УДК 004.9 http://doi.org/10.15407/pp2024.02-03.426

В.І. Шинкаренко, О.О. Жеваго

ВІДЕОВІЗУАЛІЗАЦІЯ ПРОЦЕСУ НАЛАГОДЖЕННЯ

Одним із сучасних трендів в освіті є мікронавчання, яке передбачає використання коротких відеороликів у навчальному процесі. Мікронавчання має низку переваг, серед яких те, що цей підхід більш орієнтований на студента, спрямований на підвищення рівня засвоєння знань, вимагає менше часу на навчання, дає можливість навчатися в будь-який час і в будь-якому місці. У попередніх роботах авторами розроблена конструктивно-продукційна модель і відповідне програмне забезпечення, щодо відстеження дій програміста під час підготовки тексту програми та налагодження в середовищі розробки Visual Studio. У даній роботі представлений розвиток цих програмних засобів. На основі зібраної інформації у файлах логів щодо процесів налагодження програм виконується їхня візуалізація, яка відтворює послідовність дій під час оригінального процесу налагодження. Метою є підвищення ефективності та результативності навчання програмуванню. Відеовізуалізація демонструє роботу програміста з підготовки та коригування коду під час налагодження, та синхронізована з часовими мітками у файлах логів лише у періоди активності. Виконується накладання коментарів на відео, де надаються пояснення та пропозиції щодо покращення процесу налагодження. Коментарі допомагають зрозуміти обґрунтування конкретних дій, виконаних під час налагодження, та надають вказівки щодо покращення процесів або використання альтернативних підходів. Користь візуалізації для викладача полягає у можливості: аналізу процесу налагодження конкретного студента, виявленні типових помилок конкретної групи студентів, коригування процесу навчання та допомоги в удосконаленні навичок налагодження. Користь для студента: можливість аналізу своєї роботи, розвиток критичного мислення щодо її удосконалення, можливість отримання своєчасної допомоги від викладача.
Ключові слова: налагодження, візуалізація, відео, навчання, інженерія програмного забезпечення, конструктивно-продукційне моделювання.

V.I. Shynkarenko, O.O. Zhevaho

VIDEO-BASED VISUALIZATION OF DEBUGGING PROCESS

One of the current trends in education is microlearning, which involves the use of short videos in the learning process. Microlearning has a number of advantages, including the fact that this approach is more student-centered, aims to increase the level of knowledge assimilation, requires less time for learning, and allows learning anytime and anywhere. In previous works, the authors have developed a constructive-production model and corresponding software for tracking programmer's actions during the preparation of program text and debugging in the Visual Studio development environment. This article presents an extension of these software tools. Based on the collected information in the log files about the program debugging processes, their visualization is performed, which reproduces the sequence of actions during the original debugging process. The goal is to increase the efficiency and effectiveness of programming education. The video-based visualization demonstrates the programmer's work on preparing and correcting the code during debugging and is synchronized with the time stamps in the log files only during periods of activity. Comments are overlaid on the video, providing explanations and suggestions for improving the debugging process. Comments help to understand the rationale for specific actions taken during debugging and provide guidance on how to improve processes or use alternative approaches. The benefit of visualization for the teacher is the ability to: analyze the debugging process of a particular student, identify typical mistakes of a particular group of students, adjust the teaching process accordingly, and provide targeted assistance in improving debugging skills. Benefits for the student: the ability to analyze your own work, develop critical thinking on how to improve it, and receive timely assistance from the teacher.
Key words: debugging, visualization, video, education, software engineering, constructive-synthesizing modeling.

Вступ

Систематично перевіряти програми на наявність помилок, знаходити та виправляти їх – основна компетенція розробників програмного забезпечення. Професійні програмісти витрачають від 20% до 40% свого часу на налагодження [1]. Цей пока-

зник набагато вищий у початківців, оскільки вони роблять більше помилок під час написання коду і відповідно повинні частіше налагоджувати свій код.

Навчанню налагодження часто не приділяється достатньо часу, навіть інколи його не включено до курсів програмування і, як наслідок, студенти недостатньо знайомі з методами, концепціями та стратегіями налагодження, які потенційно могли б покращити їхні навички налагодження.

Наприклад, програмісти-початківці зазвичай переглядають проблемні програми без будь-якого плану. Вони схильні вносити зміни випадковим чином, щоб зменшити різницю між помилковою та цільовою програмами, під час яких іноді забувають повернути неправильні модифікації та вносять нові помилки [2].

Початківцям важко отримати відмінні навички налагодження, просто навчившись писати програми. Тому новачкам слід надавати чіткі навчальні інструкції для вивчення ефективних стратегій налагодження. Навчання налагодження повинно відбуватися на ранніх етапах, щоб новачки не намагалися налагодити свої програми за допомогою імпровізованих і безсистемних методів.

Хоча різні моделі налагодження можуть містити різні підзадачі налагодження, їх можна згрупувати у чотири основні: ідентифікація проблеми (порівнюючи очікувані та отримані результати роботи програми), локалізація помилки, виправлення помилки та перевірка рішення.

Ці чотири підзадачі є фактично фундаментальними кроками, з якими розробники зазвичай стикаються під час налагодження. Використання систематичного поетапного підходу до налагодження покращить навички налагодження, збільшивши швидкість і правильність налагодження та зменшивши кількість введених помилок [2].

Багато студентів потребують допомоги в налагодженні, щоб досягти прогресу, коли вони вчяться писати програми. Особиста взаємодія з окремими студентами для надання зворотного зв'язку щодо їхніх програм, хоча і є безумовно ефективною для полегшення їхнього навчання, стає все складнішою у зв'язку зі збільшенням кількості студентів та дистанційною формою навчання. Викладачі не в змозі працювати цілодобово, щоб надавати допомогу студентам, коли кожен навчається у своєму власному темпі.

Онлайн-навчання суттєво змінило спосіб викладання, що призвело до використання нових технологій. Відео стає одним з інструментів, які викладачі використовують для представлення навчального контенту своїм студентам. Один зі способів використання відео – інтегрувати його в лекцію, щоб візуально проілюструвати концепції, які інакше було б складно пояснити за допомогою зображень і тексту.

Існує практична потреба у наданні допомоги студентам з налагодження, щоб навчання студентів програмуванню не було ускладнене через їхню слабкість у пошуку помилок у програмах.

Пропонується новий підхід, інструмент, який допоможе покращити викладання на початкових курсах програмування, надаючи студентам можливість отримувати поради щодо налагодження, а викладачам бачити індивідуальні траєкторії навчання студентів.

Основна ідея підходу полягає у створенні комплексної відеовізуалізації процесу налагодження, використовуючи журнали процесів налагодження та доповнюючи їх візуальними підказками. Коментарі слугують цінними анотаціями, які допомагають зрозуміти обґрунтування конкретних дій, виконаних під час налагодження, та надають вказівки щодо потенційних покращень процесу або альтернативних підходів.

Аналіз останніх досліджень і публікацій

Навчання налагодження. Попри важливість навчання налагодженню, існує напрочуд мало досліджень, присвячених саме навчанню стратегій налагодження та розвитку навичок налагодження.

Навчання систематичного процесу налагодження було оцінено в декількох роботах [2, 3]. Багато студентів не знайомі з процесом налагодження та стратегіями налагодження взагалі, і, як наслідок, застосовують метод спроб і помилок. Навчання студентів ефективному, систематичному процесу налагодження та іншим стратегіям налагодження під час навчання програмування вирішило б цю проблему. В цих дослідженнях розробили систематичну процедуру налагодження, яка ґрунтувалася на "науковому методі": на основі спостережуваної поведінки програми формулюються гіпотези, які перевіряються в експериментах і, за необхідності, уточнюються доти, доки не буде знайдено причину. Результати досліджень показали, що експериментальна група досягла значно вищих результатів у тестах із налагодження, а, отже, навчання студентів стратегіям налагодження є корисним і дійсно необхідним.

Вказані дослідження надають емпіричні аргументи на користь викладання стратегій та методів налагодження, і доводять, що процес навчання програмуванню може бути полегшений ефективним процесом налагодження.

Зворотний зв'язок та візуалізація. Своєчасний зворотний зв'язок важливий для того, щоб спрямовувати студентів під час самостійної роботи над вправами з програмування. Оскільки стає дедалі нереалістичніше очікувати на індивідуальну взаємодію викладачів зі студентами, дослідники створюють різноманітні автоматизовані системи для полегшення навчання та надання своєчасного зворотного зв'язку.

У минулому було проведено багато досліджень щодо надання автоматизованої допомоги студентам у вивченні написання програм, але є відносно мало робіт щодо надання автоматизованої підтримки з налагодження для студентів.

У процесі автоматичного створення зворотного зв'язку для завдань із програмування ключовою проблемою є розробка підказок, які є настільки ж ефективними, як і зворотний зв'язок, що надається викладачами особисто.

Однією із систем візуалізації програм для надання інтерактивних персоналізованих підказок є TraceDiff [4]. TraceDiff це інструмент, який порівнює динамічне виконання некоректного та коректного коду і показує, як помилка призводить до різниці в поведінці, й де трасування некоректного коду відхиляється від очікуваного рішення.

Інший підхід до візуалізації процесу розробки програмного забезпечення - використання засобів розширення середовищ розробки [5].

Окрім систем візуалізації, деякі дослідники створили інтерактивні середовища розробки, пристосовані для навчання студентів на вступних курсах програмування [6].

Використання відео як варіант документації у поєднанні з додатковою інформацією застосовувався для тестування графічного інтерфейсу [7], коли записують відео під час виконання тестів для доповнення тестових звітів.

Однією з небагатьох систем надання практичних автоматизованих порад з налагодження для підтримки навчання студентів є Virtual Debugging Advisor (ViDA). Система заснована на розпізнаванні поширених помилок у програмах студентів [8]. Результати дослідження вказують на те, що з увімкненим ViDA більший відсоток студентів змогли виправити власні помилки, а переважна більшість респондентів вважають ViDA корисним для навчання програмуванню.

Попри значний прогрес у розробці автоматизованих систем для навчання програмуванню та надання зворотного зв'язку студентам, досліджень у сфері надання автоматизованої підтримки з налагодження для студентів є відносно мало. Це вказує на те, що наявні роботи зосереджені переважно на навчанні написання коду, а не на важливому етапі налагодження програмного забезпечення. Таким чином, більша увага до створення і вдосконалення автоматизованих систем, спрямованих на підтримку налагодження програм, є важливим завданням для подальшого розвитку освіти в галузі програмування.

Мета та завдання дослідження

Метою даного дослідження є розробка інструментів для автоматизованого створення відеовізуалізації процесу налагодження, використовуючи наявні журнали процесів налагодження та доповнюючи їх візуальними підказками. За допомогою цього вдасться вдосконалити процес навчання студентів налагодженню.

Для досягнення мети були поставлені такі завдання:

– розробити інструмент для візуалізації процесу налагодження, який наочно демонструє послідовність дій на основі журналів подій, отриманих із середовища розробки за роботою [9];

– доповнити відео коментарями, які надають пояснення та пропозиції щодо покращення процесу налагодження;

– перевірити розроблений інструмент на журналах, зібраних під час експерименту представленого у [10].

Відеовізуалізація процесу налагодження

У попередніх роботах, використовуючи методологію конструктивно-продукційного моделювання, розроблено конструктор, призначений для створення файлу журналу налагоджувальної діяльності для формалізації процесу збору даних про використання середовища розробки під час налагодження. На основі конструктивної моделі розроблено розширення до середовища розробки Microsoft Visual Studio, в якому всі налагоджувальні дії разом із контекстом фіксуються в журналі подій [9].

У даній роботі ми представляємо розвиток цих програмних засобів. На основі розроблених моделей і засобів створюється система для візуалізація зібраної інформації, яка відтворює послідовність дій під час процесу налагодження.

Для візуалізації процесу налагодження розроблено програмне забезпечення, яке складається з наступних компонентів:

1. WebUI – надає веб-інтерфейс для доступу до функцій системи.

2. LogAPI – містить RESTful API для приймання журналів подій із середовища розробки.

3. DataManagement – надає CRUD операції для роботи з файлами журналів логів та відео.

4. LogDataProcessor – відповідає за обробку журналів подій, отриманих із середовища розробки. Журнали слугують основою для реконструкції сеансів налагодження і надання рекомендацій.

5. CommentGenerator – генерує коментарі, які містять пояснення подій з файлу журналу та рекомендації щодо налагоджувальних дій. Ці коментарі базуються на аналізі процесу налагодження для визначення ефективних стратегій і підходів, що підвищує освітню цінність відео.

6. VideoGenerator – використовуючи оброблені дані журналу та коментарі, генерує відеопредставлення процесу налагодження. Це відео фіксує взаємодію користувача з кодом, надає пояснення та рекомендації щодо налагоджувальних дій і синхронізується з часовими мітками з даних журналу. Періоди без активних дій пропускаються, щоб зробити відео коротшим і більш інформативним. Ці моменти відображаються в коментарях.

Розроблений інструмент є програмним засобом, спрямованим на автоматизоване створення відеовізуалізації процесу налагодження програмного забезпечення. Він працює на основі журналів подій (файлів логів), отриманих із середовища розробки, які генеруються в режимі реального часу, коли розробник працює з текстом програми. Кожна дія записується разом з міткою часу та контекстом, забезпечуючи хронологічний запис процесу налагодження. Для фіксування усіх подій разом з контекстною інформацією в середовищі розробки використовується інструмент, представлений в [9].

Використовуючи наявні журнали та доповнюючи їх візуальними підказками, наш підхід пропонує цілісний погляд на процес налагодження, прояснюючи послідовність дій та стратегій, які застосовують розробники.

Відеопредставлення дій з налагодження є цінним ресурсом як для викладa-

чів, так і для студентів. Воно дозволяє студентам спостерігати ефективні стратегії налагодження в дії, допомагаючи розвивати основні навички налагодження. До відеоролика додані підказки та рекомендації, які надають користувачу додаткові пояснення щодо кожної дії в процесі налагодження.

Перелічені функції та особливості розробленого інструменту спрямовані на підвищення ефективності навчання налагодження програмного забезпечення. Візуальна форма представлення інформації допомагає студентам краще зрозуміти процес налагодження та швидше опановувати необхідні навички.

Інтерфейс інструменту візуалізації налагодження (рис. 1) призначений для надання користувачам комплексного уявлення про процес налагодження, інтегруючи: загальну інформацію про сеанси налагодження (GeneralInfoSection), відображення коду (CodeSection), та коментарі з рекомендаціями (CommentsSection).

Секція GeneralInfoSection відображення загальної інформації містить: номер поточної сесії налагодження (у прикладі перша), загальну кількість сесій (у прикладі чотири), а також тривалість поточної сесії (у прикладі 30 с).

Секція CodeSection відображення коду займає помітну частину інтерфейсу, показуючи вихідний код, що налагоджується і надає декілька функцій:

– підсвічування синтаксису для покращення читабельності;

– індикація точок зупинки за допомогою візуальних маркерів, у двох режимах: активна (зелена) і не активна (червона);

– підсвічування поточного виконуваного рядка.

Секція CommentsSection з коментарями та рекомендаціями розташована поряд з відображенням коду і надає інформацію, яка пояснює дії, виконані під час налагодження, та вказівки щодо альтернативних рішень. Кожен коментар відображається з урахуванням часу відносно міток часу у файлі журналу і тривалості сесії, що дозволяє візуально відстежувати хід процесу налагодження. Рекомендації та підказки виділяються іншим кольором, щоб відрізнити їх від звичайних коментарів.

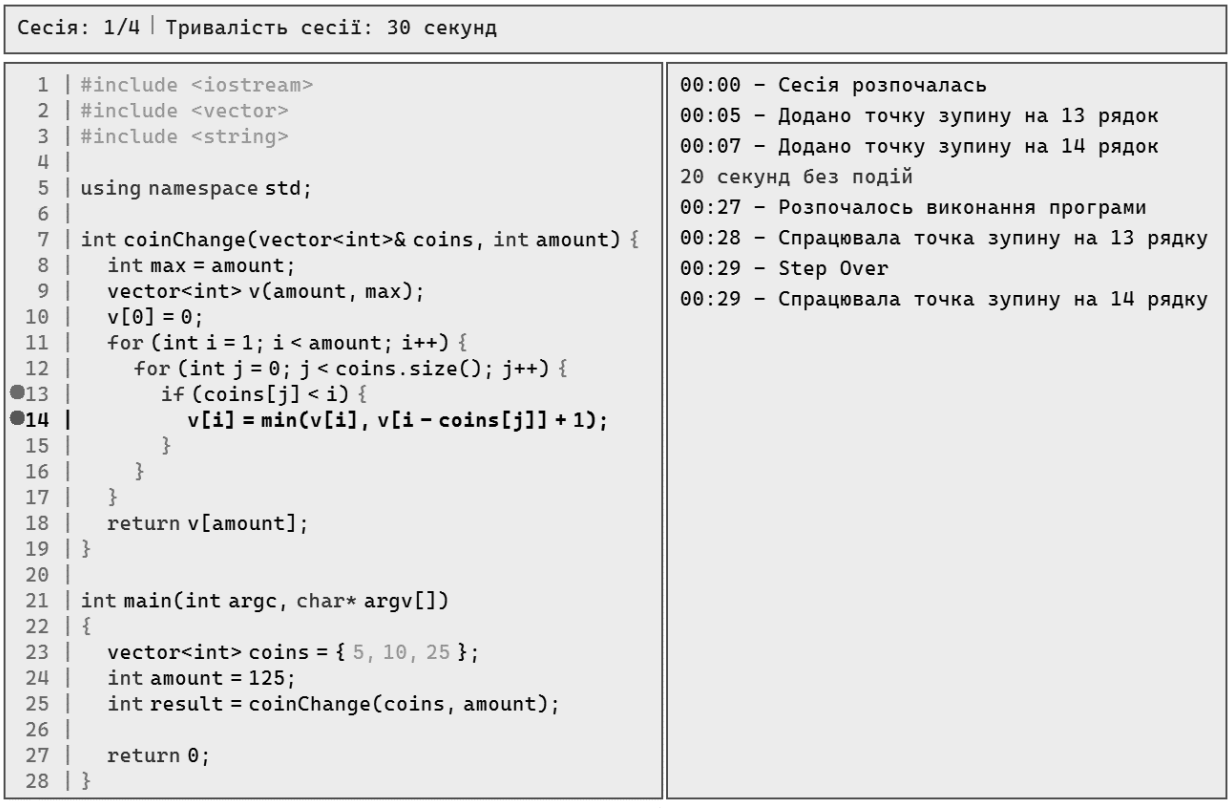

Рис. 1. Один кадр візуалізації процесу налагодження

рів. Під час візуалізації налагодження відповідні коментарі з'являються динамічно для забезпечення контексту і розуміння процесу.

Інтерфейс інструменту візуалізації процесу налагодження розроблений таким чином, аби бути інтуїтивно зрозумілим, інформативним і зручним для користувача, надаючи цілісне і глибоке уявлення про процес налагодження разом з відповідними коментарями та рекомендаціями.

Валідація розроблених інструментів

Для перевірки функціональності розробленого інструменту він пройшов ретельне тестування з використанням журналів, зібраних під час попередніх експериментів [10].

В експерименті у вигляді олімпіади з налагодження брали участь студенти з першого по п’яті курси. Під час експерименту було сформовано 487 файлів журналів подій для 41 учасника, на основі яких здійснювалася перевірка розробленого програмного забезпечення.

Етап тестування мав на меті оцінити точність відображення відео, а також правильність генерації коментарів.

Інструмент успішно реконструював процес налагодження, точно зафіксувавши послідовність дій учасників. Отримані відео відображають взаємодію студента з кодом, забезпечуючи чітке візуальне представлення сеансу налагодження.

В результаті тестування підтверджено працездатність представленого підходу. Отримані коментарі добре корелюють із результатами попереднього дослідження навичок налагодження [10].

Висновки

У роботі представлено новий підхід, який пропонує практичне рішення для покращення навичок налагодження за допомогою наочних відеопосібників з налагодження.

Запропонований підхід спрямований на підвищення ефективності та результативності навчання налагодження. Поєднуючи візуальне представлення з детальним описом дій, коментарями та рекомендаціями, наш підхід покращує розуміння процесів та полегшує передачу знань. Коментарі слугують джерелом інформації про причини певних дій, надаючи вказівки щодо потенційних поліпшень процесів та альтернативних підходів.

Інтеграція відеовізуалізації з даними файлів логів пропонує перспективний шлях для покращення навчального процесу з налагодження. Надаючи комплексне візуальне представлення процесу налагодження, новачки можуть краще зрозуміти його тонкощі, тим самим заповнюючи прогалину в навчанні налагодження.

Також цей інструмент може бути корисним для викладачів. Інструмент полегшує моніторинг розвитку навичок налагодження у студентів.

Аналізуючи індивідуальні навчальні траєкторії, викладачі мають змогу оцінити прогрес кожного студента з плином часу. Відстежуючи покращення навичок налагодження та визначаючи постійні проблеми, вони корегують методи викладання, щоб краще відповідати потребам своїх студентів.

Вивчаючи закономірності у процесах налагодження, викладачі можуть виявити типові помилки. Ця інформація може бути використана для розробки навчальних матеріалів, спрямованих на розв'язання проблемних питань.

Подальша робота може бути зосереджена на розширенні функціональності інструменту та проведенні додаткових досліджень для подальшого підтвердження його ефективності та зручності використання.

Література

1. M. Perscheid, B. Siegmund, M. Taeumel, R. Hirschfeld, Studying the advancement in debugging practice of professional software developers, *Software Quality Journal*, 2017. С. 83–110. https://doi.org/10.1007/s11219-015-9294-2.
2. T. Michaeli, R. Romeike, Improving debugging skills in the classroom: The effects of teaching a systematic debugging process, *14th Workshop in Primary and Secondary*

Computing Education, 2019. С. 1–7. https://doi.org/10.1145/3361721.3361724.

3. X. Gao, K. F. Hew, A Flipped Systematic Debugging Approach to Enhance Elementary Students' Program Debugging Performance and Optimize Cognitive Load, *Journal of Educational Computing Research*, 2023. С. 1064–1095. https://doi.org/10.1177/07356331221133560
4. R. Suzuki, G. Soares, A. Head, E. Glassman, R. Reis, M. Mongiovi, L. D'Antoni, B. Hartmann, TraceDiff: Debugging Unexpected Code Behavior Using Trace Divergences, 2017 *IEEE Symposium on Visual Languages and Human-Centric Computing*, 2017. С. 107–115. https://doi.org/10.1109/VLHCC.2017.8103457.
5. V. Shynkarenko, O. Zhevago, Visualization of program development process, *2019 IEEE 14th International Conference on Computer Sciences and Information Technologies*, 2019. С. 142–145. https://doi.org/10.1109/STC-CSIT.2019.8929774.
6. J. Moons, C. De Backer, The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism, *Computers & Education*, 2013. С. 368–384. https://doi.org/10.1016/j.compedu.2012.08.009.
7. J. Shi, K. Schneider, Creation of Human-friendly Videos for Debugging Automated GUI-Tests, *ICTSS 2021*, 2022. С. 141–147. https://doi.org/10.1007/978-3-031-04673-5_11.
8. V. C. Lee, Y. T. Yu, C. M. Tang, T. L. Wong, C. K. Poon, ViDA: A virtual debugging advisor for supporting learning in computer programming courses, *Journal of Computer Assisted Learning*, 2018. С. 243–258. http://doi.org/10.1111/jcal.12238.
9. V. Shynkarenko, O. Zhevaho, Development of a toolkit for analyzing software debugging processes using the constructive approach, *Eastern-European Journal of Enterprise Technologies*, 2020. С. 29–38. https://doi.org/10.15587/1729-4061.2020.215090.
10. В. І. Шинкаренко, О. О. Жеваго, Експериментальні дослідження процесів налагодження комп'ютерних програм студентами з використанням Process Mining, *Вісник Херсонського Національного технічного університету*, 2021. С. 83–98. http://doi.org/10.35546/kntu2078-4481.2021.3.9.

References

1. M. Perscheid, B. Siegmund, M. Taeumel, R. Hirschfeld, Studying the advancement in debugging practice of professional software developers, *Software Quality Journal*, 2017, pp. 83–110. https://doi.org/10.1007/s11219-015-9294-2.
2. T. Michaeli, R. Romeike, Improving debugging skills in the classroom: The effects of teaching a systematic debugging process, *14th Workshop in Primary and Secondary Computing Education*, 2019, pp. 1–7. https://doi.org/10.1145/3361721.3361724.
3. X. Gao, K. F. Hew, A Flipped Systematic Debugging Approach to Enhance Elementary Students' Program Debugging Performance and Optimize Cognitive Load, *Journal of Educational Computing Research*, 2023, pp. 1064–1095. https://doi.org/10.1177/07356331221133560
4. R. Suzuki, G. Soares, A. Head, E. Glassman, R. Reis, M. Mongiovi, L. D'Antoni, B. Hartmann, TraceDiff: Debugging Unexpected Code Behavior Using Trace Divergences, 2017 *IEEE Symposium on Visual Languages and Human-Centric Computing*, 2017, pp. 107–115. https://doi.org/10.1109/VLHCC.2017.8103457.
5. V. Shynkarenko, O. Zhevago, Visualization of program development process, *2019 IEEE 14th International Conference on Computer Sciences and Information Technologies*, 2019, pp. 142–145. https://doi.org/10.1109/STC-CSIT.2019.8929774.
6. J. Moons, C. De Backer, The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism, *Computers & Education*, 2013, pp. 368–384. https://doi.org/10.1016/j.compedu.2012.08.009.
7. J. Shi, K. Schneider, Creation of Human-friendly Videos for Debugging Automated GUI-Tests, *ICTSS 2021*, 2022, pp. 141–147. https://doi.org/10.1007/978-3-031-04673-5_11.
8. V. C. Lee, Y. T. Yu, C. M. Tang, T. L. Wong, C. K. Poon, ViDA: A virtual debugging advisor for supporting learning in computer programming courses, *Journal of Computer*

Assisted Learning, 2018, pp. 243–258. http://doi.org/10.1111/jcal.12238.
9. V. Shynkarenko, O. Zhevaho, Development of a toolkit for analyzing software debugging processes using the constructive approach, *Eastern-European Journal of Enterprise Technologies*, 2020, pp. 29–38. https://doi.org/10.15587/1729-4061.2020.215090.
10. V. Shynkarenko, O. Zhevaho, Experimental studies of debugging processes of computer programs by students using Process Mining, *Visnyk of Kherson National Technical University*, 2021, pp. 83–98. http://doi.org/10.35546/kntu2078-4481.2021.3.9.

Одержано: 09.04.2024
Внутрішня рецензія отримана: 19.04.2024
Зовнішня рецензія отримана: 23.04.2024

Про авторів:

Шинкаренко Віктор Іванович,
доктор технічних наук, професор
https://orcid.org/0000-0001-8738-7225.

Жеваго Олександр Олександрович,
доктор філософії, доцент
https://orcid.org/0000-0003-0019-8320.

Місце роботи авторів:

Український державний університет науки і технологій,
тел. +38-056-373-15-05
E-mail: office@ust.edu.ua
Веб-сайт: https://ust.edu.ua