

УДК 004.912

DOI <https://doi.org/10.32689/maup.it.2023.3.10>

### Якіє ЮСИН

доктор філософії з інженерії програмного забезпечення, асистент кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Берестейський проспект, 37, Київ, Україна, індекс 03056 (yusyn@pzks.fpm.kpi.ua)

ORCID: 0000-0001-6971-3808

### Yakiv YUSYN

PhD in Software Engineering, Assistant of the Computer Systems Software Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Prospect Beresteyskiy, 37, Kyiv, Ukraine, postal code 03056 (yusyn@pzks.fpm.kpi.ua)

**Бібліографічний опис статті:** Юсин, Я. (2023). Автоматизація процесу аналізу та перевірки студентських кваліфікаційних робіт. *Інформаційні технології та суспільство*, 3, 72–79. DOI: <https://doi.org/10.32689/maup.it.2023.3.10>

**Bibliographic description of the article:** Yusyn, Y. (2023). Avtomatyzatsiia protsesu analizu ta perevirky studentskykh kvalifikatsiinykh robit [Automation of the process of analysis and checking of student qualification papers]. *Informatsiini tekhnolohii ta suspilstvo – Information technology and society*, 3, 72–79. DOI: <https://doi.org/10.32689/maup.it.2023.3.10>

## АВТОМАТИЗАЦІЯ ПРОЦЕСУ АНАЛІЗУ ТА ПЕРЕВІРКИ СТУДЕНТСЬКИХ КВАЛІФІКАЦІЙНИХ РОБІТ

**Анотація. Проблематика.** Поширеною формою атестації здобувачів вищої освіти є виконання кваліфікаційної роботи, невід'ємною складовою якої є її текстова частина. На сьогодні процес перевірки текстової частини є практично не автоматизованим і ґрунтується на зусиллях працівників випускних кафедр. **Мета дослідження.** Зменшити показник середньої кількості помилок у кваліфікаційних роботах за рахунок можливості виконання перевірки студентами самостійно, під час написання робіт, шляхом автоматизації процесу перевірки. **Методика реалізації.** Помилки, що можуть допускатися студентами при написанні кваліфікаційних робіт, розподілено на три групи: помилки технічного оформлення, помилки логічного оформлення, граматичні помилки. Для кожної групи розглянуто існуючі інструменти, які можуть запобігати та виявляти помилки цієї групи, і визначено ті класи помилок, на запобігання та виявлення яких інформаційні системи автоматичної перевірки мають спеціалізуватися. В якості прикладу, за допомогою хмарних технологій, розроблено таку інформаційну систему, що реалізовує 9 правил виявлення помилок. **Результати дослідження.** Оцінювання ефективності автоматизації процесу перевірки кваліфікаційних робіт виконано на прикладі розробленої інформаційної системи за допомогою аналізу шістдесят чотирьох випускних робіт двох минулих років, що виконувались студентами двох різних структурних підрозділів. Для переважної більшості реалізованих правил (сім із дев'яти) отримано ненульову кількість істинних спрацювань. При цьому, кількість істинних спрацювань менше кількості хибних спрацювань лише для одного правила серед цих семи. **Висновки.** Експериментальна перевірка розробленої інформаційної системи доказала ефективність (за критерієм зменшення показника середньої кількості помилок) автоматизації процесу аналізу кваліфікаційних робіт, адже всі виявлені помилки могли би бути виявленими ще під час написання робіт студентами самостійно.

**Ключові слова:** кваліфікаційна робота; аналіз кваліфікаційних робіт; перевірка граматичної; перевірка оформлення; нормоконтроль; хмарні технології; безсерверна архітектура.

## AUTOMATION OF THE PROCESS OF ANALYSIS AND CHECKING OF STUDENT QUALIFICATION PAPERS

**Abstract. Background.** A common form of attestation for students of higher education is the performance of a qualification work, an integral component of which is its textual part. Today, the process of checking the text part is practically not automated and is based on the efforts of employees of graduate departments. **Objective.** To reduce the average number of errors in qualifying papers due to the possibility of self-checking by students during the writing of papers, by automating the checking process. **Methods.** Errors that can be made by students when writing qualification papers are divided into three groups: errors of technical design, errors of logical design, and grammatical errors. For each group, the existing tools that can prevent and detect errors of this group are considered, and those classes of errors are defined, on which prevention and detection of automatic verification information systems should be specialized. As an example, with the help of cloud technologies, such an information system was developed that implements 9 error detection rules. **Results.** The evaluation of the effectiveness of the automation of the process of checking qualification papers was carried out on the example of the developed information system through the analysis of sixty-four qualification papers of the past two years, which were performed by students of two different structural units. For the vast majority of implemented rules (seven out of nine), a non-zero number of true activations was obtained. At the same time, the number of true activations is less than the number of false activations for only one rule among these seven. **Conclusions.** The experimental verification of the developed information system proved the effectiveness (according to the criterion of reducing the average number of errors)

of the automation of the process of analysis of qualifying papers because all the detected errors could have been detected even during the writing of the papers by the students themselves.

**Key words:** qualification paper; analysis of qualification papers; spell checking; design checking; normative control; cloud technologies; serverless architecture.

### **Вступ.**

Однією із форм атестації здобувачів вищої освіти, що використовується в Україні, є виконання кваліфікаційної роботи. «Кваліфікаційна робота – це засіб діагностики ступеня сформованості компетентностей щодо вирішення типових завдань діяльності згідно з вимогами стандартів вищої освіти. Видами кваліфікаційних робіт є: дипломний проєкт, дипломна робота, магістерська дисертація» [1]. Виконання кваліфікаційної роботи передбачає «... систематизацію, закріплення, розширення теоретичних і практичних знань зі спеціальності та застосування їх при вирішенні конкретних наукових, технічних, економічних виробничих та інших завдань» [2]. Вибір форми атестації здобувачів вищої освіти у вигляді написання кваліфікаційної роботи визначається освітньою програмою, на основі вимог стандартів вищої освіти.

Обов'язковою складовою виконання кваліфікаційної роботи є написання та оформлення текстової частини – пояснювальної записки та іншої супровідної документації. На сьогодні, якість текстової частини (відповідність оформлення і змісту встановленим вимогам) забезпечується працею самого студента, контролем його керівника та нормоконтролюванням. Єдиним процесом забезпечення якості документації кваліфікаційної роботи, який наразі широко автоматизований, є процес перевірки дотримання принципів академічної доброчесності за допомогою спеціалізованих інформаційних систем [3; 4]. Всі інші процеси виконуються вручну, тому потребують багато людських зусиль та часу, є не захищеними від помилок. Також це призводить до великої затримки надання зворотного зв'язку студенту: проміжок часу між написанням текстової частини та виявленням помилок у ній в найкращому випадку складає дні, а в найгіршому випадку може сягати місяців (якщо певні помилки будуть знайдені лише під час завершального нормоконтролювання).

Таким чином, розроблення спеціалізованих інформаційних систем для автоматизації процесу перевірки кваліфікаційних робіт є актуальною задачею, вирішення якої підвищить якість робіт, а також зменшить навантаження на науково-педагогічних працівників підрозділів, які є керівниками та/або виконують нормоконтролювання.

### **Постановка задачі.**

Метою роботи є удосконалення процесу перевірки студентських кваліфікаційних робіт шляхом автоматизації цього процесу, що зменшить показник середньої кількості помилок у роботах за рахунок можливості виконання перевірки студентами самостійно, під час написання робіт, та за рахунок зменшення людського фактору.

### **Теоретичні аспекти аналізу кваліфікаційних робіт.**

Всі помилки, які можуть бути наявними у кваліфікаційних роботах, можливо розділити на три групи: помилки технічного оформлення, помилки логічного оформлення та граматичні помилки. Розглянемо кожну із груп помилок та програмні засоби, які можуть застосовуватися для запобігання та/або виявлення таких помилок.

До помилок технічного оформлення можливо віднести всі помилки, які пов'язані з порушенням технічних вимог до документації кваліфікаційної роботи – розмір та поля сторінок, відступи, шрифти, міжрядкові інтервали і так далі. Дані вимоги задаються галузевими стандартами, розробленими на їх основі положеннями окремих вищих навчальних закладів або їх підрозділів, тощо. Для запобігання таких помилок можуть застосовуватися шаблони оформлення кваліфікаційних робіт (наприклад, Word або TeX), які будуть містити правильно задані стилі всіх структурних елементів роботи, та які будуть надаватися студентам. Проте, як показує аналіз, практика підготовки таких шаблонів не є поширеною; також, надавання шаблонів не гарантує те, що студент буде його правильно використовувати чи використовувати взагалі – тому це потребує додаткової ручної перевірки.

До помилок логічного оформлення можливо віднести всі помилки, які пов'язані з порушенням встановлених вимог до оформлення документації кваліфікаційної роботи, але які не пов'язані з технічними вимогами. Наприклад, такими вимогами може бути наявність певних структурних елементів документації, їх іменування або присвоєння шифрів, наявність посилань у тексті роботи на всі джерела із списку використаних джерел, тощо. Існуючі на сьогодні програмні засоби здатні запобігати або виявляти лише незначну частку від можливих помилок цієї групи. Серед них можливо виділити наступне:

- Системи перевірки на плагіат можуть перевіряти правильність цитування джерел, наявність посилань на список використаних джерел, але це не є їх основною функцією та перевірка за допомогою таких систем виконується в кінці написання роботи.

- Деякі текстові процесори надають можливість керування джерелами, що включає у себе автоматичне створення та форматування бібліографії, посилань, підсвітку невикористаних джерел; також для цього існує спеціалізоване програмне забезпечення.

- Також текстові процесори можуть надавати інші функціональні можливості, що можуть запобігти виникненню помилок даної групи, такі як автоматичне створення змісту, підписів рисунків (з їх нумерацією), тощо.

Проте, на практиці забезпечити використання вище описаних функціональних можливостей текстових процесорів (або спеціалізованого програмного забезпечення) може бути неможливим з різних причин: відсутність підтримки у всіх текстових процесорах, несумісність наявного у студента апаратного чи програмного забезпечення з текстовим процесором, тощо.

До граматичних помилок можливо віднести всі помилки, які пов'язані з порушенням правил правопису (орфографії та пунктуації), стилістичні помилки. На сьогодні більшість текстових процесорів надають вбудовані можливості перевірки правопису, у тому числі українською мовою, а для систем комп'ютерної типографії (таких як TeX) існують відповідні розширення. Перевірка правопису у текстових процесорах ввімкнута за замовчуванням, тому вона здатна запобігти більшості помилок даної групи (хоча вона також може мати певні недоліки – наприклад, не відповідати останній прийнятій редакції правопису).

На основі даного огляду можемо зробити висновок, що у інформаційній системі аналізу кваліфікаційних робіт доцільно зосередитись на підтримці виявлення помилок перших двох груп. Також доцільною є підтримка виявлення помилок третьої групи, але не у повному обсязі: варто зосередитись на специфічних для предметної галузі правилах та правилах, які не підтримуються текстовими процесорами (наприклад, відповідність останній редакції правопису).

За наявності інформаційної системи аналізу та перевірки кваліфікаційних робіт, робочий процес дипломного керівника із студентом може будуватися наступним чином:

1. Студент завершує написання документації кваліфікаційної роботи або її частини (наприклад, певний розділ).
2. Студент самостійно перевіряє свій.docx файл з текстом за допомогою інформаційної системи аналізу.
3. Студент виправляє знайдені помилки.
4. Студент надсилає файл керівнику на перевірку.
5. Керівник встановлює факт перевірки студентом свого тексту.
6. Керівник перевіряє документацію фокусуючись на тих аспектах роботи, що автоматично не перевіряються.

Перед проходженням нормоконтролю, після завершення кваліфікаційної роботи, може відбуватися фінальна перевірка.

#### **Огляд розробленої інформаційної системи аналізу та перевірки.**

В якості прикладу інформаційної системи аналізу та перевірки кваліфікаційних робіт розглянемо систему, розроблену для робіт з ІТ спеціальностей.

#### **Архітектура.**

Розроблена інформаційна система реалізована у вигляді класичного вебзастосунок. Даний вибір обґрунтовано тим, що вебзастосунок взагалі не потребує окремого встановлення студентами (потрібен лише сучасний веббраузер), тому помилки сумісності з наявним у студента апаратним чи програмним забезпеченням виникнути не можуть. Крім того, у випадку вебзастосунку значно спрощується процес оновлення, оскільки є лише одне місце розгортання (сервер).

Інформаційна система розроблена з використанням хмарних технологій та безсерверної архітектури (так звані безсерверні обчислення). В контексті предметної галузі аналізу документації кваліфікаційних робіт, безсерверна архітектура має як технічні (спрощення апаратної інфраструктури та операцій з нею, спрощення розгортання), так і економічні переваги (оскільки сплачуються лише де-факто спожиті ресурси під час виконання аналізу) [5; 6].

Основна логіка аналізу реалізована на платформі.NET мовою програмування C#. Графічний інтерфейс користувача реалізовано за допомогою HTML/CSS, з інтерактивною логікою реалізованою на мові програмування JavaScript. Схематичне зображення розробленої системи, розбитої на компоненти, наведено на рис. 1.

Головною одиницею розбиття логіки аналізу та перевірки на компоненти є сервіс. В контексті розробленого програмного забезпечення, сервіс – це компонент, який реалізовує одне або декілька правил

аналізу, об'єднаних одним предметом, що аналізується. Наприклад, це може бути сервіс аналізу посилань, сервіс аналізу пунктуації, тощо. Сервіси між собою не взаємодіють, тому для спрощення повторного використання загального коду між сервісами, такий код винесено в окремі компоненти.

Кожному сервісу аналізу відповідає одна безсерверна функція, яка слугує публічно доступним інтерфейсом до нього. Кожна безсерверна функція запускається за допомогою HTTP-запиту, в тілі якого передається.docx файл для аналізу.



Рис. 1. Схематичне зображення компонентів розробленої системи

Враховуючи вищеописані особливості серверної частини, клієнт (графічний інтерфейс) не містить обчислювально складної логіки: лише викликає кожну безсерверну функцію (надсилаючи обраний користувачем.docx файл) та відображає отримані результати.

Серверна частина розробленої інформаційної системи розгорнута за допомогою хмарного провайдеру Azure [7], а клієнтську частину розгорнуто на сервісі розміщення статичних сторінок GitHub Pages [8]. Обидві частини розгортаються автоматично при внесенні змін до головної гілки репозиторію коду, де ведеться розроблення інформаційної системи [9].

#### **Реалізовані правила.**

На момент написання даної роботи, розроблена інформаційна система аналізу та перевірки кваліфікаційних робіт з ІТ спеціальностей реалізує дев'ять правил, згрупованих по семи сервісам.

**Сервіс аналізу посилань.** Сервіс аналізу посилань реалізує одне правило REF01, яке перевіряє наявність посилань на використані джерела у тексті розділу чи кваліфікаційної роботи загалом. Правило підтримує посилання за стандартами ДСТУ [10] та APA [11], виконуючи їх пошук за допомогою регулярних виразів.

**Сервіс аналізу розмітки документу.** Даний сервіс реалізує два правила LAY01 та LAY02, які перевіряють базові налаштування розмітки документів – встановлені розмір (відповідність формату А4) та поля сторінки (відповідність встановленим вимогам).

**Сервіс аналізу правопису 2019-го року.** Даний сервіс реалізує правило ORT01, яке перевіряє відповідність тексту кваліфікаційної роботи змінам, що були внесені з новим українським правописом 2019-го року [12]. Доцільність реалізації даного правила зумовлена тим, що з новим правописом були внесені зміни у написання слів «проект» та префіксу «веб-», які дуже часто зустрічаються у документації кваліфікаційних робіт з ІТ спеціальностей. Пошук застарілого написання слів виконується за допомогою списку регулярних виразів.

**Сервіс аналізу використання кальки англомовних слів.** Даний сервіс реалізує правило ORT02, яке перевіряє відсутність жаргонізмів у тексті кваліфікаційної роботи. Такі жаргонізми – утворені за допомогою кальки англомовних слів, наприклад, «деплой» – поширені в усній мові в індустрії ІТ, проте для письмової документації їх використання є неприпустимим. Пошук жаргонізмів реалізовано за допомогою списку регулярних виразів, що доповнюється при зустрічі нових форм на практиці.

**Сервіс аналізу використання слів у правильному контексті.** Даний сервіс реалізує правило ORT03. Дане правило є подібним до правила ORT02, але тут перевіряються певні україномовні слова, які часто вживаються студентами у неправильному значенні або контексті (наприклад, слово «додаток» у значенні «застосунок»). На даний час, це правило реалізовано тільки за допомогою регулярних виразів.

**Сервіс аналізу змішування символів різних алфавітів.** Даний сервіс реалізує правило ORT04, яке перевіряє відсутність змішування символів різних алфавітів у одному слові (наприклад, викори-

стання «а» латиницею замість кирилиці). Таке змішування може виникати з різних причин (це не обов'язково є спробою обманути системи перевірки плагіату), тому є доцільним включення такої перевірки до розробленої інформаційної системи, для попередньої перевірки. Дане правило реалізовано за допомогою регулярних виразів.

**Сервіс аналізу пунктуації.** Даний сервіс реалізовує два правила PNC01 та PNC02, які перевіряють використану у тексті кваліфікаційної роботи пунктуацію, а саме: коректне виділення розділових знаків пробілами; використання лапок-ялинок, які є найбільш поширеними в українській мові.

**Експеримент.**

**Опис експерименту.**

Для експериментальної перевірки ефективності розробленої інформаційної системи вирішено виконати перевірку робіт минулих років. Якщо така перевірка здатна виявити достатню кількість помилок у вже перевірених людьми роботах, то, відповідно, розроблене програмне забезпечення здатне виявити всі ці помилки в автоматичному режимі під час написання робіт, таким чином запобігши їх появі.

Недоліком такого експерименту є те, що розподіл виявлених помилок між правилами буде відрізнятися від того, який буде отриманий з використанням розробленої інформаційної системи під час написання кваліфікаційних робіт. Це пов'язано з тим, що певні типи помилок людині виявити легше, ніж інші; відповідно, у вже прийнятих роботах такі типи помилок будуть зустрічатися рідше. Проте, даний недолік не має великого значення, адже ціллю експерименту є принципова перевірка ефективності розробленого програмного забезпечення, тобто здатність знаходити помилки в цілому.

Для проведення експерименту використано актуальну на той момент версію розробленого програмного забезпечення 1.0.0-beta.2.2.1.

**Експериментальні дані та їх підготовка.**

В якості вхідних даних для проведення експерименту використано 64 бакалаврських дипломних проєктів, що виконувалися у 2021 та 2022 роках у двох різних структурних підрозділах.

Оскільки бакалаврські дипломні проєкти було представлено у форматі.pdf, аналіз якого не підтримується розробленою інформаційною системою (адже його метою є аналіз вихідних файлів під час написання кваліфікаційної роботи, а формат.pdf є кінцевим форматом для подання результатів), всі файли конвертовано у формат.docx за допомогою стороннього застосунку. Також з отриманих файлів.docx було видалено елементи, які не підлягають аналізу реалізованими правилами – титульні аркуші, списки використаних джерел, додатки, тощо.

**Результати.**

Отримана статистика по кількості спрацювань кожного із підтримуваних розробленою інформаційною системою правил наведено на рис. 2.

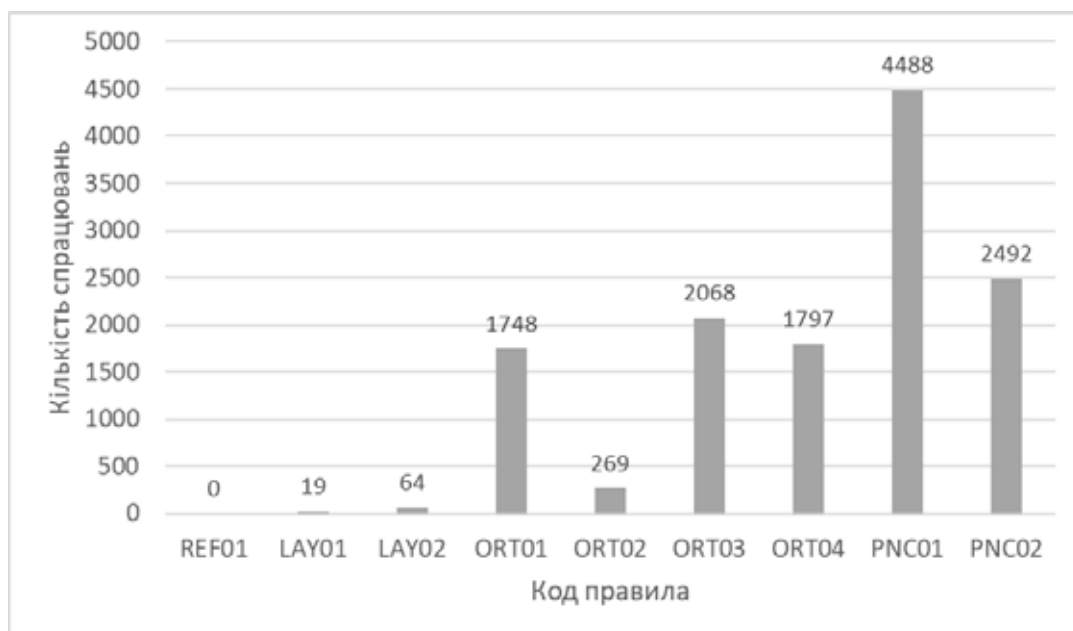


Рис. 2. Результати аналізу

Далі кожне спрацювання правил було проаналізовано і розподілено до однієї з трьох наступних категорій:

- істинне спрацювання (I);
- хибне спрацювання, викликане недоліками розробленої програмної системи (X1);
- хибне спрацювання, викликане недоліками вхідних даних (X2).

Наявність третьої категорії викликана тим, що дані для проведення експерименту конвертовано з кінцевих файлів формату.pdf, а не використовувалися вихідні файли формату.docx. Процес конвертації не завжди може правильно відтворити оригінальну структуру документу, тому за умови використання вихідних файлів формату.docx, цих спрацювань не було би отримано взагалі (детальний опис таких спрацювань наведено далі).

Отримані результати класифікації всіх спрацювань за описаними категоріями наведено на рис. 3.

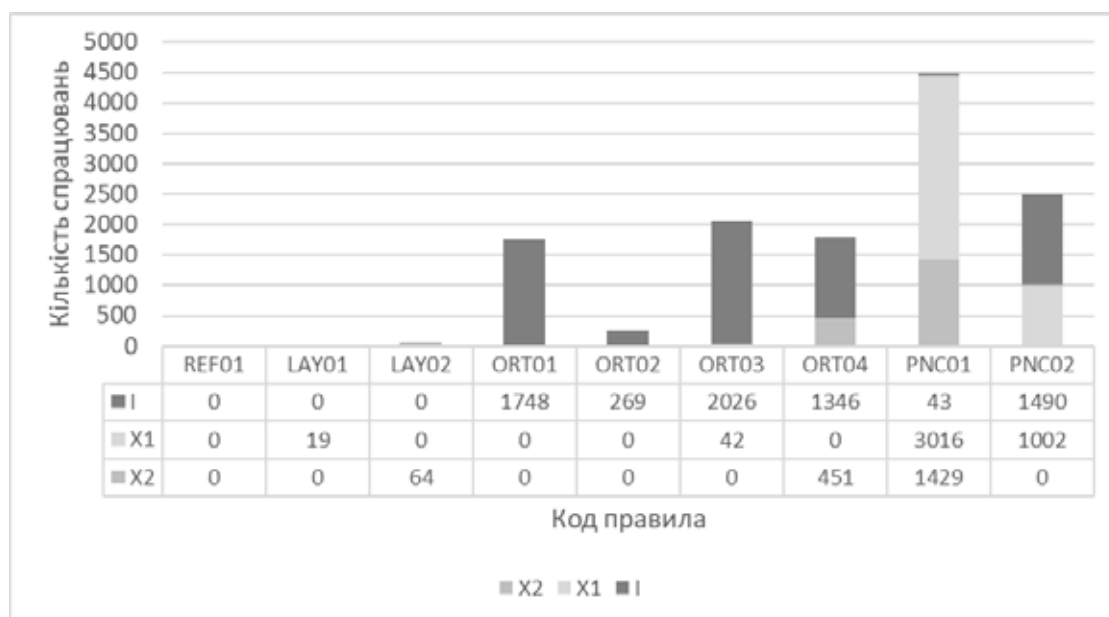


Рис. 3. Розподіл отриманих спрацювань за трьома категоріями

#### Обговорення результатів.

Розглянемо отримані результати для кожного із реалізованих правил.

**Відсутність посилань (REF01).** Кількість отриманих спрацювань даного правила дорівнює нулю. Нульова кількість спрацювань даного правила на вже прийнятих роботах є очікуваною, адже відсутність у тексті посилань на використані джерела легко перевіряється людиною, і така робота не буде допущеною до захисту.

**Розмір сторінки (LAY01).** Перевірка робіт на відповідність розміру сторінки формату A4 спрацювала 19 разів. Після аналізу дані спрацювання віднесено до категорії «X1», тобто викликаних недоліками розробленої інформаційної системи. Це пов'язано з особливостями внутрішнього подання.docx файлу, у якому розміри сторінки зберігаються у двадцятих частинах пункту [13].

Оскільки одна двадцята частина пункту приблизно дорівнює 0.0017 сантиметра (1/1440 дюйма), розроблене програмне забезпечення повинно перевіряти відповідність розміру сторінки з досить великим допустимим відхиленням (у двадцятих частинах пункту) від значень, що відповідають розмірам формату A4.

**Поля сторінки (LAY02).** Дане правило спрацювало для кожної роботи із набору експериментальних даних. Всі спрацювання віднесено до хибних спрацювань, викликаних недоліками вхідних даних (категорія «X2»), оскільки, як показав аналіз, використане для конвертації з.pdf у.docx програмне забезпечення неправильно розпізнало поля документів.

**Правопис 2019-го року (ORT01), використання кальки англomовних слів (ORT02).** Всі отримані спрацювання двох правил є істинними.

**Використання слів у неправильному контексті (ORT03).** Більшість отриманих спрацювань даного правила є істинними. Невелика частина (близько 2%) спрацювань віднесено до категорії «X1», тобто викликаних недоліками розробленого програмного забезпечення.

Для виправлення зазначених хибних спрацювань, необхідно удосконалити використовуваний регулярний вираз для слова «додаток», відсікаючи посилання на додатки до роботи, що пронумеровані літерами.

**Змішування символів різних алфавітів (ORT04).** Близько 33% спрацювань даного правила викликані невеликими програмними забезпеченнями, використаними для конвертації файлів.pdf у.docx, які призвели до склеювання декількох слів у одне слово.

**Виділення пунктуації (PNC01).** Спрацювання, віднесені до категорії «X2», викликані вище описаним невеликим програмним забезпеченням для конвертації файлів у формат.docx, який призводить до склеювання слів. За їх виключенням, переважна більшість спрацювань (98%) віднесено до категорії «X1», у зв'язку з хибним спрацюванням правила на шифри робіт, що наводяться на кожній сторінці.

**Використані лапки (PNC02).** Близько 40% спрацювань даного правила віднесено до категорії «X1». Ці хибні спрацювання викликані лістингами коду, які автори робіт наводили у тексті – оскільки мови програмування підтримують лише один вид лапок для рядкових літералів, це призводить до хибних спрацювань. Виправити це можливо за допомогою додаткового автоматичного попереднього оброблення тексту роботи, що буде знаходити та видаляти лістинги коду.

#### **Висновки.**

Проведений аналіз поточного стану автоматизації процесу перевірки кваліфікаційних робіт показав доцільність розроблення спеціалізованих інформаційних систем для автоматизації цього процесу. У дальній перспективі такі інформаційні системи, у тому числі, можуть повністю замінити процес ручного нормоконтролювання для кваліфікаційних робіт.

В даному дослідженні розглянуто класифікацію помилок, що можуть допускатися студентами при написанні кваліфікаційних робіт, та можливість їх запобігання та виявлення існуючими інструментами. На даній основі сформульовано вимоги до інформаційних систем автоматичної перевірки кваліфікаційних робіт, а саме визначено ті класи помилок, на запобігання та виявлення яких такі ІС мають спеціалізуватися.

В якості прикладу розроблено інформаційну систему аналізу кваліфікаційних робіт з ІТ спеціальностей, що реалізує 9 правил виявлення помилок. Інформаційна система побудована у вигляді класичного вебзастосунку, з використанням хмарних безсерверних технологій. Такий вибір обґрунтований технічними (спрощення розгортання, відсутність необхідності в налаштуванні та моніторингу серверу), організаційними (студентам не потрібно встановлювати програмне забезпечення для аналізу, немає необхідності у вирішенні проблем сумісності) та економічними (більшість часу ІС не використовується, а безсерверна технологія дає можливість сплачувати лише за фактично спожиті ресурси) перевагами такої архітектури.

Проведена експериментальна перевірка ефективності розробленої інформаційної системи аналізу кваліфікаційних робіт полягала у виконанні аналізу кваліфікаційних робіт минулих років. Перевірка шістдесят чотирьох робіт, виконаних студентами протягом двох років у двох різних структурних підрозділах, виявила достатню кількість істинних спрацювань більшості реалізованих правил. Це означає, що використання спеціалізованої інформаційної системи ефективно виявило і запобігло би цим помилкам ще під час написання кваліфікаційних робіт.

Розроблена інформаційна система аналізу кваліфікаційних робіт має значний потенціал для подальшого розвитку, наприклад:

- зменшення кількості хибних спрацювань існуючих правил, наприклад, за рахунок розроблення механізму автоматичного виключення програмного коду з аналізу;
- розроблення нових правил для виявлення та запобігання помилок оформлення, граматичних помилок, у тому числі, зі застосуванням методів оброблення природної мови;
- впровадження нових типів аналізу – наприклад, виявлення текстів, написаних за допомогою засобів штучного інтелекту;
- розширення супровідної до аналізу функціональності – наприклад, автоматичне виправлення деяких помилок.

#### **Список використаних джерел:**

1. Положення про екзаменаційну комісію та атестацію здобувачів вищої освіти в КПІ ім. Ігоря Сікорського. *КПІ ім. Ігоря Сікорського*. Київ, 2020.
2. Положення про організацію освітнього процесу у Київському національному університеті імені Тараса Шевченка (друга редакція). *КНУ ім. Тараса Шевченка*. Київ, 2022.
3. Положення про систему виявлення та запобігання академічному плагіату у Київському національному університеті імені Тараса Шевченка. *КНУ ім. Тараса Шевченка*. Київ, 2020.

4. Положення про систему запобігання академічному плагиату в КПІ ім. Ігоря Сікорського. *КПІ ім. Ігоря Сікорського*. Київ, 2020.
5. Roberts M. Serverless Architectures // martinFowler.com. 2018. URL: <https://martinfowler.com/articles/serverless.html>.
6. Wen J. et al. Rise of the Planet of Serverless Computing: A Systematic Review // *ACM Transactions on Software Engineering and Methodology*. 2023. 5 (32). P. 1-61.
7. Azure Functions – Serverless Apps and Computing. *Microsoft Azure*. 2016. URL: <https://azure.microsoft.com/en-us/services/functions/>.
8. GitHub Pages. *GitHub*. 2009. URL: <https://pages.github.com/>.
9. GitHub. Features. GitHub Actions. *GitHub*. 2018. URL: <https://github.com/features/actions>.
10. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. *ДСТУ*. 2015.
11. Publication Manual of the American Psychological Association, Seventh Edition. *American Psychological Association*. 2020.
12. Український правопис. *Українська національна комісія з питань правопису*. 2019.
13. ISO/IEC 29500-1:2016. Information technology – Document description and processing languages – Office Open XML File Formats – Part 1: Fundamentals and Markup Language Reference. *ISO*. 2016.

#### References:

1. Igor Sikorsky Kyiv Polytechnic Institute. (2020). Polozhennia pro ekzamenatsiinu komisiuu ta atestatsiuiu zdobuvachiv vyshchoi osvity v KPI im. Ihoria Sikorskoho [Regulations on the examination board and attestation of applicants for higher education at Igor Sikorsky Kyiv Polytechnic Institute]. Kyiv [in Ukrainian].
2. Taras Shevchenko Kyiv National University. (2022). Polozhennia pro orhanizatsiuiu osvitnoho protsesu u Kyivskomu natsionalnomu universyteti imeni Tarasa Shevchenka (druha redaktsiia) [Regulations on the organization of the educational process at Taras Shevchenko Kyiv National University (second edition)]. Kyiv [in Ukrainian].
3. Taras Shevchenko Kyiv National University. (2020). Polozhennia pro systemu vyivlennia ta zapobihannia akademichnomu plahiatu u Kyivskomu natsionalnomu universyteti imeni Tarasa Shevchenka [Regulations on the system of detection and prevention of academic plagiarism at Taras Shevchenko Kyiv National University]. Kyiv [in Ukrainian].
4. Igor Sikorsky Kyiv Polytechnic Institute. (2020). Polozhennia pro systemu zapobihannia akademichnomu plahiatu v KPI im. Ihoria Sikorskoho [Regulations on the system of prevention of academic plagiarism at Igor Sikorsky Kyiv Polytechnic Institute]. Kyiv [in Ukrainian].
5. Roberts, M. (2018). Serverless Architectures. martinFowler.com: <https://martinfowler.com/articles/serverless.html>.
6. Wen, J., Chen, Z., Jin, X., & Liu, X. (2023). Rise of the Planet of Serverless Computing: A Systematic Review. *ACM Transactions on Software Engineering and Methodology*, 32(5), pp. 1-61. <https://doi.org/10.1145/3579643>.
7. Microsoft. (2016). Azure Functions – Serverless Apps and Computing. Microsoft Azure: <https://azure.microsoft.com/en-us/services/functions/>.
8. GitHub. (2009). GitHub Pages. GitHub: <https://pages.github.com/>.
9. GitHub. (2018). Features. GitHub Actions. GitHub: <https://github.com/features/actions>.
10. DSTU. (2015). DSTU 8302:2015. Informatsiia ta dokumentatsiia. Bibliohrafichne posylannia [Information and documentation. Bibliographic reference] [in Ukrainian].
11. American Psychological Association. (2020). *Publication Manual of the American Psychological Association, Seventh Edition*. American Psychological Association.
12. Ukrainian National Commission on Orthography. (2019). *Ukrainskyi pravopys* [Ukrainian orthography]. [in Ukrainian].
13. ISO. (2016). ISO/IEC 29500-1:2016. Information technology – Document description and processing languages – Office Open XML File Formats – Part 1: Fundamentals and Markup Language Reference.