

поєднавши можливості дослідження локальних властивостей з пошуком сценаріїв, що призводять до помилок.

Зменшення кількості помилок та вразливостей завжди було одним з пріоритетних напрямів розвитку комп'ютерної інженерії. В даній сфері важливою є наявність засобів, що є теоретично обґрунтованими, та мають відповідну наукову основу. Висока ефективність та масштабованість систем статичного аналізу коду та інших засобів визначення помилок, зумовлює актуальність подальшого їх розвитку, адже вони також орієнтовані на знаходження вразливостей, та потенційних сценаріїв, що до них призводять. Тому потрібні інженери з відповідною кваліфікацією, та подальші наукові розробки. Так як логіка некоректності є основою таких рішень і є дуальною до логіки Флойда-Хоара, що широко вивчається в межах навчальних дисциплін ІТ-галузі, доцільним є включення її розгляду до програм дисциплін верифікації програмних систем, кібербезпеки, формальних методів, та подальших досліджень.

### **Список використаних джерел**

1. Apt K. Fifty years of Hoare's logic. K. Apt, E. Olderog. *Formal Aspects Comput*, 2019. № 6. P. 751–807.
2. Facebook. Infer. URL: <https://fbinfer.com/docs/separation-logic-and-bi-abduction>.
3. Local Reasoning About the Presence of Bugs: Incorrectness Separation Logic. [A. Raad, J. Berdine, H. Dang та ін.]. *Lecture Notes in Computer Science*, 2020. № 12225. С. 225–252.
4. O'Hearn P. Incorrectness logic. Peter W. O'Hearn. *Proc. ACM Program. Lang.* 2020. № 4. С. 1–32.

## **ФОРМУВАННЯ У МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ ГОТОВНОСТІ ДО ВИКОРИСТАННЯ ПЛАТФОРМ РОЗРОБКИ Й ОЦІНКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **Кубік Михайло Анатолійович**

здобувач третього (освітньо-наукового) рівня вищої освіти спеціальності  
011 Освітні, педагогічні науки,  
Тернопільський національний педагогічний університет імені Володимира Гнатюка,  
kub@tnpu.edu.ua

### **Мартинюк Сергій Володимирович**

кандидат фізико-математичних наук, доцент кафедри інформатики та методики її навчання,  
Тернопільський національний педагогічний університет імені Володимира Гнатюка,  
sergmart65@tnpu.edu.ua

Сучасний світ переживає стрімкий розвиток технологій, що вимагає постійного оновлення й адаптації освітньої системи до нових вимог. У цьому контексті особливо актуальною є підготовка майбутніх учителів інформатики, які мають бути готові до використання платформ розробки й оцінки програмного забезпечення. Тому важливо оцінити важливість такої підготовки та методи формування готовності майбутніх учителів до цих навичок.

В умовах інформатизації суспільства інформатика як наукова дисципліна стає однією з фундаментальних галузей наукового знання, у рамках якої вивчаються інформаційні процеси, методи та засоби отримання, опрацювання,

передавання, зберігання та використання різноманітних даних і повідомлень [2, с. 9]. Вона не лише надає знання про комп'ютерні технології, але й сприяє розвитку критичного мислення, аналітичних навичок і творчого мислення учнів. Однак, щоб майбутні вчителі могли успішно викладати інформатику, вони повинні мати не лише теоретичні знання, але й практичні навички у використанні сучасних платформ розробки й оцінки програмного забезпечення.

На сьогоднішній день на ринку існує безліч платформ розробки програмного забезпечення, які надають можливості для створення різноманітних програм і додатків. Деякі з них, такі як Python, Java або Scratch, спрощують процес розробки та навчання програмуванню. Інші, наприклад, GitHub, надають можливості для спільної роботи над проектами й оцінки програмного коду. Важливо, щоб майбутні вчителі оволоділи принаймні базовими навичками роботи з цими платформами, щоб ефективно використовувати їх у своїй практиці.

Оцінка програмного забезпечення є одним із ключових етапів у процесі розробки програм. Вона дозволяє визначити якість та ефективність програми перед її впровадженням у життя [2]. Застосування методів оцінки допомагає виявити потенційні проблеми та помилки в програмному коді, забезпечуючи високу якість і надійність програми.

Одним із ключових аспектів оцінки програмного забезпечення є тестування. Цей процес включає в себе проведення різноманітних тестів, таких як модульні тести, інтеграційні тести, системні тести тощо. Наприклад, тестування може виявити помилки у функціональності програми, помилки в алгоритмах, проблеми з безпекою чи ефективність виконання програмних операцій.

Тестування відіграє життєво важливу роль у процесі розробки і створення якісного програмного забезпечення. Необхідно серйозно ставитися до аналізу та проєктування структурованого процесу, який забезпечить своєчасний і успішний випуск проєкту [3].

Аналіз коду є важливою складовою оцінки програмного забезпечення. Цей процес передбачає детальний огляд програмного коду з метою виявлення потенційних проблем, використання кращих практик програмування й оптимізації продуктивності програми. Майбутні вчителі повинні мати глибоке розуміння структури коду, можливостей використання різних мов програмування та знання найкращих практик аналізу й оптимізації коду.

Крім тестування і аналізу коду, важливою частиною оцінки програмного забезпечення є вміння користуватися відповідними інструментами для проведення цих процедур. Наприклад, для тестування можуть використовуватися автоматизовані тестові системи, а для аналізу коду – спеціалізовані інструменти для виявлення помилок та оптимізації. Майбутні вчителі повинні мати практичні навички роботи з цими інструментами та знати їх особливості та можливості.

Тому розуміння методів оцінки програмного забезпечення є ключовим для майбутніх вчителів, які прагнуть ефективно використовувати сучасні технології у своїй практиці навчання та розробки програмного забезпечення. Засвоєння цих

навичок допоможе їм готувати учнів до вимог сучасності та забезпечувати високу якість своєї майбутньої професійної діяльності.

Формування готовності майбутніх учителів до використання платформ розробки й оцінки програмного забезпечення має відбуватися на кількох рівнях. По-перше, це може бути включено у навчальні плани і програми вищих навчальних закладів, де майбутні вчителі отримують необхідні знання та навички. По-друге, важливо проводити практичні заняття та майстер-класи з використання платформ розробки й оцінки програмного забезпечення. Також необхідно стимулювати самостійну роботу майбутніх учителів у цьому напрямку, наприклад, за допомогою завдань і проєктів. Застосування платформ розробки й оцінки програмного забезпечення в навчальному процесі може зробити заняття цікавішим і практично орієнтованим для учнів. Наприклад, учні можуть створювати власні програми або додатки, а потім аналізувати їх ефективність та якість за допомогою відповідних інструментів оцінки. Це дозволить їм не лише отримувати теоретичні знання, але й використовувати їх у практичних ситуаціях.

Формування готовності майбутніх учителів інформатики до використання платформ розробки й оцінки програмного забезпечення є надзвичайно важливим завданням, яке стоїть перед освітніми установами сьогодні. Засвоєння цих навичок впливає на якість навчання та підготовку молодого покоління до сучасних вимог ринку праці.

Застосування платформ розробки та оцінки програмного забезпечення у навчальному процесі дозволить створити учням реальні сценарії роботи, які сприятимуть збільшенню мотивації й інтересу до предмета. Це не лише дозволить розвивати практичні навички, але й сприятиме формуванню критичного мислення та творчого підходу до розв'язання завдань.

Для ефективного впровадження сучасних технологій у навчальний процес важливо підвищувати кваліфікацію вчителів інформатики та надавати їм можливість постійного професійного розвитку. Це допоможе вчителям бути в курсі останніх технологічних тенденцій та ефективно впроваджувати їх у навчальний процес. Набуті майбутніми вчителями навички роботи з платформами розробки та оцінки програмного забезпечення не лише полегшать їхню власну роботу в майбутньому, але й підготують студентів до вимог індустрії його використання та дозволять бути конкурентоспроможними на ринку праці. Необхідно також активно співпрацювати з представниками ІТ-індустрії для оновлення навчальних програм і забезпечення відповідності вимогам ринку [1, с. 40]. Це дозволить максимально ефективно підготувати майбутніх учителів до реальних викликів, з якими вони зіткнуться у професійній діяльності.

### Список використаних джерел

1. Експортна стратегія для сектору інформаційних технологій. URL: <https://www.me.gov.ua/Documents/Download?id=6deb0289-40af-41ab-91eb-ce7f8d298fc4> (дата звернення: 25.03.2024).
2. Носенко Т. І., Бонч-Бруевич Г. Ф. Соціальна інформатика: навчальний посібник. К. : ун-т ім. Б. Грінченка, ін-т суспільства, каф. інформатики, 2013. К. 136 с.
3. Тестування веб-проєктів: основні етапи та поради. URL: <https://qalight.ua/baza-znaniy/testuvannya-veb-proektiv-osnovni-etapi-ta-poradi> (дата звернення: 20.03.2024).