розділі даної випускної кваліфікаційної роботи, було прийнято рішення розробити діаграму прецедентів [14, 15, 16]. Діаграма прецедентів – це діаграма, що відображає відношення між акторами та прецедентами у системі. Актори – це хтось або шось, що використовує систему для досягнення мети. Актори позначаються в діаграмі стилізованими людськими фігурами. Актори діляться на основних та

другорядних. Основні актори – ті, хто ініціюють взаємодію з системою, другорядні – більш реакційні. Прецедент – це опис певного аспекта системи. Слід зазначити, що прецедент не показує, “як” досягається певний результат, а лише “що” саме виконується. Прецеденти позначаються на діаграмі у вигляді овалів.

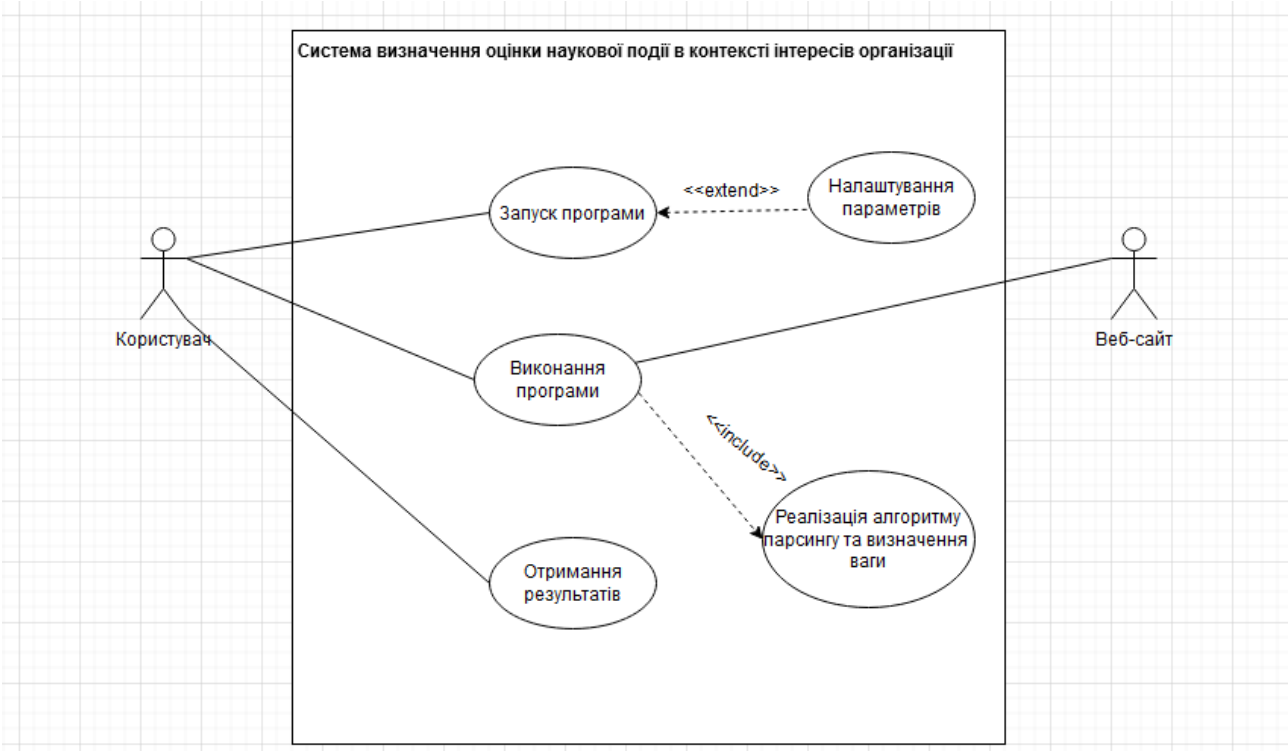


Рис.2.9. Розроблена діаграма прецедентів

Як видно з діаграми, акторами виступають:

1) Користувач системи;
2) Веб-сайт, що надає дані. У даній дипломній роботі таким веб-сайтом виступає CEUR.

У даній діаграмі прецедентів наявні такі відношення між акторами та прецедентами:

1) Асоціації – позначаються звичайною лінією ;
2) включення (include) – позначається пунктирною лінією з відповідною назвою. Включений прецедент є обов’язковим, тобто процес,

пов’язаний з включеним прецедентом, не буде завершеним без виконання включеного прецедента ;

3) розширення (extend) - також позначається пунктирною лінією з відповідною назвою. Розширений прецедент, на відміну від включеного, не є обов’язковим ;

Відповідно до діаграми прецедентів, користувач запускає програму, за бажанням проводить налаштування певних параметрів, починається виконання програми, що включає в себе реалізацію алгоритму парсинга даних та визначення ваги наукової події, за допомогою даних, які надає веб-сайт CEUR. Після завершення програми, користувач отримує відповідні результати.

2.4. Вибір мови програмування та використовуваних бібліотек

Першим кроком для створення системи визначення оцінки наукової події в контексті інтересів організації є визначення мови програмування, адже мова програмування - це основа для всієї системи, оскільки певні нюанси використання мови безпосередньо впливають на продуктивність та можливості продукту розробки.

Основа функціоналу розробленої системи визначення оцінки наукової події в контексті інтересів організації складається з обробки програмного коду сторінки веб-сайту, що складається з тегів. Отже, при виборі мови програмування слід спиратися саме на цей факт. Взагалі, парсинг даних можна здійснювати на багатьох мовах програмування (Python, C++, C#, Ruby). Проаналізувавши переваги і недоліки мов програмування, наведених у переліку, для розробки системи визначення оцінки наукової події в контексті інтересів організації було обрано мову програмування Python через наявність зручних інструментів у вигляді бібліотек для опрацювання HTML розмітки та PDF файлів.

Python – це динамічна інтерпретована мова програмування високого рівня, що фокусується на читанні коду. Python має комплексну і велику стандартну бібліотеку, яка надає автоматичне управління пам'яттю та дозволяє

використовувати динамічні функції. Python використовується в багатьох проектах в якості основної мови програмування, також її використовують для створення певних розширень або додатків.

Мова програмування Python підтримує різні парадигми програмування, в тому числі структурне, об'єктно-орієнтоване, функціональне та імперативне програмування. Python вимагає чіткої організації коду у класи та функції, які в свою чергу, можуть бути згруповані в модулі і надалі в пакети. Саме групування коду в пакети, надає розробникам можливість використання готових рішень у своїх проектах.

Для побудови системи визначення оцінки наукової події в контексті інтересів організації, була обрана мова Python версії 3.10.1. На даний момент, саме ця версія є найбільш стабільною та надає користувачам можливість використання великої кількості бібліотек.

Основні переваги Python:

1) Велика кількість різноманітних бібліотек. Python має як багато стандартних вбудованих бібліотек, так і велику кількість створених іншими користувачами бібліотек, що можуть значно полегшити розробку певного продукту.
2) Простота. У порівнянні з іншими мовами програмування, Python є досить простим і легким у навчанні, тому Python легко читати.
3) Портативність. Python підтримують усі платформи, такі як Windows, Linux, Macintosh, також наявна підтримка Python у ігрових станціях. Слід зазначити, що у Python можливо навіть використання біблотек інших мов програмування, таких як C, C++.
4) Зручна робота зі штучним інтелектом. Python є найпопулярнішою мовою програмування для створення моделей машинного навчання, побудови нейронних мереж. Причиню цього є наявність

зручних бібліотек, що дають можливість побудувати нейронну мережу, наприклад, у декілька рядків.

5) Можливість обробки великої кількості даних завдяки паралельним обчисленням.

Основні недоліки Python:

1) Динамічна типізація, внаслідок якої споживається більше ресурсів.
2) Продуктивність. Python не є найбільш “спритною” мовою програмування через свою інтерпретованість.
3) Неможливість модифікації вбудованих класів.

Для реалізації програмного коду Python було обрано середовище Jupyter Notebook. Jupyter Notebook - це середа розробки, де одразу можна побачити результат виконаного коду, що є досить зручно. Jupyter Notebook підтримує такі мови програмування:

6) Ruby;
7) Perl;
8) R;
9) bash-скрипти

Jupyter Notebook можна запускати як на комп’ютері безпосередньо, так і у хмарному середовищі. Серед переваг роботи програми в хмарі є те, що хмарна система виділить ресурси, що є плюсом, якщо поставлена задача, пов’язана з обробкою великої кількості даних. Також відсутня необхідність щось встановлювати на комп’ютер, сервіс зробить все сам. Серед мінусів можна виділити можливу низьку швидкість роботи, в порівнянні з комп’ютером.

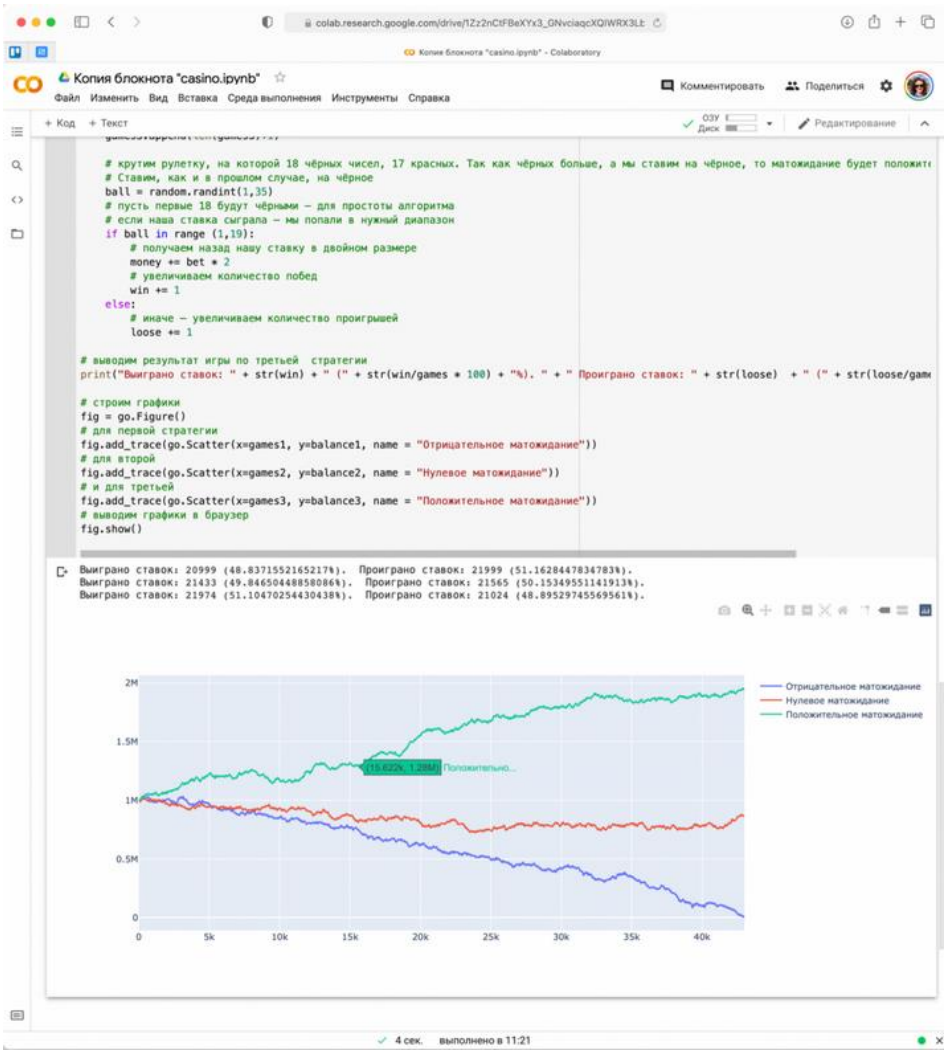


Рис.2.10. Приклад програмного коду Jupyter Notebook у хмарному середовищі

Якщо розробник має бажання контролювати все самостійно, то слід інсталювати Jupyter Notebook безпосередньо на комп’ютер.

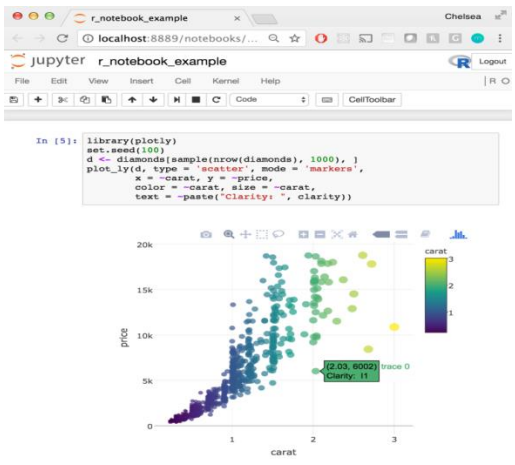


Рис.2.11. Приклад програмного коду Jupyter Notebook на комп’ютері

Перед тим як витягати відповідні дані з веб-сторінки, до сторінки необхідно провести запит. Для цього використовується бібліотека requests[18].

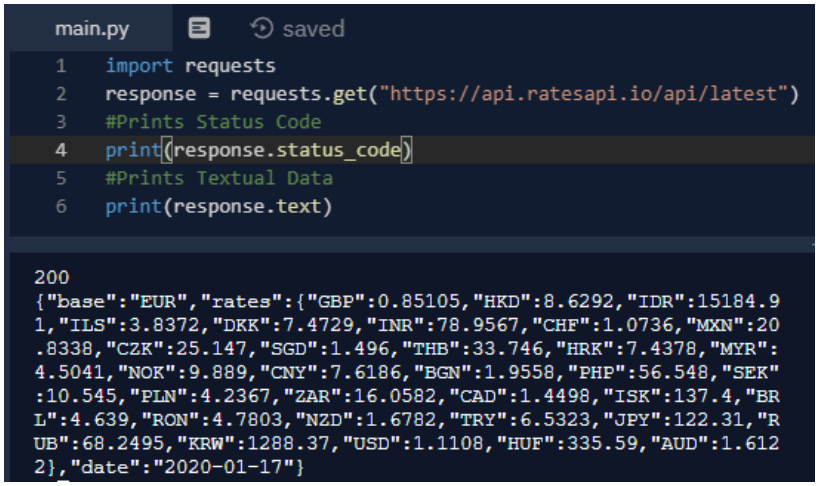


Рис.2.12. Приклад використання бібліотеки requests у python

Як вже було сказано, для витягу інформації можуть використовуватися спеціалізовані бібліотеки або регулярні вирази[18]. Регулярний вираз - це рядок, що описує або збігається з множиною рядків, відповідно до набору спеціальних синтаксичних правил. Регулярні вирази використовуються для пошуку та модифікування тексту на основі певних шаблонів. У мові програмування Python для роботи з регулярними виразами використовується бібліотека re. Модуль re є встроєним у Python, отже відсутня потреба додаткового встановлення даної бібліотеки. Для використання даної бібліотеки лише слід імпортувати її за допомогою команди “import re”.

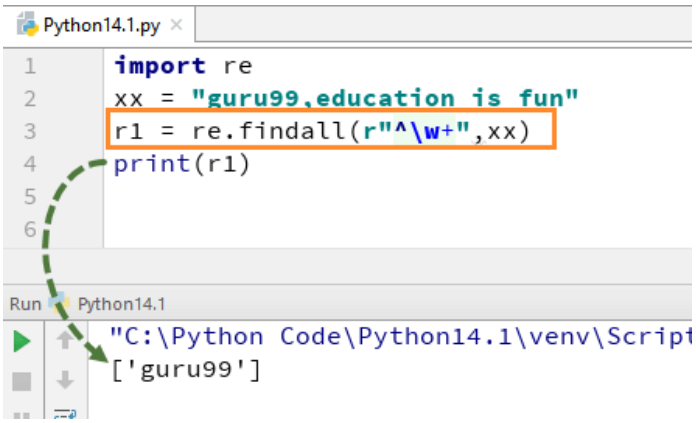


Рис.2.13. Приклад використання регулярних виразів на мові програмування Python

Для безпосереднього вилучення даних з веб-сторінок була використана розповсюджена для цих задача - бібліотека BeautifulSoup [19, 20, 21]. Вона дозволяє вилучати дані з файлів формату HTML та XML.

До найпоширеніших і основних методів бібліотеки BeautifulSoup слід віднести методи find () і find_all (). Вони дозволяють вилучати інформацію з відповідних тегів. Різниця між двома методами в тому, що метод find () вилучає лише перший елемент, в той час як find_all () вилучає всі елементи під відповідним тегом.

```
soup = BeautifulSoup(page.content, 'html.parser')
soup.find_all('p')
```

```
[<p>Here is some simple content for this page.</p>]
```

Рис.2.14. Приклад використання методу find_all ()

Для обробки PDF-файлів використовувалася бібліотека PyPDF2 [22]. Вона дозволяє розділяти, об’єднувати, обрізати та перетворювати сторінки PDF-файлів.

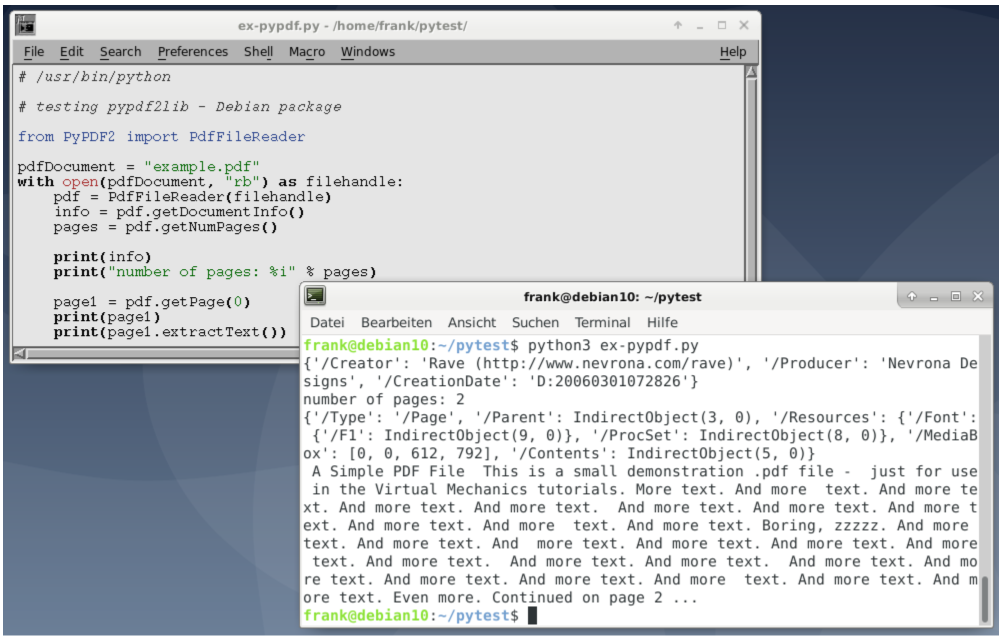


Рис.2.15. Приклад використання методу бібліотеки PyPDF2

Для створення графічного інтерфейсу програми була використана бібліотека Tkinter [23]. Вона входить до стандартної бібліотеки Python. Основний плюсом Tkinter є те, що він є відносно легким у використанні, порівняно з іншими бібліотеками.

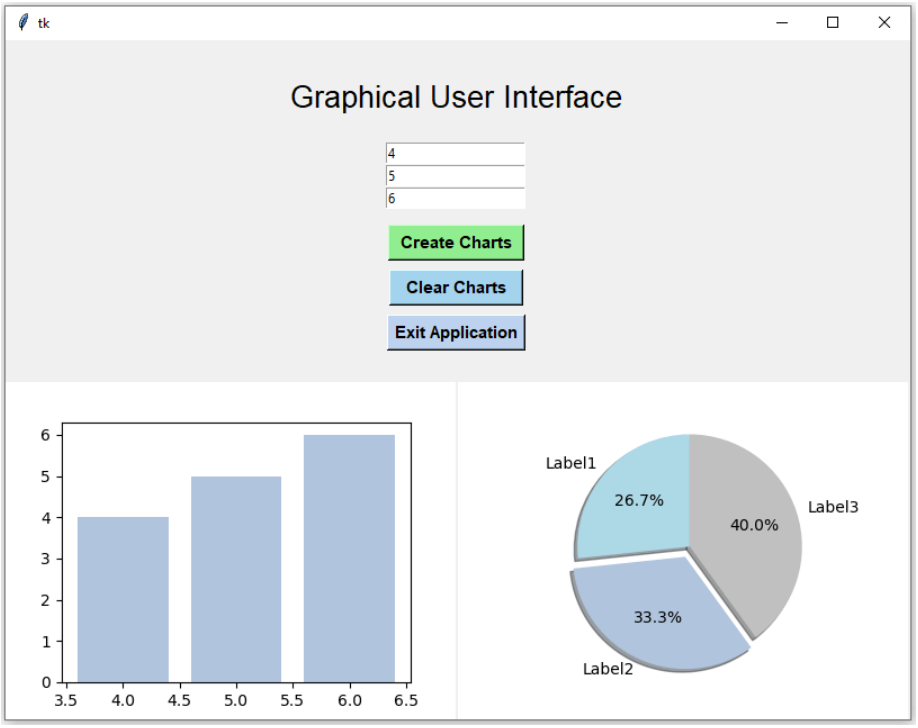


Рис.2.16. Приклад створеного графічного інтерфейсу за допомогою використання бібліотеки Tkinter

РОЗДІЛ 3. ОПИС ТА ТЕСТУВАННЯ СИСТЕМИ ВИЗНАЧЕННЯ ОЦІНКИ НАУКОВОЇ ПОДІЇ В КОНТЕКСТІ ІНТЕРЕСІВ ОРГАНІЗАЦІЇ

3.1. Опис розробленої системи

Після запуску програми, користувача зустрічає 2 вікна:

- вікно допомоги:

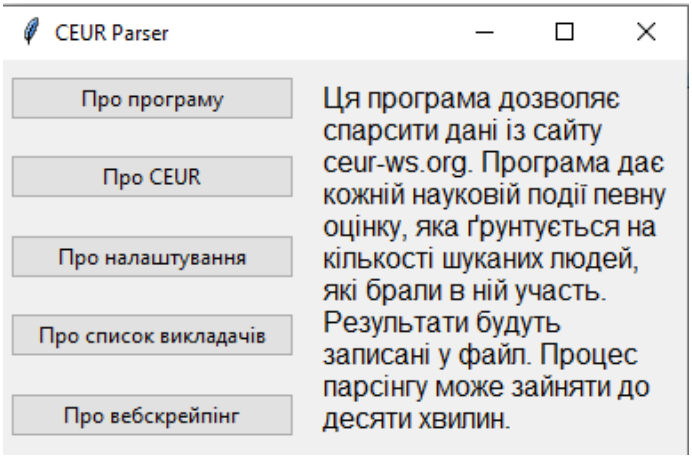


Рис.3.1. Вікно допомоги

- основне вікно програми, куди буде виводитися інформація про події:

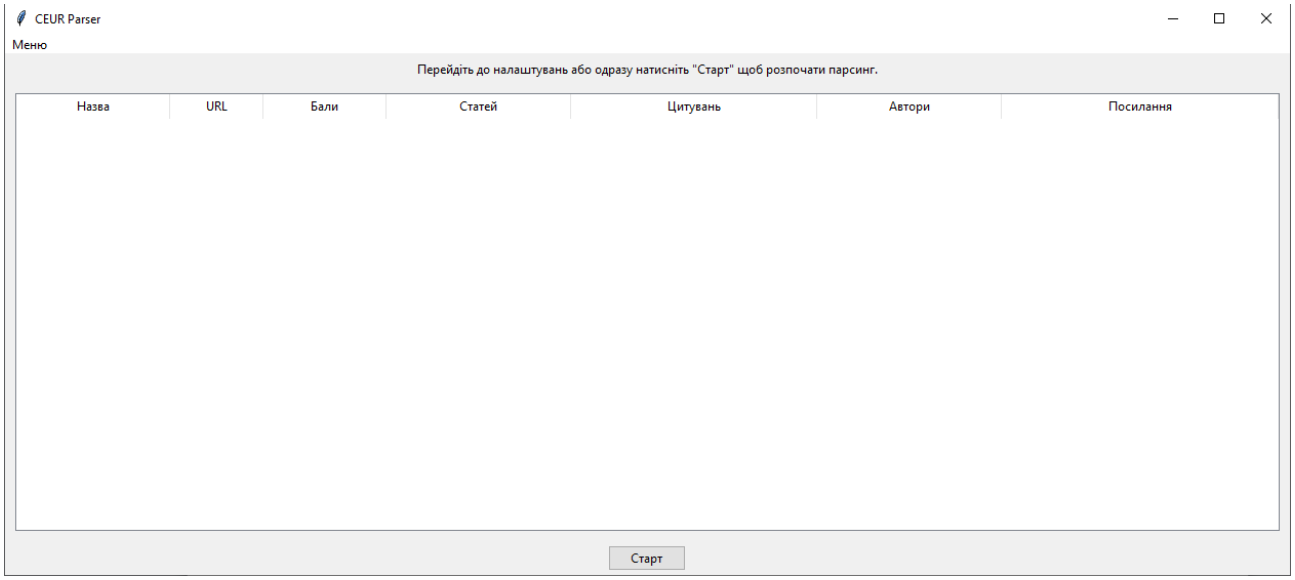


Рис.3.2. Основне вікно програми

Про вікно допомоги. Якщо клацнути зліва на кнопку “Про програму”, тоді користувач отримає коротку інформацію про те, як працює сама програма:

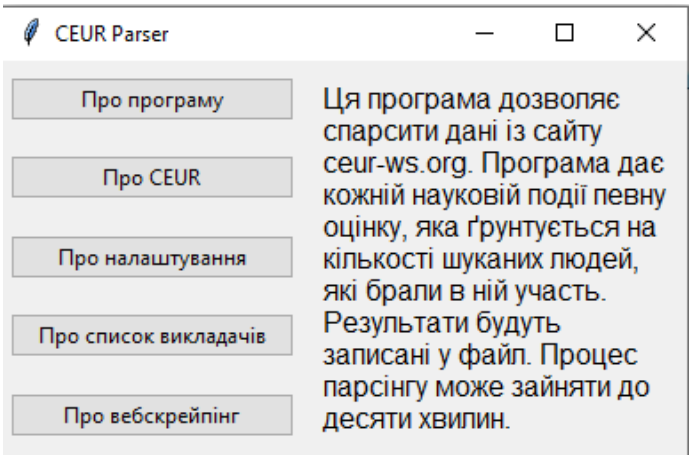


Рис.3.3. Кнопка “Про програму” у вікні допомоги

Якщо клацнути зліва на кнопку “Про CEUR”, тоді користувач отримає коротку інформацію про те, що таке CEUR:

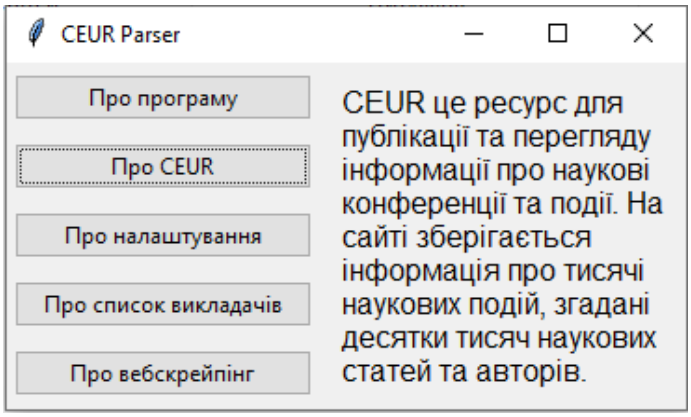


Рис.3.4. Кнопка “Про CEUR” у вікні допомоги

Якщо клацнути зліва на кнопку “Про налаштування”, тоді користувач отримає коротку інформацію про вкладку “Налаштування” :

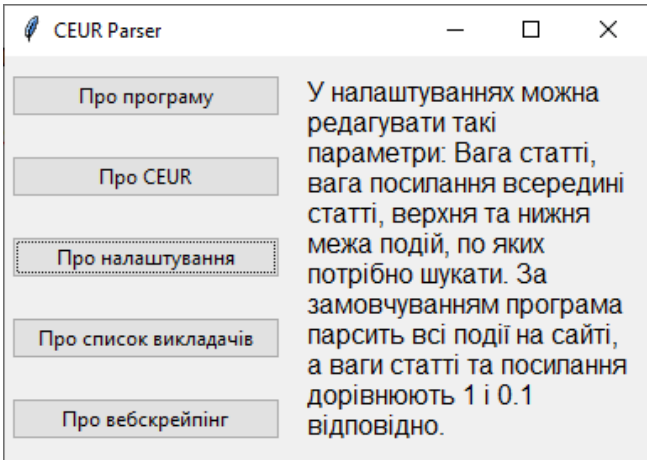


Рис.3.5. Кнопка “Про налаштування” у вікні допомоги

Якщо клацнути зліва на кнопку “Про список викладачів”, тоді користувач отримає коротку інформацію про вкладку “ Про список викладачів ” :

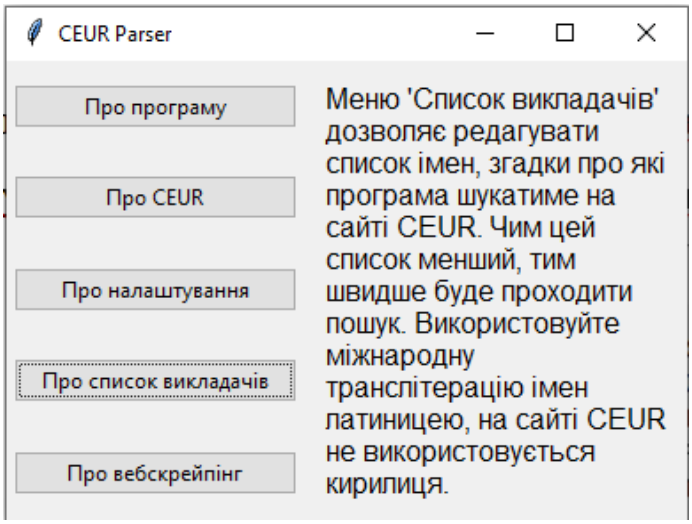


Рис.3.6. Кнопка “Про список викладачів” у вікні допомоги

Остання кнопка – кнопка “Про вебскрейпінг” дає стислу інформацію про те, що таке вебскрейпінг:

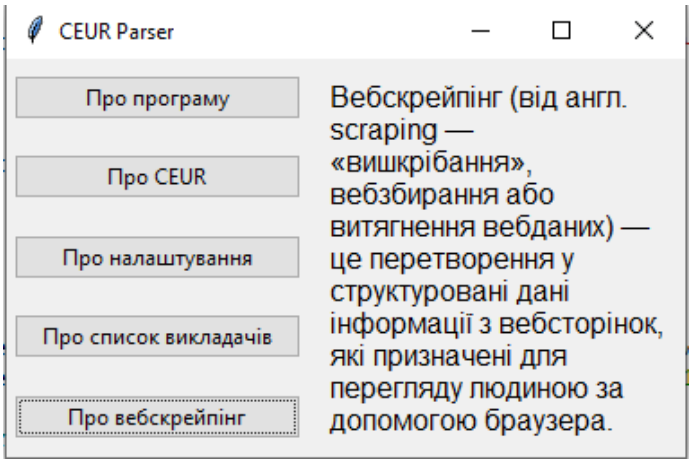


Рис.3.7. Кнопка “Про вебскрейпінг” у вікні допомоги

Опис меню програми. Воно знаходиться у верхньому лівому куту програми:

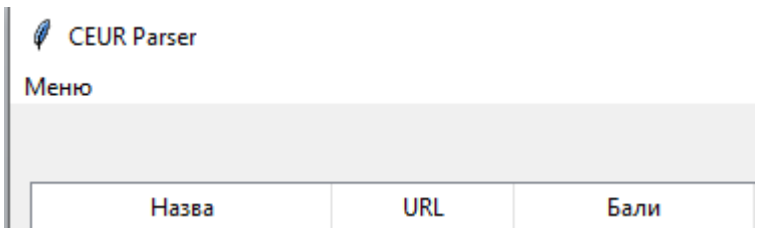


Рис.3.8. Розташування меню програми

Клацнувши на нього лівою кнопкою миші, можна побачити 3 основні пункти: “Допомога” (була описана попередньо), “Налаштування”, “Список викладачів”.

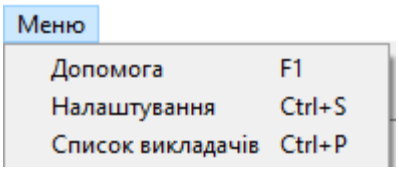


Рис.3.9. Основні розділи меню програми

Розбіл розділу “Налаштування”.

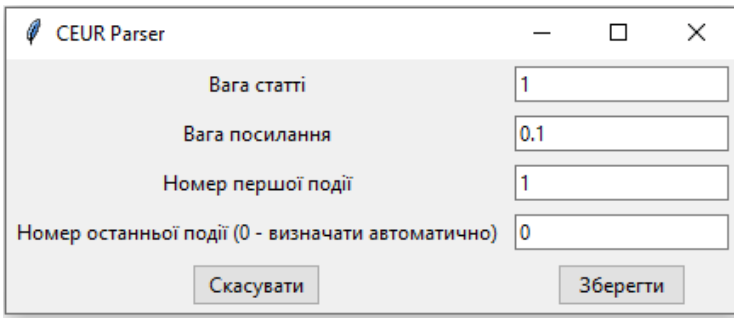


Рис.3.10. Вигляд розділу “Налаштування”

У даному розділі користувач має можливість вводити вагу статті, посилання та вказувати межі парсингу подій, визначаючи номер першої події для парсингу та номер останньої події. Чим більше подій для парсингу потрібно обробити – тим довший час виконання програми. Якщо вказати номер останньої події 0, то програма сама визначить номер останньої події на сайті. Це дуже зрочно, у випадку якщо користувач не знає скільки всього подій знаходиться на CEUR.

Розбір розділу “Список викладачів”.

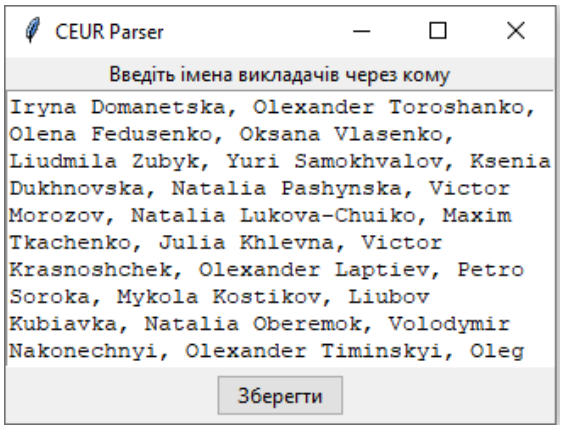


Рис.3.11. Вигляд розділу “ Список викладачів ”

У даному розділі можна побачити початковий список викладачів ФІТ. Користувач має можливість додавати користувачів через кому і програма буде шукати події, спираючись на оновлений список.

Розбір основного вікна програми.

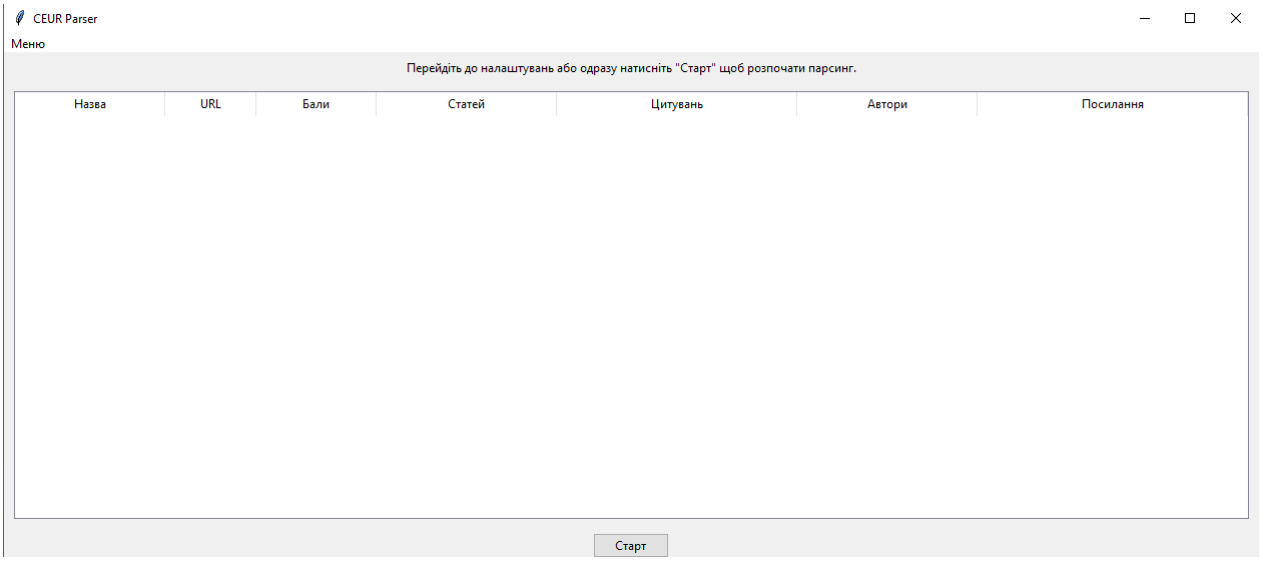


Рис.3.12. Основне вікно програми

Як видно, основне поле складається з таблиці (на скріншоті вона порожня). Дана таблиця складається з таких полів:

- Назва – відображає назву наукової події;
- URL – посилання на відповідну наукову подію;
- Бали – вага наукової події;

- Статей – кількість статей та скільки балів додається до ваги наукової події;
- Цитувань – кількість цитувань та скільки балів додається до ваги наукової події;
- Автори – автори статей;
- Посилання – цитування викладачів з ФІТ у статтях наукової події.

Знизу наявна кнопка “Старт”, клацнувши на яку починається основна робота програми.

3.2. Тестування системи

Спочатку розглянемо предметно роботу системи на прикладі однієї наукової події. Було обрано розгляду наукову подію “Information Technology and Implementation 2021.”:


Vol-3132 Information Technology and Implementation 2021.
Selected Papers of the VIII International Scientific Conference “Information Technology and Implementation” (IT&I-2021). Conference Proceedings, Kyiv, Ukraine, December 01-03, 2021.
Edited by: Anatoly Anisimov, Vitaliy Snytyuk, Aldrich Chris, Andreas Pester, Frederic Mallet, Hiroshi Tanaka, Iurii Krak, Karsten Henke, Mykola Nikitchenko, Oleg Chertov, Oleksandr Marchenko, Sándor Bozóki, Vitaliy Tsyganok, Vladimir Vovk
Submitted by: Vitaliy Snytyuk
Published on CEUR-WS: 2-May-2022
ONLINE: http://ceur-ws.org/Vol-3132/
URN: urn:nbn:de:0074-3132-6
ARCHIVE: http://sunsite.informatik.rwth-aachen.de/ftp/pub/publications/CEUR-WS/Vol-3132.zip
see also: Vol-2845,Vol-2833

Рис.3.13. Обрана наукова подія на сайті CEUR

Перелік статей, де серед авторів наявні викладачі ФІТ:


- Model for Determining the Protection Level of a Complex System
  Tetiana Babenko, Hryhorii Hnatiienko, Andrii Bigdan
- Comparative Evaluation of a Universities’ Websites Quality
  Yuri Kravchenko, Olga Leshchenko, Nataliia Dakhno, Maksym Radko
- Web Application for an Information System for Diagnosing the Quality of Electricity Consumers Using Cloud Technologies
  Nikolay Kiktev, Alexey Kutyrev, Dmitriy Khort, Oleksii Kalivoshko
- Student – Training Environment Interaction: Soft Skills Development within E-learning
  Tetyana Sergeyeva, Dorin Festeu, Sergiy Bronin, Natalya Turlakova
- Alternative Method of Cryptocurrency Wallets Managing
  Gabit Omarov, Dzholdas Dzhuruntayev, Andriy Fesenko, Sanzhar Umbet, Serhii Dorozhynskyi

Рис.3.14. Перелік статей події “Information Technology and Implementation 2021.”, де в авторах наявні викладачі ФІТ

Всього наявно 5 статей. Визначимо кількість цитувань викладачів ФІТ у даних статтях. Переглянемо список цитувань статті “Model for Determining the Protection Level of a Complex System”:

[16] Palko, D., Hnatienko, H., Babenko, T., Bigdan, A. Determining key risks for modern distributed information systems / CEUR Workshop Proceedings, 2021, 3018, pp. 81–100.

[17] Palko, D., Myrutenko, L., Babenko, T., Bigdan, A. Model of Information Security Critical Incident Risk Assessment / 2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020, 2021, pp. 157–161, 9468107.

[18] Kravchenko, Y., Vialkova, V. The problem of providing functional stability properties of information security systems // Modern Problems of Radio Engineering, Telecommunications and Computer Science, Proceedings of the 13th International Conference on TCSET 2016, pp. 526–530.

[19] Hrechko Viktoriia; Hrygorii Hnatienko; Tetiana Babenko. An intelligent model to assess information systems security level // 2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), London, United Kingdom, 29-30 July 2021/ Date Added

[23] Hnatiienko, H., Snytyuk, V. A posteriori determination of expert competence under uncertainty / CEUR Workshop Proceedings, 2019, 2577, pp. 82–99.

[25] Babenko, T., Hnatiienko, H., Vialkova, V. Modeling of the integrated quality assessment system of the information security management system / CEUR Workshop Proceedings, 2021, 2845, pp. 75–84.

[27] Bozóki Sándor & Tsyganok Vitaliy The (logarithmic) least squares optimality of the arithmetic (geometric) mean of weight vectors calculated from all spanning trees for incomplete additive (multiplicative) pairwise comparison matrices International Journal of General Systems. 2019.

[28] Kraevsky, V., Kostenko, O., Kalivoshko, O., Kiktev, N., Lyutyy, I. 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC

Рис.3.15. Перелік цитувань у статті події “Model for Determining the Protection Level of a Complex System”:.”, де в авторах наявні викладачі ФІТ

Як видно, у даній статті наявно понад десяти цитувань викладачів ФІТ. Внаслідок обрахування всіх цитувань, було встановлено, що їх кількість дорівнює 32.

Формула обрахунку ваги наукової події, що була описана у 1-ому розділі:

$$I = q_w * w + q_v * v$$

Отже, виходячи з формули маємо:

к-ть наукових статей - $q_w = 5$;
к-ть цитувань – $q_v = 32$.

Ваги однієї наукової статті (w) і цитування (v) будуть дорівнювати 1 і 0.1, відповідно. Тоді, вага наукової події “Information Technology and Implementation 2021.” дорівнює:

$$I = 5 * 1 + 32 * 0.1 = 8.2$$

Перевіримо даний результат у програмі.

Налаштування ваг посилання і цитувань у програмі:

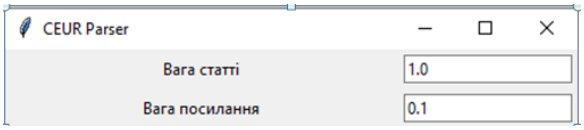


Рис.3.16. Налаштування ваг посилання і цитувань у програмі

Результат програми:

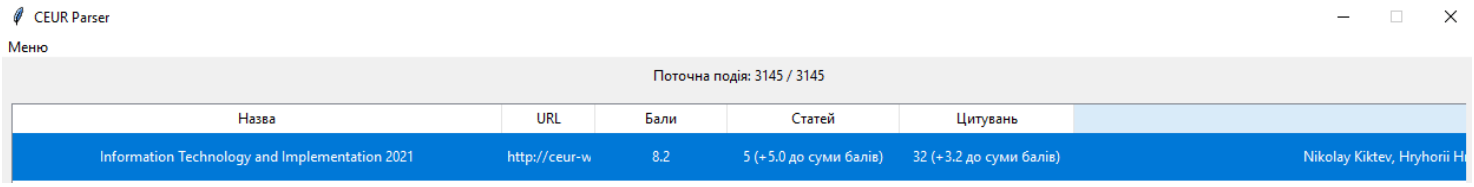


Рис.3.17. Результат програми для наукової події “Information Technology and Implementation 2021.”

Як видно, показник ваги у програмі співпадає з тим, що був обрахован вручну. Далі будуть описані тестування, що показують роботу програми на певній множині наукових подій, без розбіру конкретних випадків.

1) Для першого тестування – налаштування за замовчуванням.

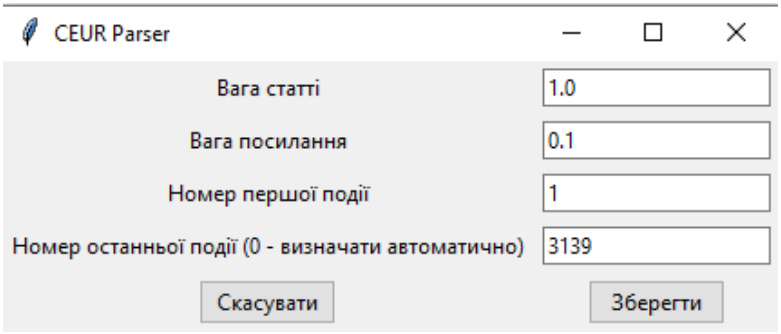


Рис.3.18. Налаштування програми для першого тесту (за замовчуванням)

Результати програми:

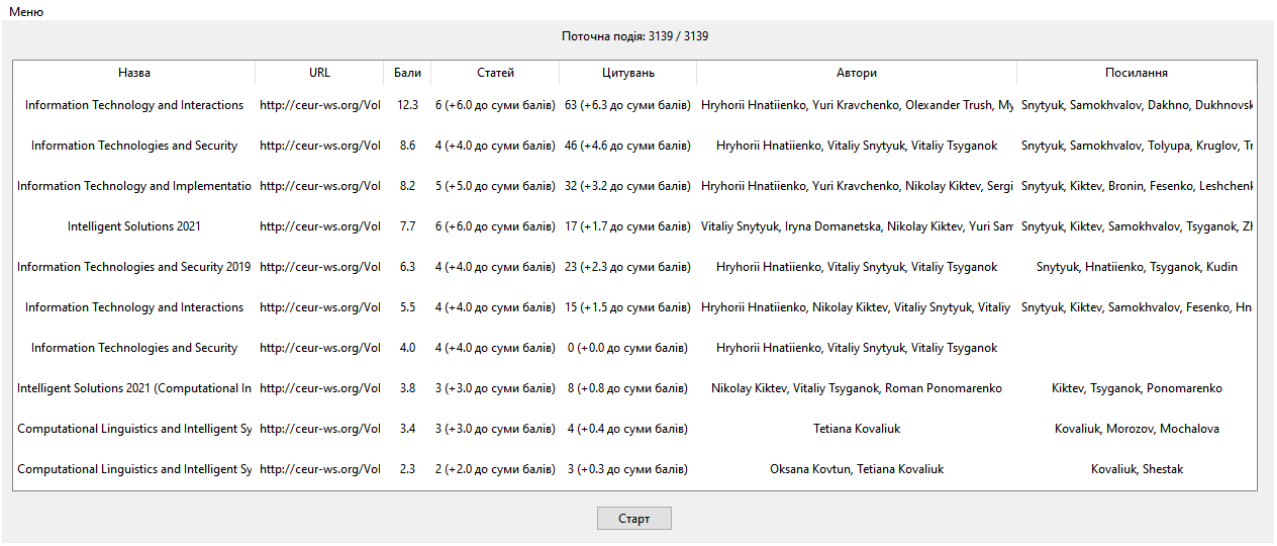


Рис.3.19. Результати роботи програми після першого тесту

Як видно, програма спрацювала коректно. Події сортуються автоматично від найбільшої ваги до найменшої. Слід зазначити, що процес парсингу займав приблизно 10 хвилин. Після завершення процесу парсингу, створюється CSV файл, що зберігає результуючу інформацію:

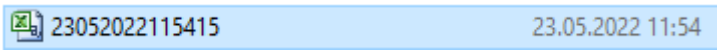


Рис.3.20. Створений CSV файл

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Sergiy Paliy	Mykola Kostikov	Denys Berestov	Natalia Tmienova	Olexander Laptiev	Petro Soroka	Bogdan Yeremer	Volodymir Kudin	Tetiana Kovaliuk	Svetlana Poperes	Anna Kolomiets	Snizhana Gamot	Victor Morozov	Iryna Domanetsk	Anastasia I
2	Article weight = 1.0	Reference weight = 0.1	Lower limit = 3000	Upper limit = 3142											
3	Title	URL	Score	Articles count	References count	Authors	References								
4	Information Technology and Implementation 2021	http://ceur-ws.org/Vol-3132/	8.2	5 (+5.0 to final score)	32 (+3.2 to final score)	Andriy Fesenko, Nikolay	Kiktev, Babenko, Bronin, Snytyuk, Myrutenko, Fesenko, Tsyganok, Trush, Kravchenko, Hnatiienko, Dudnik, Leshchenko, Bigdan								
5	Intelligent Solutions 2021	http://ceur-ws.org/Vol-3018/	7.7	6 (+6.0 to final score)	17 (+1.7 to final score)	Nikolay Kiktev, Iryna Do	Samokhvalov, Kiktev, Domanetska, Babenko, Yeremenko, Snytyuk, Zhabska, Merkulova, Myrutenko, Tmienova, Tsyganok, Bychkov, Krasovska, Bigdar								
6	Intelligent Solutions 2021 (Computational Intelligence & Decision Making Theory)	http://ceur-ws.org/Vol-3106/	3.8	3 (+3.0 to final score)	8 (+0.8 to final score)	Roman Ponomarenko, N	Ponomarenko, Tsyganok, Kiktev								
7	ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer 2021	http://ceur-ws.org/Vol-3013/	1.1	1 (+1.0 to final score)	1 (+0.1 to final score)	Tetiana Kovaliuk	Kovaliuk								
8	Intellectual Systems and Information Technologies 2021	http://ceur-ws.org/Vol-3126/	1.0	1 (+1.0 to final score)	0 (+0.0 to final score)	Nikolay Kiktev									

Рис.3.21. Вигляд створеного CSV файлу

Назва CSV файлу відображає повну дату створення та час створення, що є досить зручно, оскільки за назвою можна відслідковувати та знаходити файл зі збереженою інформацію за конкретною датою. Сам CSV файл коректно відображає всю результуючу інформацію. На першому рядку відображається список усіх викладачів, що знаходяться у відповідному розділі. На другому

рядку знаходяться відповідні параметри з налаштувань програми: вага статті, вага посилання, номер першої події та номер останньої події.

2) Було проведено зміну налаштувань так, щоб парсинг даних відбувався, починаючи з 3000-ої наукової події, та було проведено зміну ваги статей і цитувань на 0.7 і 0.5 відповідно:

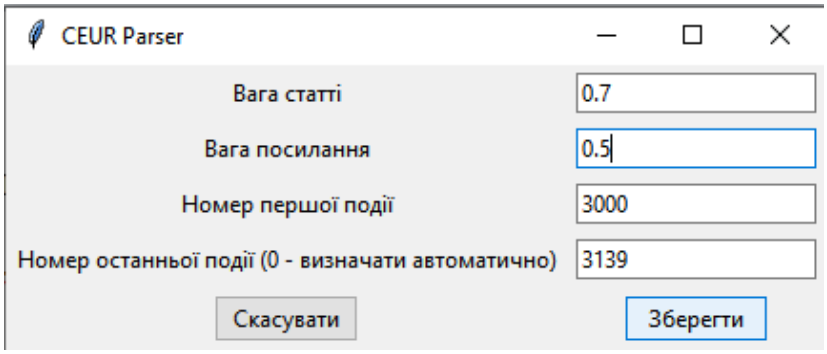


Рис.3.22. Налаштування програми для другого тесту

Результати програми:

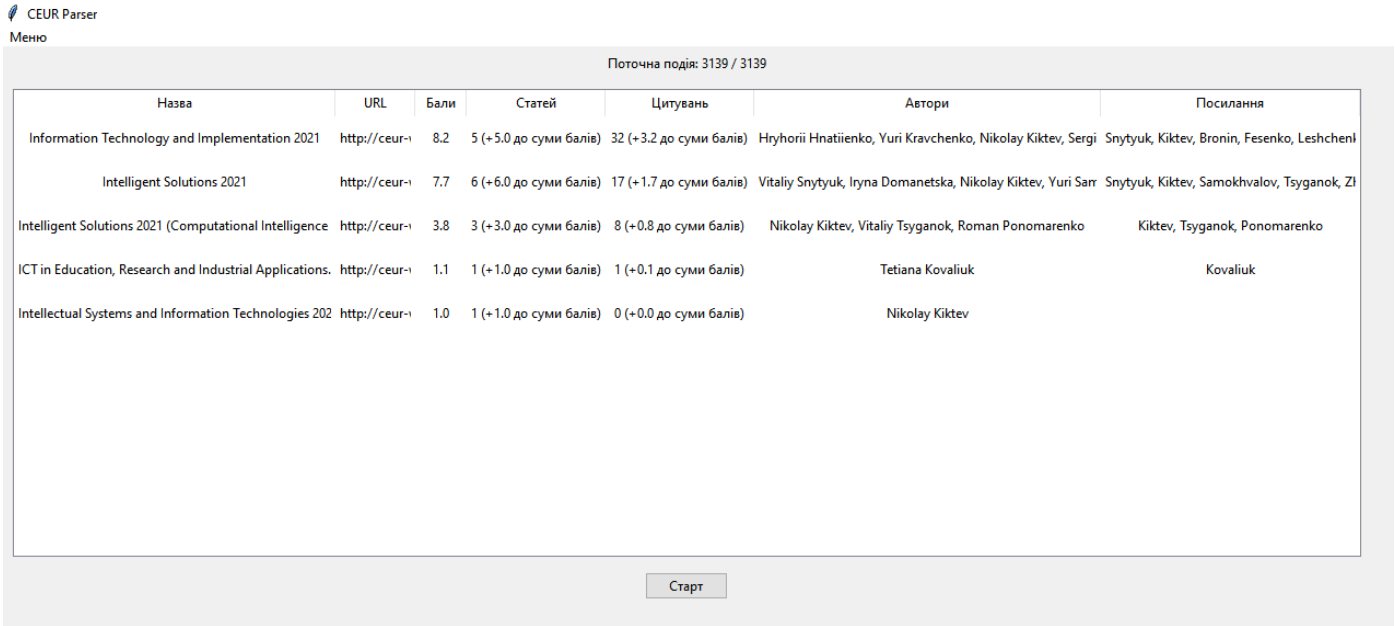


Рис.3.21. Результати роботи програми після другого тесту

Як видно, програма спрацювала коректно, кількість подій значно зменшилася, що є правильно, відповідно до налаштувань програми. Також змінилися ваги, в порівнянні з першим тестом, що також є правильним, відповідно до налаштувань. Час роботи програми також значно скоротився.

3) Було проведено ще один тест, додавши до списка викладачів нового викладача не з ФІТ, що наявний на останній події - Mathieu Lega для перевірки коректності функціоналу додавання викладача до списку.

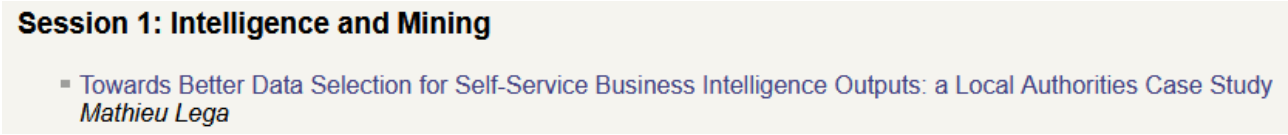


Рис.3.22. Наявність Mathieu Lega в авторстві на останній події

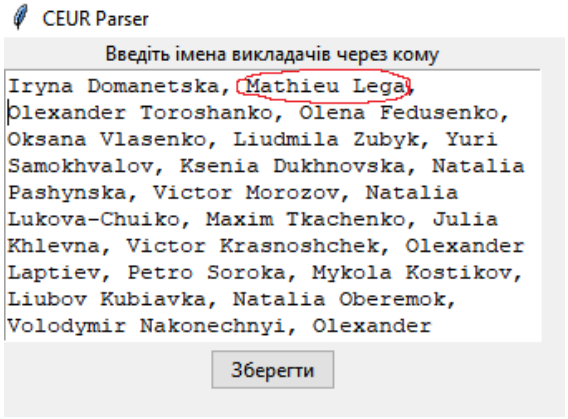


Рис.3.23. Наявність Mathieu Lega в новому списку викладачів

Для економії часу в налаштуваннях було визначено обробляти лише перші 10 подій, ваги залишилися аналогічні 2-ому тесту:

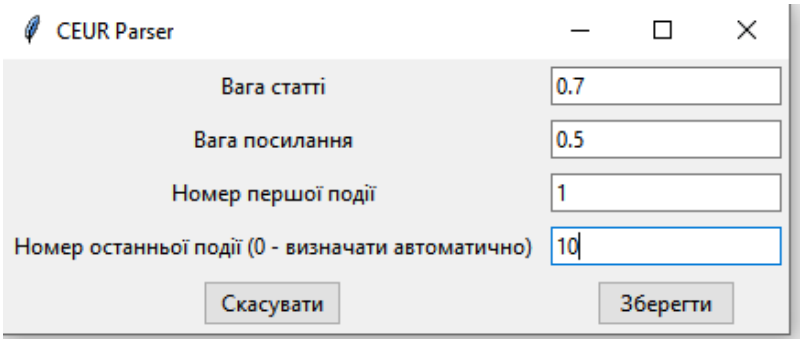


Рис.3.24. Налаштування програми для третього тесту

Результат програми:

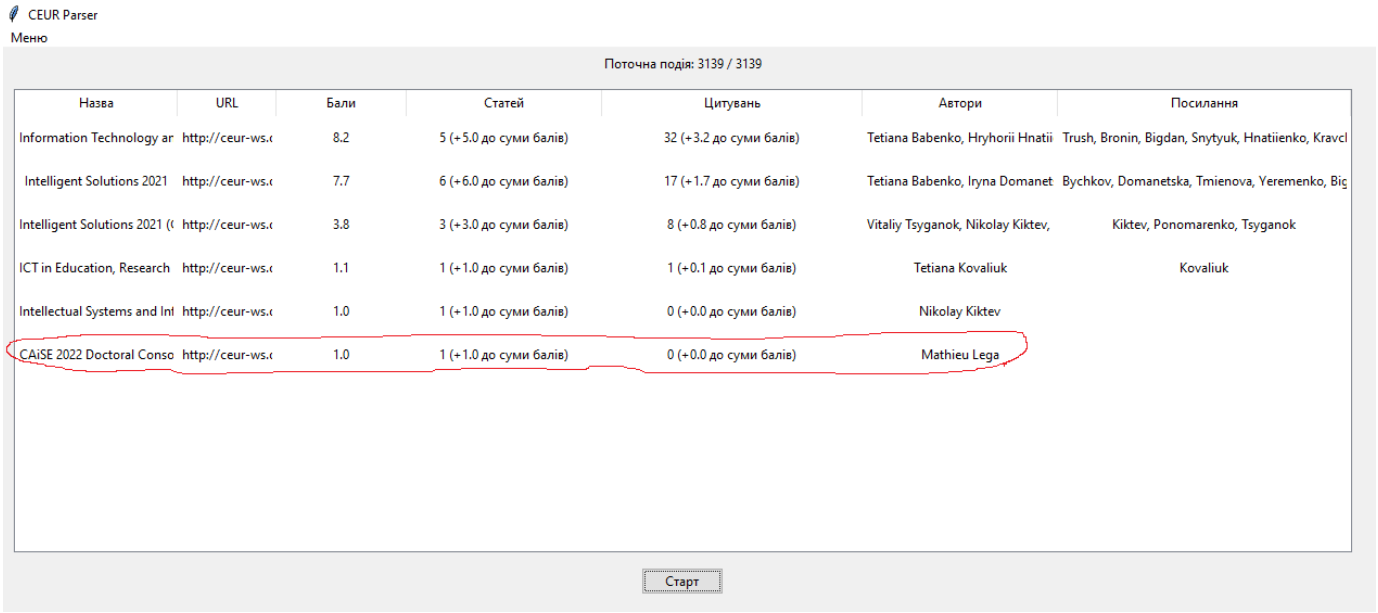


Рис.3.25. Результати програми для третього тесту

Як видно, програма після внесених змін спрацювала коректно і вивела подію, участь в якій приймав доданий до списку викладач.

4) Спроба ввести некоректні значення у номер останньої події:

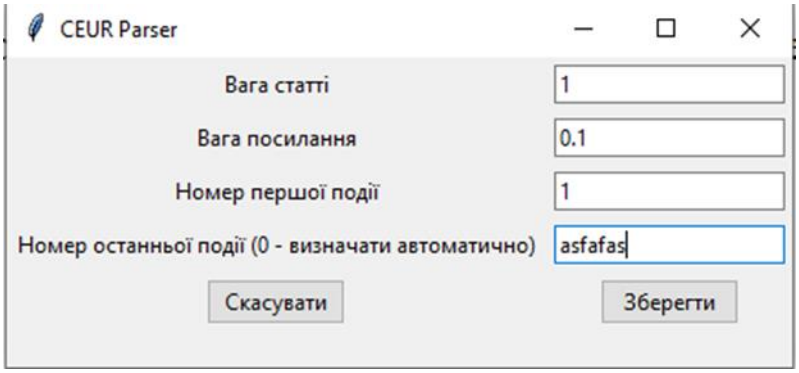


Рис.3.26. Введення некоректного значення у номер останньої події

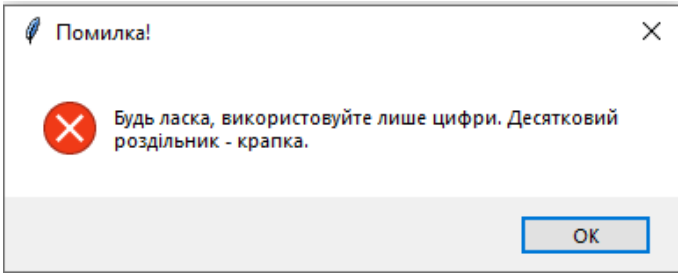


Рис.3.27. Помилка при введенні некоректного значення у номер останньої події

5) Спроба ввести некоректні значення у номер першої події:

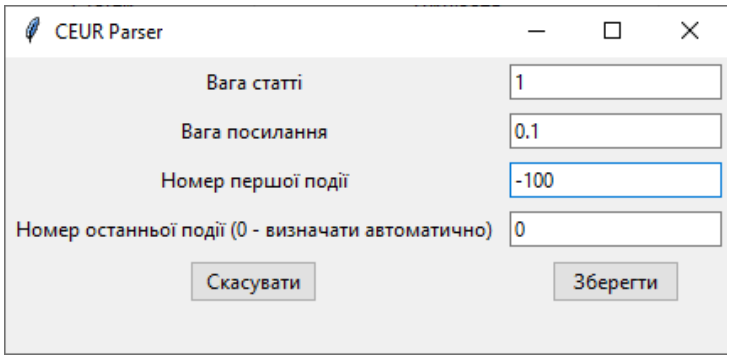


Рис.3.28. Введення некоректного значення у номер першої події

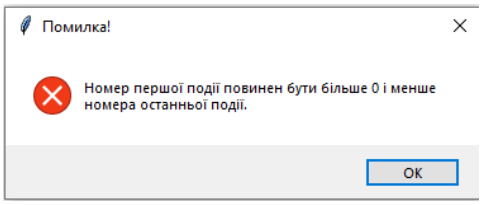


Рис.3.29. Помилка при введенні некоректного значення у номер першої події

Як видно з останніх двох тестів, дана програма передбачає введення некоректних значень при визначенні номера першої та/або останньої події.

ВИСНОВКИ

У результаті виконання дипломного проекту було розроблено систему визначення оцінки наукової події в контексті інтересів організації. Створена система дозволяє відслідковувати події, участь в яких приймали викладачі факультету та обраховувати вагу наукової події.

Під час роботи над дипломним проектом було проведено аналітичний огляд наукометричих баз та метрик, за якими проводиться оцінка наукових матеріалів, сформована математична модель для обрахунку ваги наукової події. У ході проектування розробки програми були створені діаграма різних типів. У якості інструментів розробки програми було обрано мова програмування Python, середовище програмування Jupyter Notebook та бібліотеки BeautifulSoup, PyPDF2, Tkinter.

Для взаємодії користувачів із системою визначення оцінки наукової події в контексті інтересів організації було створено гнучкий інтерфейс, який коректно та зручно виводить результати роботи програми.

У ході тестування роботи було встановлено, що програма працює коректно після різних налаштуваннях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1) Гнатієнко Г.М., Снитюк В.Є., Тмєнова Н.П. Визначення інтегральної якості наукової події у контексті інтересів організації // Інформаційні технології та безпека. Матеріали XXI Міжнародної науково-практичної конференції ІТБ-2021. – Київ: Інжиніринг. – 228 с. С.129-132.

2) “Методологія наукових досліджень”/ Білуха М.Т.

3) “Основи наукових досліджень”/ Філіпенко А.С.

4) https://dduvs.in.ua/wpcontent/uploads/files/Structure/library/student/lectures/0926/11.1.pdf

5) https://event-camp.org/articles/conference-types/

6) https://pdatu.edu.ua/pro-universytet/naukometriya.html

7) https://research.sfu-kras.ru/quartile

8) http://easy-english.com.ua/ukrayinski-imena-anglijskoyu-movoyu/

9) http://ceur-ws.org/

10) https://dblp.org/

11) https://blog.ringostat.com/ru/parsing-dannyh-s-saytov-chto-eto-i-zachem-on-nuzhen/

12) https://aboutmarketing.info/internet-marketynh/instrumenty/shcho-take-parsynh/

13) https://romi.center/ru/learning/article/what-is-data-parsing

14) https://naurok.com.ua/prezentaciya-diagrami-uml-diagrami-precedentiv-238715.html

15) https://www.youtube.com/watch?v=zid-MVo7M-E

16) https://uk.wikipedia.org/wiki/Діаграма_прецедентів

17) https://itteach.ru/bpwin/metodologiya-idef0

18) https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D0%B8%D0%B9_%D0%B2%D0%B8%D1%80%D0%B0%D0%B7

19) https://www.crummy.com/software/BeautifulSoup/bs4/doc.ru/bs4ru.html

20) https://docs-python.ru/packages/paket-beautifulsoup4-python/metody-find-all/

21) https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/

22) https://pypi.org/project/PyPDF2/

23) https://python-scripts.com/tkinter

24) https://uk.wikipedia.org/wiki/Scopus

25) https://uk.wikipedia.org/wiki/Web_of_Science

26) Шейко В. М., Кушнаренко Н. М. Організація та методика науково-дослідницької діяльності: підручник для вищих навчальних закладів / В.М. Шейко, Н.М.Кушнаренко - Х: ХДАК, 1998. – 288 с. (1998)

27) Волкович В. Л. Проблеми створення інтелектуальних систем підтримки прийняття рішень. – К., 1990. – 190 с. (1990)

28) Макаров І.М., Виноградська Т.М., Рубчинський А.А., Соколов В.В. Теорія вибору та прийняття рішень. М.: Наука, 1982. – 330 с. (1982)

29) Литвак, Б.Г. Експертна інформація: Методи отримання та аналізу / Б.Г. Литвак. – М.: Радіо і зв’язок, 1982. – 184 с. (1982)

ДОДАТКИ

Програмний код:

```
import threading
from typing import Set
from bs4 import BeautifulSoup
import requests
import re
import PyPDF2, urllib
import io
from time import sleep
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
from dataclasses import dataclass, field
import csv
import datetime



class Parser(tk.Tk) #  клас репрезентуючий наукову подію
    @dataclass
    class Event:
        url: str  # url події
        title: str  # назва події
        articles_count: int  # кількість статей, які були написані викладачами із списку
```

```
        references_count: int  # кількість посилань на викладачів всередині статей
        authors: Set  # набор авторів статей
        references: Set  # набор викладачів, на яких були знайдені посилання
        score: float = field(init=False)  # фінальний бал події

        #  функція для обчислення фінального балу події
        def calculate_score(self):
            self.score = round(self.articles_count * Parser.article_weight +
                               self.references_count * Parser.reference_weight, 1)

        # цей метод буде викликан після __init__(). До конструктора не передається фінальна вага та її потрібно обчислити
        def __post_init__(self):
            self.calculate_score()  # обчисляємо фінальну вагу

        #  методи для сортування подій
        def __gt__(self, other):
            return self.score > other.score

        def __lt__(self, other):
            return self.score < other.score

    #  Значення вагових коефіцієнтів за замовчуванням, які юзер може редагувати
        article_weight = 1
        reference_weight = 0.1
```

```
lower_limit = 1

upper_limit = 0

latest_event = 3139


USER_AGENT = 'Mozilla/5.0'

#  Імена викладачів факультету

professors = {'Vitaliy Snytyuk', 'Natalia Tmienova', 'Vitaliy Tsyganok',
'Hryhorii Hnatiienko', 'Nikolay Kiktev',

'Oleg Ilarionov','Oksana Vlasenko', 'Georgy Gaina', 'Snizhana
Gamotska', 'Iryna Domanetska', 'Olexander Kruglov',

'Ganna Krasovska', 'Volodymir Kudin', 'Julia Minaeva', 'Julia
Nakvasiuk', 'Natalia Pashynska', 'Yuri Samokhvalov',

'Petro Soroka', 'Olena Fedusenko', 'Alexiy Bychkov', 'Denys
Berestov', 'Maxim Brazhynenko', 'Anastasia Vecherkovska',

'Sergiy Dotsenko', 'Ksenia Dukhnovska', 'Elyzaveta Zhabska',
'Liudmila Zubyk', 'Anastasia Ivanytska', 'Evgen Ivanov',

'Tetiana Kovaliuk', 'Oksana Kovtun', 'Oleg Kurchenko', 'Anna
Martsafei', 'Kateryna Merkulova', 'Anastasia Nikolaenko',

'Svetlana Popereshniak', 'Gennady Poriev', 'Maxim Tkachenko',
'Ruslan Fedorenko', 'Victor Shevchenko', 'Iryna Iurchuk',

'Natalia Lukova-Chuiko', 'Tetiana Babenko', 'Andriy Bigdan', 'Olena
Boguslavska', 'Mykola Brailovskyi', 'Sergiy Buchyk',

'Sergiy Dakov', 'Olexander Laptiev', 'Larisa Myrutenko', 'Inna
Mychalchuk', 'Volodymir Nakonechnyi', 'Ivan Parkhomenko',

'Sergiy Tolyupa', 'Olexander Toroshanko', 'Andriy Fesenko', 'Yanina
Shestak', 'Yuri Shcheblanin', 'Victor Morozov', 'Alexiy Yehorchenkov',

'Bogdan Yeremenko', 'Anna Kolomiets', 'Liubov Kubiavka', 'Tetiana
Latysheva', 'Daria Mochalova', 'Natalia Oberemok', 'Grygory Steshenko',

'Olexander Timinskyi', 'Julia Khlevna', 'Andriy Khlevnyi', 'Andriy
Onyshchenko', 'Sergiy Bronin', 'Miroslava Hladka', 'Mykola Kostikov',
```

```
'Olga Kravchenko', 'Olexander Kuchanskyi', 'Rostislav Lisnevskyi', 'Sergiy Paliy', 'Roman Ponomarenko', 'Mykhailo Stepanov',

'Ivan Chychkan', 'Yuri Kravchenko', 'Oksana Herasymenko', 'Kostyantyn Herasymenko', 'Natalia Dakhno', 'Andriy Dudnik',

'Olga Leshchenko', 'Olexander Makhovych', 'Roman Mykolaichuk', 'Olexander Pliushch', 'Olena Starkova', 'Olexander Trush',

'Valentine Pleskach', 'Julia Boiko', 'Olena Vashchilina', 'Iryna Garko', 'Victor Krasnoshchek', 'Jaroslav Kryvolapov',

'Victoria Mironova', 'Mykola Pyroh', 'Volodymir Saiko', 'Alexiy Sholokhov'}

# Прізвища викладачів для пошуку посилань в PDF-файлах. Частіш за все там зазначені тільки прізвище та ініціали

# тому пошук повного імені не дасть результату

last_names = {name.split()[-1] for name in professors}


def __init__(self):

    super(Parser, self).__init__()  # виклик батьківського конструктору


    self.events = []  # сюди будемо записувати усі оцінені події


    # починаємо розташовування єлементів інтерфейсу


    self.title('CEUR Parser')  # назва вікна

    self.resizable(False, False)

    style = ttk.Style(self)

    style.configure('Treeview', rowheight=40)  # призначаємо таблиці висоту стрічки
```

```
        #  інформаційний ярлик
        self.events_label = ttk.Label(self,
                                      text='Перейдіть до налаштувань або одразу натисніть "Старт" щоб розпочати парсинг.')
        self.events_label.grid(row=0, column=0, padx=4, pady=4)


        #  кнопка старту
        self.start_button = ttk.Button(self, text='Старт', command=self.start)
        self.start_button.grid(row=2, column=0, padx=4, pady=4)


        # таблиця подій
        columns = ('Назва', 'URL', 'Бали', 'Статей', 'Цитувань', 'Автори', 'Посилання')
        self.events_table = ttk.Treeview(self, columns=columns, show='headings', height=10)
        for column in columns:
            self.events_table.column(column, stretch=True, width=len(column) * 30, anchor=tk.CENTER)
        for column in columns:
            self.events_table.heading(column, text=column)
        self.events_table.grid(row=1, column=0, padx=10, pady=10)


        #  контекстне меню
        self.menubar = tk.Menu(self)
        self.controls = tk.Menu(self.menubar, tearoff=0)
        self.controls.add_command(label="Допомога", command=self.about, accelerator="F1")
```

```
        self.controls.add_command(label="Налаштування",
command=self.settings, accelerator="Ctrl+S")

        self.controls.add_command(label="Список викладачів",
command=self.edit_professors, accelerator="Ctrl+P")


        self.menubar.add_cascade(label="Меню", menu=self.controls)

        self.config(menu=self.menubar)

        self.bind_all("<Control-s>", self.settings)

        self.bind_all("<Control-p>", self.edit_professors)

        self.bind_all("<F1>", self.about)


        self.about()


    #  функція аналізу PDF-файлу статті за посилання

    def analyze_article(self, article_url):

        #  підключення до серверу та считування файлу

        r = urllib.request.Request(article_url, headers={'User-Agent':
self.USER_AGENT})

        try:

            article_pdf = urllib.request.urlopen(r).read()  # спроба відкрити url

        except urllib.error.HTTPError as e:

            if e.code == 404:  # якщо ресурса не існує, пропускаємо його

                return 0, set()

            elif e.code == 503:  # якщо сервер тимчасово не відповідає, чекаємо
3 секунди та пробуємо ще раз

                sleep(3)

                return self.analyze_article(article_url)
```

```
        #  считуємо файл
        article_bytes = io.BytesIO(article_pdf)
        article = PyPDF2.PdfFileReader(article_bytes)

        # ініціалізація лічильнику статей та пустого набору викладачів, який буде заповнюватися знайденими викладачами із
        # списку
        references_count = 0
        found_professors = set()

        #  Препроцессінг даних із PDF-файлу
        pages = []
        #  Проходимо усі сторінки файлу з кінця, тому що список посилань зазвичай знаходиться наприкінці статті
        for page in article.pages[::-1]:
            # прибираємо усі мусорні символи із кожної сторінки
            page = re.sub(r'[\W]', '', page.extractText())

            #  якщо дойшли до сторінки, де починаютється список посилань, виходимо з циклу
            if 'references' in page.lower():
                page = re.sub(r'.+references', '', page, 1, re.IGNORECASE)
                pages.append(page)  # додаємо сторінку до нового масиву
                break
            else:
                pages.append(page)  # додаємо сторінку до нового масиву
```

```
            #  Пошук викладачів на оброблених сторінках списку посилань
            for page in pages:
                for professor in self.last_names:
                    professor_mentions = page.count(professor)

                    #  якщо знайшли викладача на сторінці, додаємо його до набору знайдених викладачів та збільшуємо лічильник
                    if professor_mentions:
                        references_count += professor_mentions
                        found_professors.add(professor)

            return references_count, found_professors

        #  функція аналізу наукової події
        def analyze_event(self, request):
            #  використовуємо BS4 задля обробки відповіді з сервера
            soup = BeautifulSoup(request.content, 'html.parser')

            #  знаходимо назву події двома способами
            title = soup.find('span', {'class': 'CEURVOLTITLE'})
            if not title:  # якщо перший не дає результатів
                title = soup.find('h1')  # пробуємо інший
            title = title.text if title else "Не знайдено"

            #  лічильники
```

```
        articles_count = 0
        references_count = 0

        #  сюди будемо додавати знайдених викладачів
        participated_professors = set()  # якщо автор
        referenced_professors = set()  # якщо посилання

        #  знаходимо на сторінці усі статті
        articles = soup.find_all('li')
        if articles:
            #  оброблюємо кожну статтю
            for article in articles:
                #  знаходимо авторів статті першим способом
                authors = {span.text for span in article.findChildren('span', {'class':
'CEURAUTHOR'}, recursive=False)}
                #  якщо не знайшли, шукаємо другим способом
                if not authors:
                    span = article.findChild('span', {'class': 'CEURAUTHORS'})
                    if span:
                        authors = set(span.text.split(', '))
                #  якщо не знайшли, шукаємо третім способом
                if not authors:
                    i = article.findChild('i')
                    if i:
                        authors = set(i.text.split(', '))
```

```
                # знаходимо перетин набору викладачів та набору авторів статті
                matched_authors = self.professors.intersection(authors)

                # якщо набор викладачів та набор авторів статті перетинаються
                if matched_authors:
                    articles_count += 1  # збільшуємо лічильник на 1
                    pdf_link = article.findChild('a')  # знаходимо посилання на PDF
                    article_references, found_professors =
self.analyze_article(request.url + pdf_link[
                        'href'])  # аналізуємо статтю

                    references_count += article_references  # додаємо до лічильнику
події значення лічильнику статті
                    participated_professors.update(
                        matched_authors)  # додаємо нових викладачів до набору
викладачів-авторів
                    referenced_professors.update(
                        found_professors)  # додаємо нових викладачів до набору
посилань, якщо знайдені

            # створюємо екземпляр класу події та повертаємо його
            return self.Event(request.url, title, articles_count, references_count,
participated_professors,
                              referenced_professors)

        # функція оновлення даних щодо останньої події на сайті
        @staticmethod
        def get_latest_event():
```

```
        r = requests.get('http://ceur-ws.org/',
                         headers={'User-Agent': Parser.USER_AGENT})  # питаємо головну сторінку сайту
        soup = BeautifulSoup(r.content, 'html.parser')  # парсимо її
        main_table = soup.find('table', {'id': 'MAINTABLE'})  # шукаємо основну таблицю
        latest_event = main_table.findChild('a', {'name': re.compile(r'Vol-\d+')})  # беремо останню подію

        #  якщо вдалося
        if latest_event:
            return int(latest_event['name'].split('-')[-1])  # повертаємо номер знайденої події
        else:
            return 3139  # інакше повертаємо номер 3139 - номер останньої події на момент 23.05.2022

    #  функція налаштування програми
    def settings(self, event=None):

        # викликаємо нове вікно та додаємо елементи інтерфейсу
        settings_window = tk.Toplevel(self)

        #  підказки юзеру
        ttk.Label(settings_window, text='Вага статті').grid(row=0, column=0, padx=4, pady=4)
        ttk.Label(settings_window, text='Вага посилання').grid(row=1, column=0, padx=4, pady=4)
```

```
ttk.Label(settings_window, text='Номер першої події').grid(row=2, column=0, padx=4, pady=4)

ttk.Label(settings_window, text='Номер останньої події (0 - визначати автоматично)').grid(row=3, column=0,
                                                                                          padx=4,
                                                                                          pady=4)
#  поле вводу ваги статті
article_weight_entry = ttk.Entry(settings_window)
article_weight_entry.grid(row=0, column=1, padx=4, pady=4)
article_weight_entry.insert(tk.END, self.article_weight)

# поле вводу ваги посилання
reference_weight_entry = ttk.Entry(settings_window)
reference_weight_entry.grid(row=1, column=1, padx=4, pady=4)
reference_weight_entry.insert(tk.END, self.reference_weight)

# поле вводу номеру події, від якої буде проводитися парсінг
lower_limit_entry = ttk.Entry(settings_window)
lower_limit_entry.grid(row=2, column=1, padx=4, pady=4)
lower_limit_entry.insert(tk.END, self.lower_limit)

# поле вводу номеру події, до якої буде проводитися парсінг
upper_limit_entry = ttk.Entry(settings_window)
upper_limit_entry.grid(row=3, column=1, padx=4, pady=4)
upper_limit_entry.insert(tk.END, self.upper_limit)
```

```
            #  функція зберігання введених даних
            def save():
                errors = []

                try:
                    #  зчитування даних з полей
                    entered_article_weight = float(article_weight_entry.get())
                    entered_reference_weight = float(reference_weight_entry.get())
                    entered_lower_limit = int(lower_limit_entry.get())
                    entered_upper_limit = int(upper_limit_entry.get())

                    #  валідація
                    if 0 >= entered_article_weight or entered_article_weight > 1:
                        errors.append("Вага статті повинна бути в межах від 0 до 1")

                    if 0 >= entered_reference_weight or entered_reference_weight > 1:
                        errors.append("Вага посилання повинна бути в межах від 0 до
1")

                    if entered_upper_limit < 0:
                        errors.append("Остання подія не може бути негативним числом.
")

                    if entered_upper_limit > 0:
                        if 1 > entered_lower_limit or entered_lower_limit >
entered_upper_limit or entered_upper_limit > self.latest_event:
                            errors.append(
```

```
                    "Вводіть числа за наступним законом: 1 <= Перша подія
<= Остання подія <= Остання подія на сайті")
                else:
                    if 1 > entered_lower_limit or entered_lower_limit >
self.latest_event:
                        errors.append("Номер першої події повинен бути більше 0 і
менше номера останньої події.")

            except ValueError:
                errors.append('Будь ласка, використовуйте лише цифри.
Десятковий роздільник - крапка. ')

            if errors:
                messagebox.showerror('Помилка!', ';\n'.join(errors))  # показ
помилок, якщо вони присутні
            else:
                #  зберігання даних
                self.article_weight = entered_article_weight
                self.reference_weight = entered_reference_weight
                self.lower_limit = entered_lower_limit
                self.upper_limit = entered_upper_limit

                #  закриття вікна
                settings_window.destroy()

        #  дві кнопки для скасування або збереження змін
        ttk.Button(settings_window, text='Скасувати',
command=settings_window.destroy).grid(row=4, column=0, pady=4,
```

```
                                                                padx=4)
        ttk.Button(settings_window, text='Зберегти', command=save).grid(row=4, column=1, pady=4, padx=4)


    # функція внесення змін до списку викладачів
    def edit_professors(self, event=None):
        # функція збереження змін
        def save():
            try:
                user_input = textarea.get('1.0', tk.END).split(',')  # считуємо дані та сплітимо їх за комою
                #  записуємо считані дані до нового набору викладачів
                new_professors = set([professor.strip() for professor in user_input if len(professor.strip()) > 3])
                self.professors = new_professors  # зберігаємо набор як основний набор викладачів
                self.last_names = {name.split()[-1] for name in self.professors}  # оновлюємо набор прізвищ
                edit_window.destroy()  # закриваємо вікно
            except:
                label['text'] = 'Виникла помилка! Перевірте, чи правильно ви ввели імена'


        #  нове вікно та елементи інтерфейсу на ньому
        edit_window = tk.Toplevel(self)
        label = ttk.Label(edit_window, text='Введіть імена викладачів через кому')  # лейбл для вказівок
        label.grid(row=0, column=0)
```

```
        textarea = tk.Text(edit_window, width=40, height=10, wrap=tk.WORD)
        textarea.grid(row=1, column=0)
        textarea.insert(tk.END, ', '.join(self.professors))  # вставляємо поточні дані у текстбокс
        ttk.Button(edit_window, text='Зберегти', command=save).grid(row=2, column=0, pady=4,
                                                                    padx=4)  # кнопка для зберігання змін


    def about(self, event=None):
        def display_text(text):
            help_label['text'] = text

        greet_window = tk.Toplevel(self)
        greet_window.attributes('-topmost', 'true')
        about = "Ця програма дозволяє спарсити дані із сайту ceur-ws.org. Програма дає кожній науковій події певну " \
                "оцінку, яка ґрунтується на кількості шуканих людей, які брали в ній участь. Результати будуть " \
                "записані у файл. Процес парсінгу може зайняти до десяти хвилин. "
        ceur = "CEUR це ресурс для публікації та перегляду інформації про наукові конференції та події. На сайті " \
               "зберігається інформація про тисячі наукових подій, згадані десятки тисяч наукових статей та авторів. "
        settings = "У налаштуваннях можна редагувати такі параметри: Вага статті, вага посилання всередині статті, " \
                   "верхня та нижня межа подій, у яких потрібно шукати. За замовчуванням програма парсить всі події " \
```

"на сайті, а ваги статті та посилання дорівнюють 1 і 0.1 відповідно. "

professors = "Меню 'Список викладачів' дозволяє редагувати список імен, згадки про які програма шукатиме на " \

"сайті CEUR. Чим цей список менший, тим швидше буде проходити пошук. Використовуйте міжнародну " \

"транслітерацію імен латиницею, на сайті CEUR не використовується кирилиця."

parsing = "Вебскрапінг (від англ. scraping — «вишкрібання», вебзбирання або витягнення вебданих) — " \

"це перетворення у структуровані дані інформації з вебсторінок, " \

"які призначені для перегляду людиною за допомогою браузера."

ttk.Button(greet_window, text='Про програму', command=lambda: display_text(about), width=25).grid(row=0,

column=0,

padx=4,

pady=4)

ttk.Button(greet_window, text='Про CEUR', command=lambda: display_text(ceur), width=25).grid(row=1, column=0,

padx=4,

pady=4)

ttk.Button(greet_window, text='Про налаштування', command=lambda: display_text(settings), width=25).grid(row=2,

column=0,

padx=4,

pady=4)

```
        ttk.Button(greet_window, text='Про список викладачів',
command=lambda: display_text(professors), width=25).grid(
            row=3,
            column=0,
            padx=4,
            pady=4)
        ttk.Button(greet_window, text='Про вебскрейпінг', command=lambda:
display_text(parsing), width=25).grid(row=4,
                                                                column=0,
                                                                padx=4,
                                                                pady=4)

        help_label = ttk.Label(greet_window, text=about, wraplength=200,
justify=tk.LEFT, font='Arial 12')
        help_label.grid(row=0, column=1, rowspan=5, padx=10, pady=10)

    # функція парсингу
    def parse(self, event=None):
        self.events = []  # очищення списку подій
        self.update_table()  # очищення таблиці
        self.events_label['text'] = "З'єднання з сервером..."
        #  оновлення номеру останньої події
        if self.upper_limit <= 0 or self.upper_limit > self.latest_event:
            self.upper_limit = self.latest_event = self.get_latest_event()

        # парсінг подій
        current_event = self.lower_limit
```

```
            while current_event <= self.upper_limit:
                url = f'http://ceur-ws.org/Vol-{current_event}'  # URL поточної події
                r = requests.get(url, headers={'User-Agent': self.USER_AGENT})  # робимо запрос на сервер
                if r:  # якщо запрос вдалий
                    self.update_counter(current_event)
                    event = self.analyze_event(r)  # аналізуємо наукову подію подію
                    if event.score > 0:  # якщо бали більше нуля, подія містить викладачів, які нас цікавлять
                        self.events.append(event)  # додаємо подію до списку
                        self.update_table()  # оновлюємо таблицю
                    current_event += 1
                elif r.status_code == 404:  # якщо ресурсу не існує, пропускаємо його
                    current_event += 1
                else:  # якщо сервер відповів відмовою, чекаємо 5 секунд та питаємо ще раз
                    sleep(5)

            #  запис до файлу
            self.write_to_file()

            #  вмикаємо елементи управління
            self.start_button['state'] = 'normal'
            self.controls.entryconfig('Налаштування', state='normal')
            self.controls.entryconfig('Список викладачів', state='normal')

        # функія старту процесу парсінгу в окремому потоці
```

```
        def start(self):
            parse_thread = threading.Thread(target=self.parse)  # починаємо парсінг в окремому треді
            parse_thread.start()

            # вимикаємо елементи управління
            self.start_button['state'] = 'disabled'
            self.controls.entryconfig('Налаштування', state='disabled')
            self.controls.entryconfig('Список викладачів', state='disabled')

        # функція оновлювання таблиці подій
        def update_table(self):
            self.events_table.delete(*self.events_table.get_children())  # видаляємо дані з таблиці

            #  додаємо кожну подію з відсортованого списку подій
            for event in sorted(self.events, reverse=True):
                self.events_table.insert('', tk.END, values=(
                    event.title, event.url, event.score,
                    f'{event.articles_count} (+{(event.articles_count * Parser.article_weight):.1f} до суми балів)',
                    f'{event.references_count} (+{(event.references_count * Parser.reference_weight):.1f} до суми балів)',
                    ', '.join(event.authors), ', '.join(event.references)))

        # функція для оновлення інформаційного ярлику
        def update_counter(self, current_event):
```

```
            if self.events_label:
                self.events_label['text'] = f"Поточна подія: {current_event} /
{self.upper_limit}"


        def write_to_file(self):
            with
open(f"{datetime.datetime.now().strftime('%d%m%Y%H%M%S')}.csv", 'w',
encoding='utf8',
                      newline='') as csvfile:
                writer = csv.writer(csvfile, delimiter=',')
                writer.writerow(self.professors)
                writer.writerow([f"Article weight = {self.article_weight}", f"Reference
weight = {self.reference_weight}",
                                 f"Lower limit = {self.lower_limit}", f"Upper limit =
{self.upper_limit}"])
                writer.writerow(['Title', 'URL', 'Score', 'Articles count', 'References
count', 'Authors', 'References'])
                for event in sorted(self.events, reverse=True):
                    writer.writerow([event.title, event.url, event.score,
                                     f'{event.articles_count} (+{(event.articles_count *
Parser.article_weight):.1f} to final score)',
                                     f'{event.references_count} (+{(event.references_count *
Parser.reference_weight):.1f} to final score)',
                                     ', '.join(event.authors), ', '.join(event.references)])



    app = Parser()
    app.mainloop()
```