ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.7:378.091 https://doi.org/10.35546/kntu2078-4481.2022.4.5

С. В. БЛИЗНЮК
ПВНЗ «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука»
ORCID: 0000-0002-4363-3524
О. В. ШПОРТЬКО
ПВНЗ «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука»
ORCID: 0000-0002-4013-3057

ВІДСЛІДКОВУВАННЯ ПОМИЛОК ПІД ЧАС СУПРОВОДЖЕННЯ САЙТУ УНІВЕРСИТЕТУ В ОДНІЙ З БАГТРЕКІНГОВИХ СИСТЕМ

На початку роботи наголошено на тому, що відстеження помилок має величезне значення, як частина супроводу і обслуговування програмного забезпечення. Коротко перелічено та дано характеристику існуючих на ринку систем багтрекнгу. Названо переваги та недоліки таких систем, як Mantis Bug Tracker (MantisBT), Jira, YouTrack. YouTrack пропонований JetBrains як хмарний або автономний сервер. Є можливість управління проектами, користувачами, групами і ролями. JetBrains надає свій YouTrack для безкоштовного використання розробникам відкритих проектів і для навчання. YouTrack доступний також у вигляді сервісу (SaaS). Обмеження безкоштовної версії YouTrack: не більше десяти користувачів; хмарна версія має обмеження на об'єм зберігання даних. Jira можна створювати проекти через e-mail. Існує можливість імпорту звітів в Excel, а також можливість Wiki-форматування. Підтримує інтеграцію з Confluence. Даний багтрекер здатний працювати через захищене з'єднання із застосуванням SSL.

Зокрема, для Mantis Bug Tracker сказано, що система має гнучкі можливості конфігурування, що дозволяє настроювати її не тільки для роботи над програмними продуктами, але і в якості системи обліку заявок для технічної підтримки. Вона надає наступні можливості: простота в установці; немає обмежень на кількість користувачів, проблем або проектів; прийнятна швидкість роботи; підтримує основні платформи ОС; підтримує декілька СУБД; високий рівень настройки; інтуїтивно зрозумілий огляд помилок; надсилання повідомлень електронною поштою; відстеження часу; модулі, які значно покращують використання MantisBT; експорт в CSV, Microsoft Excel, Microsoft Word; інтеграція управління вихідним кодом (GIT, SVN і CVS). До недоліків системи можна віднести: якість інтерфейсу; відсутність можливості генерування звіту про виконану роботу та можливості автоматизації, а також те, що у процесі створення звіту про помилку до нього можна прикріпити тільки один знімок екрану.

За допомогою Mantis Bug Tracker проведено відстежування помилок системою Mantis Bug Tracker на сайті університету МЕГУ та зроблено рекомендації з усунення виявлених помилок. В роботі приведено також фрагмент коду згідно якого відбувалось відслідковування помилок сайту.

Ключові слова: *автоматизоване тестування, Mantis Bug Tracker, багтрекінг, інтерфейс, помилки сайту.*

S. V. BLYZNIUK
Academician Stepan Demianchuk International University of Economics and Humanities
ORCID: 0000-0002-4363-3524
O. V. SHPORTKO
Academician Stepan Demianchuk International University of Economics and Humanities
ORCID: 0000-0002-4013-3057

TRACKING ERRORS DURING THE MAINTENANCE OF THE UNIVERSITY WEBSITE IN ONE OF THE BAG TRACKING SYSTEMS

At the beginning of the work, it is emphasized that error tracking is of great importance as part of software support and maintenance. The characteristics of existing bug tracking systems on the market are briefly listed and given. Advantages and disadvantages of such systems as Mantis Bug Tracker (MantisBT), Jira, YouTrack are named. YouTrack is offered by JetBrains as a cloud or stand-alone server. It is possible to manage projects, users, groups and roles. JetBrains provides its YouTrack for free use by developers of open source projects and for training. YouTrack is also available as a service (SaaS). Limitations of the free version of YouTrack: no more than ten users; the cloud version has data storage limits. Jira projects can be created via e-mail. There is the possibility of importing reports into Excel, as well as the possibility of Wiki-formatting. It supports integration with Confluence. This bug tracker is able to work through a secure connection using SSL.

In particular, for Mantis Bug Tracker, it is said that the system has flexible configuration options, which allows you to set it up not only for work on software products, but also as a system for recording requests for technical support. It provides the following features: ease of installation; there are no limits on the number of users, issues or projects; acceptable work speed; supports major OS platforms; supports several DBMS; high level of customization; intuitive error overview; sending messages by email; time tracking; modules that significantly improve the use of MantisBT; export to CSV, Microsoft Excel, Microsoft Word; integration of source code management (GIT, SVN and CVS). Disadvantages of the system include: interface quality; the lack of the ability to generate a report of the work performed and the possibility of automation, and the fact that in the process of creating a bug report, only one screenshot can be attached to it.

With the help of Mantis Bug Tracker, errors were tracked by the Mantis Bug Tracker system on the website of the MEG University and recommendations were made to eliminate the detected errors. The work also includes a fragment of the code according to which site errors were tracked.

***Key words**: automated testing, Mantis Bug Tracker, bug tracking, interface, site errors.*

Постановка проблеми

В наш час, коли програмні продукти стали надзвичайно великими, для задоволення різноманітних потреб користувачів, використання таких багтрекінгових систем особливо актуальні. Діяльність, по супроводу в життєвому циклі розробки програмного забезпечення, використовує велику частину бюджету проекту, пов'язаних з витратами на розробку програмного забезпечення. Отже, відстеження помилок має величезне значення, як частина супроводу і обслуговування програмного забезпечення. Вибір системи відслідковування помилок безпосередньо впливає на ефективність процесу розробки [2].

Аналіз останніх досліджень і публікацій

Розгляд комплексу питань, пов'язаних з використанням сучасних інформаційно-комунікаційних технологій в освітньому процесі в середній і вищій школі, започатковано в роботах Р. Вільямса, К. Макліна, А. П. Єршова, М. І. Жалдака, Е. І. Кузнєцова, О. А. Кузнєцова, В. М. Монахова. Значні за обсягом дослідження з відслідковування помилок під час супроводження сайту університету в одній з багтрекінгових систем проведені такими вченими, як О. М. Алєксєєв, Т. А. Вакалюк, О. Г. Глазунова, О. Г. Колгатін, К. Р. Колос, С. Г. Литвинова, В. В. Осадчий.

Мета роботи полягає у відслідковуванні помилок системою Mantis Bug Tracker на сайті університету.

Виклад основного матеріалу дослідження

Багтрекінгові системи (БТС) або системи відстеження помилок представляють собою програмні продукти, які дозволяють реєструвати і відслідковувати хід вирішення кожної помилки (бага), виявленої тестувальником, до тих пір, поки проблема не буде вирішена. На даний момент можна назвати такі багтрекінгові системи, як BugZilla, Redmine, PivotalTracker, Trello, GitLab, Jira, Trac, Mantis, Airbrake. io, Backlog, ReQtest, BugHerd, FogBugz, Lighthouse та інші. Наведемо коротку характеристику найбільш поширеним системам.

Mantis Bug Tracker (MantisBT) – це найпоширеніший представник систем стеження за багами. Він написаний на мові PHP. Його не можна назвати ідеальним багтрекером, однак він здатний вирішувати всі основні завдання, які від нього потрібні. Тестувальник програмного забезпечення змушений працювати в даному багтрекер безпосередньо за допомогою браузера. Користувачі даного продукту постійно нарікають на проблеми з Unicode. Загалом, даний продукт ще вимагає особливої доопрацювання. Система має гнучкі можливості конфігурування, що дозволяє настроювати її не тільки для роботи над програмними продуктами, але і в якості системи обліку заявок для технічної підтримки. Вона надає наступні можливості: простота в установці; немає обмежень на кількість користувачів, проблем або проектів; прийнятна швидкість роботи; підтримує основні платформи ОС; підтримує декілька СУБД; високий рівень налаштувань; інтуїтивно зрозумілий огляд помилок; надсилання повідомлень електронною поштою; відстеження часу; модулі, які значно покращують використання MantisBT; експорт в CSV, Microsoft Excel, Microsoft Word; інтеграція управління вихідним кодом (GIT, SVN і CVS).

До недоліків системи можна віднести: якість інтерфейсу; відсутність можливості генерування звіту про виконану роботу та можливості автоматизації, а також те, що у процесі створення звіту про помилку до нього можна прикріпити тільки один знімок екрану [5].

CodeBeamer – це не просто проста система відстеження проблем, але це платформа розвитку співпраці з інтегрованим управлінням життєвим циклом додатків. Під цим заголовком ми можемо представити набір таких служб, як управління документами, пов'язаний з проектом, Wiki, Форум, Інтернет-чат і, звичайно, відстеження випусків, інтегрований з контролем версій (SVN). Він доступний безкоштовно для студентів та оцінювачів; однак для використання в бізнесі це комерційно. Як приклад публічного використання, ми 36 можемо назвати, наприклад, спільноту JavaForge. com, яка розміщує декілька проектів з відкритим кодом Java та працює над CodeBeamer. CodeBeamer – програма Java EE, що працює на контейнері сервлетів Apache Tomcat. Він поставляється у вигляді двійкового інсталяційного файлу або для Windows, або для Linux. Файли встановлення Tomcat включені в пакет; однак, немає жодної проблеми встановити CodeBeamer у свій власний екземпляр Tomcat. Закінчивши установку, потрібно встановити підключення до бази даних (та деякої іншої конфігурації) у файлі властивостей. Здається, інтерфейс користувача має кращу структуру, ніж інтерфейс Bugzilla; однак це все одно не виграє ціну за зручність

використання. Як ви бачите нижче у сценарії використання, для додавання нових запитів на випуск потрібно досить глибоко скористатися інтерфейсом. Це допоможе перенести деякі найбільш використовувані функції на деякі додаткові панелі чи інші подібні реорганізації. Під час користування інтерфейсом CodeBeamer інтерфейс нічого не знайшов, лише багато речей передбачали занадто багато кроків. CodeBeamer забезпечує робочий процес за замовчуванням, який можна налаштувати в інтерфейсі користувача.

Jira. Багтрекер написаний на мові Java. Відображає хід виконання проектів, є зручні посилання, за допомогою яких можна контролювати звіти і поточні завдання. За допомогою даної системи можна створювати проекти через e-mail. Існує можливість імпорту звітів в Excel, а також можливість Wiki-форматування. Підтримує інтеграцію з Confluence. Даний багтрекер здатний працювати через захищене з'єднання із застосуванням SSL. Потенційних клієнтів, однак, може відлякувати вартість комерційної ліцензії. Таким чином, основними перевагами системи є: високий рівень налаштування; простий і зручний інтерфейс; крос-браузерний і багатоплатформний інструмент; проста і глибока інтеграція з іншими популярними інструментами; унікальні функції, доступні тільки в JIRA; хмарне сховище; набір функцій для гнучкого тестування; широкий асортимент доповнень; відстеження роботи персоналу; високий рівень безпеки [3].

До недоліків системи можна віднести: складний процес налаштування; значні витрати часу, щоб навчитися ефективно використовувати JIRA; ціну; складність для невеликих команд.

Bugzilla дозволяє адміністраторам створювати та змінювати так звані «дерева класифікації» продуктів. Це означає, що немає жодної проблеми створити дерево, де, наприклад, на першому рівні були б назви проектів, а на другому рівні – назви продуктів, що редагуються. Ніяких спеціальних злому не потрібно. Bugzilla має дуже просунуту систему звітування, яку на перший погляд досить важко почати використовувати – користувальницький інтерфейс не дуже дружній. Інтеграція Bugzilla із системами управління вихідним кодом забезпечується за допомогою плагінів. Офіційно підтримуються CVS, Subversion, Bonsai, Teamtrack Performance та Tinderbox. Ці та інші можна знайти на сторінці ресурсів додатків. До Buggilla можна отримати доступ та змінити через інтерфейс вебслужб XML-RPC. Це дає можливість легко писати зовнішні інструменти, які взаємодіють з Bugzilla.

Bugzilla повідомляє користувачів про будь-які нові або оновлені помилки електронною поштою. Bugzilla підтримує базове відстеження часу. Bugzilla також підтримує систему голосування, в якій користувачі можуть голосувати за питання або функції, які хочуть бачити реалізованим.

YouTrack – це добре відомий інструмент відстеження помилок, пропонований JetBrains як хмарний або автономний сервер. Це продукт, орієнтований як на окремих програмістів, так і на команду розробників. Високий рівень локалізації зробив його популярним у всьому світі. Багато користувачів відзначає можливість швидкого налаштування системи відслідковування багів під себе та можливість інтеграції з іншими сервісами. Є можливість управління проектами, користувачами, групами і ролями. JetBrains надає свій YouTrack для безкоштовного використання розробникам відкритих проектів і для навчання. YouTrack доступний також у вигляді сервісу (SaaS). Обмеження безкоштовної версії YouTrack: не більше десяти користувачів; хмарна версія має обмеження на об'єм зберігання даних [1].

Таким чином, система відстеження помилок сприяє економії часу всіх учасників процесу розробки. Багтрекер в простій і зрозумілій формі надає всі виявлені невідповідності та помилки. Наявність багтрекера є дуже важливим компонентом у розробці програмного забезпечення, вони широко застосовуються компаніями, що розробляють програмні продукти. Загалом, використання багтрекера є однією з «ознак хорошої команди програмістів».

Основні етапи тестування сайту МЕГУ імені академіка Степана Дем'янчука можна розділити на: – аналіз продукту; – робота з вимогами; – розробка стратегії тестування; – створення тестової документації; – тестування прототипу; – основне тестування; – стабілізація; – експлуатація [4].

На рисунку 1 схематично представлено роботу Mantis Bug Tracker по алгоритму, який відображає виконувані дії під час тестування та аналіз веб-сайту. Кроки перевірки є незалежними один від одного тому можуть виконуватись в іншому порядку, якщо буде така потреба у програміста.

Для запуску Mantis Bug Tracker потрібно перейти в консолі до папки з алгоритмом та виконати наступну команду:
npm run babel-node src / index. js.

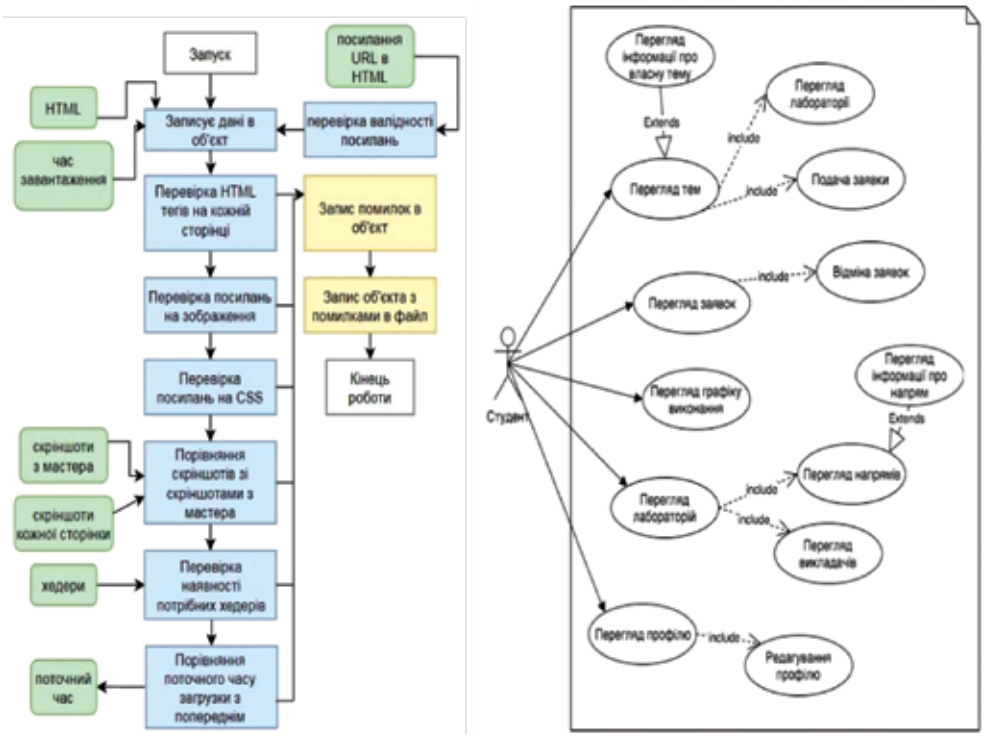
Ця команда запускає роботу Mantis Bug Tracker і вона запускає скрипт з назвою index. js. З файлу index. js завантажується app – це головний клас який контролює роботу всього алгоритму та виконує крок за кроком всі потрібні дії.

export default class App {
constructor() {
this. shared = new Shared();

Список класів, які завантажують потрібні дані.

this. preloaders = [LinksPreloader,];

Список класів, які перевіряють поведінку веб-сайту. Ці класи включають: перевірку HTML, посилань на зображення та CSS, створюють та порівнюють скріншоти, роблять перевірку хедерів та затримку кожної зі сторінок.



а) б)

Рис. 1. а) Діаграма прецедентів веб-сайту МЕГУ та б) схема роботи алгоритму

Джерело: Результати власних досліджень.

this. cases = [
HtmlCase,
ImgCase,
CssCase,
PuppeteerCase,
HeadersCase,
DelayCase,];}
run = async () => {

Наступний код запускає всі прелоадери Mantis Bug Tracker – це класи, які завантажують дані попередньо, які потрібні для роботи.

await this. runPreload();
await this. runCases();
await this. writeToFile();}

Після запуску Mantis Bug Tracker алгоритм обробляє конфігураційний файл та отримує маршрут за яким потрібно брати вхідні дані для тестування. Нижче наведений цей файл:

export default {
url: 'http: // localhost: 8001',
link_depth: 3,
time_max_delay_multiplier: 3,};

Далі за посиланням на веб-ресурс алгоритм завантажує код HTML сторінки та посилання на інші сторінки. Далі він за кожним з посилань рекурсивно переходить збирає ті ж дані, які були зібрані для першої сторінки і так рекурсивно до параметра вказаного в конфігурації. Також для кожної зі сторінок вимірюється час її завантаження.

const timeStart = Date. now();
const response = await fetch (getFullUrl (url));

Перевіряє чи статус відповіді від сторінки не є помилковим та отримуємо код HTML у вигляді тексту.

if (response. status === 200) {
const text = await response. text();

Знаходимо всі посилання на сторінці за допомогою regex – коду. Цей код широко використовуються для пошуку інформації у тексті за заданими параметрами.

const urls = findLinks (text, /< 2 && url !== baseUrl) {
this. shared. pages [url] = {
links: urls,
html: text,
Наступний код отримує час за який було завантажено веб-сторінку.
time: Date. now() – timeStart,};}
await Promise. all(
urls. map (async childUrl => {
Даний код робить рекурсивну перевірку кожного з посилань – перевіряє доступність посилання, знаходить інші посилання в HTML та вимірює час завантаження.
await this. checkUrl (childUrl, url, depth + 1);})) }
Наступним кроком є перевірка тегів на кожній HTML сторінці сайту МЕГУ. Під час перевірки, якщо виникає помилка вона записується у відповідний об'єкт з помилками. Для перевірки беруться такі теги:
– div
– p
– h1, h2, h3, h4, h5, h6
– head
– body
– header
– article
Згідно з документацією HTML наведені вище теги повинні бути обов'язково закриті.
tags. forEach (tag => {
const openTagLength = html. split('<' + tag + '>'). length – 1;
const closingTagLength = html. split(' 0) { this. addError('Знайдено ' + notClosedSize + ' не закритих тегів ' + tag, SEVERITY_MEDIUM, url); }});
Якщо не має тегу! DOCTYPE, записуємо помилку, що цей тег відсутній.
if (!html. includes('')) { this. addError('Doctype
Наступник кроком відбувається перевірка посилань в HTML коді на файли CSS. За допомогою regex знаходимо та записуємо в масив links всі посилання на файли. Наступний код створює зображення для кожної зі сторінок та порівнює їх із зображеннями створеними попередньо. Наступна функція завантажує кожну зі сторінок та робить скріншот на цій сторінці. Задля збільшення швидкості кожна зі сторінок відкривається у окремій вкладці. Наступний код звіряє очікуване значення хедера з отриманим та записує помилку, якщо вони різні. Останньою перевіркою алгоритму є перевірка кількості часу завантаження сторінки.

Аналізуючи отримані результати роботи Mantis Bug Tracker по алгоритму автоматизованого тестування, була створена таблиця 1, в якій описано кожну з груп помилок та кількість помилок для відповідної групи.

Таблиця 1

Аналіз результатів тестування сайту МЕГУ системою Mantis Bug Tracker

Назва групи помилок	Кількість помилок	Середній пріоритет групи	Кроки для усунення помилок
Посилання на сторінку недоступне	12	3	Потрібно додати відповідні методи до контролерів, чи перевірити наявність інформації в базі даних.
Посилання на файл CSS недоступне	1	2	Потрібно додати відповідні файли до проекту або вилучиш посилання на файли CSS.
Посилання на зображення недоступне	7	2	Потрібно додати зображення в відповідне місце або змінити маршрут посилання.
HTML структура не валідна	2	2	Потрібно додати відповідні теги, які відсутні.
Зміна зовнішнього вигляду сайту	2	3	Потрібно змінити стилі та HTML до відповідних або нові зміни записати до майстра.
Відсутність потрібних хедерів	2	3	Потрібно додати необхідні хедери безпеки.
Час завантаження сторінки перебільшує заданий час	0	0	Потрібно перевірити, які зміни призвели до погіршення часу завантаження сторінки.

Джерело: Результати власних досліджень.

Висновки

Встановлено, що відстеження помилок має величезне значення, як частина супроводу і обслуговування програмного забезпечення. Коротко перелічено та дано характеристику існуючих на ринку систем багтрекнгу. Названо переваги та недоліки таких систем, як Mantis Bug Tracker (MantisBT), Jira, YouTrack. Зокрема, для Mantis

Bug Tracker сказано, що система має гнучкі можливості конфігурування, що дозволяє настроювати її не тільки для роботи над програмними продуктами, але і в якості системи обліку заявок для технічної підтримки. За допомогою Mantis Bug Tracker проведено відстежування помилок системою Mantis Bug Tracker на сайті університету МЕГ та зроблено рекомендації з усунення виявлених помилок. В роботі приведено також фрагмент коду згідно якого відбувалось відслідковування помилок сайту.

Список використаної літератури

1. Гриценко В. Г., Подолян О. М. Теоретичні основи проектування і створення інформаційно-аналітичних систем управління навчальним закладом. *Педагогіка вищої та середньої школи*. 2014. Вип. 40. С. 166–173.

2. Киричек Г. Г., Киричек О. О. Модель оцінки плагіату програмного коду на основі системи контролю версій. *Східно-Європейський журнал передових технологій.* 2012. № 2/2. Вип. 56. С. 25–28.

3. Луценко Є. С. Багтрекер, як інструмент контролю за процесом тестування. *Збірник наукових праць студентів спеціальностей «Інформаційні управляючі системи і технології», «Комп'ютерний еколого-економічний моніторинг»* / редкол.: В. С. Пономаренко [ті ін.]. Харків: ХНЕУ, 2011. 308 с.

4. Шпортько О. В., Гаврилюк В. І. Сучасні багтрекерні системи відслідковування помилок: переваги та недоліки. *Сучасні тендеції в математичному моделюванні і його програмному забезпеченні*. Рівне, 2020. С. 54–56.

5. Tryus Yu., Stetsenko I., Herasymenko I., Grytsenko V. Information-analytical learning management system universities. *Informational Technologies in Education*. 2016. № 29. P. 15–30.

References

1. Hrytsenko V. H., Podolian O. M. (2014). Teoretychni osnovy proektuvannia i stvorennia informatsijno-analitychnykh system upravlinnia navchal'nym zakladom. *Pedahohika vyschoi ta seredn'oi shkoly*. Vyp. 40. S. 166–173.

2. Kyrychek H. H., Kyrychek O. O. (2012). Model' otsinky plahiatu prohramnoho kodu na osnovi systemy kontroliu versij. *Skhidno-Yevropejs'kyj zhurnal peredovykh tekhnolohij.* № 2 / 2. Vyp. 56. S. 25–28.

3. Lutsenko Ye. S. (2011). Bahtreker, iak instrument kontroliu za protsesom testuvannia. *Zbirnyk naukovykh prats' studentiv spetsial'nostej «Informatsijni upravliaiuchi systemy i tekhnolohii», «Komp'iuternyj ekoloho-ekonomichnyj monitorynh*» / redkol.: V. S. Ponomarenko [ti in.]. Kharkiv: KhNEU. 308 s.

4. Shport'ko O. V., Havryliuk V. I. (2020). Suchasni bahtrekerni systemy vidslidkovuvannia pomylok: perevahy ta nedoliky. *Suchasni tendetsii v matematychnomu modeliuvanni i joho prohramnomu zabezpechenni*. Rivne. S. 54–56.

5. Tryus Yu., Stetsenko I., Herasymenko I., Grytsenko V. (2016). Information-analytical learning management system universities. *Informational Technologies in Education.* № 29. P. 15–30.