



POLTAVA UNIVERSITY OF
ECONOMICS AND TRADE

ЗБІРНИК НАУКОВИХ СТАТЕЙ МАГІСТРІВ



Полтава
2022

УДК 339.1+640+664+37+657+005+004+80(062.552)

3-41

Друкується відповідно до Наказу по університету № 173-Н від 05 вересня 2022 р.

Редакційна колегія:

Головний редактор – **О. О. Нестуля**, д. і. н., професор, ректор Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі» (ПУЕТ).

Заступник головного редактора – **Н. С. Педченко**, д. е. н., професор, перший проректор ПУЕТ.

Відповідальний секретар – **Н. І. Манжура**, завідувач науково-організаційного відділу ПУЕТ.

Відповідальні редактори:

О. В. Гасій, к. е. н., доцент, директор Навчально-наукового центру забезпечення якості вищої освіти ПУЕТ;

С. В. Гаркуша, д. т. н., професор, в. о. директора Навчально-наукового інституту міжнародної освіти ПУЕТ;

А. С. Ткаченко, к. т. н., доцент, директор Навчально-наукового інституту денної освіти ПУЕТ;

В. Л. Шимановська, директор Навчально-наукового інституту заочно-дистанційного навчання ПУЕТ;

Л. М. Діденко, в. о. директора Центру інформаційного забезпечення освітнього процесу ПУЕТ.

Члени редакційної колегії:

А. В. Артеменко, к. е. н., доцент, завідувач кафедри міжнародної економіки та міжнародних економічних відносин ПУЕТ;

Н. М. Бобух, д. філол. н., професор, завідувач кафедри української, іноземних мов та перекладу ПУЕТ;

В. Л. Іщенко, к. філол. н., доцент завідувач кафедри ділової іноземної мови ПУЕТ;

Т. В. Капліна, д. т. н., професор, завідувач кафедри готельно-ресторанної та курортної справи ПУЕТ;

Т. А. Костишина, д. е. н., професор, завідувач кафедри управління персоналом, економіки праці та економічної теорії ПУЕТ;

Ю. С. Матвієнко, к. п. н., проректор з науково-педагогічної роботи ПУЕТ;

А. І. Мілька, к. е. н., доцент, завідувач кафедри бухгалтерського обліку і аудиту ПУЕТ;

О. В. Ольховська, к. ф.-м. н., завідувач кафедри комп'ютерних наук та інформаційних технологій ПУЕТ;

І. М. Петренко, д. і. н., професор, завідувач кафедри педагогіки та суспільних наук ПУЕТ;

О. В. Яріш, к. е. н., доцент, завідувач кафедри фінансів та банківської справи ПУЕТ.

Збірник наукових статей магістрів. – Полтава : ПУЕТ, 3-41 2022. – 268 с.

ISBN 978-966-184-435-2

У збірнику представлено результати наукових досліджень магістрів спеціальностей: Готельно-ресторанна справа; Економіка; Комп'ютерні науки; Міжнародні економічні відносини; Облік і оподаткування; Освітні педагогічні науки; Філологія; Фінанси, банківська справа та страхування.

УДК 339.1+640+664+37+657+005+004+80(062.552)

Матеріали друкуються в авторській редакції мовами оригіналів.

За виклад, зміст і достовірність матеріалів відповідальні автори.

Розповсюдження та тиражування без офіційного дозволу ПУЕТ заборонено.

© Вищий навчальний заклад Укоопспілки
«Полтавський університет економіки і торгівлі», 2022

ISBN 978-966-184-435-2

Семикоз Д. С., Черненко О. О.

Методи штучного інтелекту при стратегічному
плануванні діяльності компанії.....87

Сімперович М. М., Черненко О. О.

Алгоритм тренажера з теми «Верифікація
програм» для англomовного дистанційного
курсу «Теорія програмування»92

Собіборець О. Ю., Гаркуша С. В., Ольховська О. В.

Розробка навчального тренажера з теми «Системи
числення, арифметичні операції в різних системах числення»
дисципліни «Архітектура обчислювальних систем»97

Chagonda N. S., Koshova O. P., Orikhivska O. G.

Some Peculiarities of Development of a Simulator
for the Distance Learning Course Programming II 102

Ghrmida S. M., Harkusha S. V., Koshova O. P., Orikhivska O. G.

Some Peculiarities of Development of
Desktop Application «Using Array in Java» 105

СПЕЦІАЛЬНІСТЬ

МІЖНАРОДНІ ЕКОНОМІЧНІ ВІДНОСИНИ

Освітня програма «Міжнародні економічні відносини»

Білокіз М. Б., Пожар А. А.

Особливості виникнення та поширення
кредитних кооперативів у країнах світу..... 109

Коробко А. В., Артеменко А. В.

Глобальні тенденції розвитку світової
криптовалютної індустрії..... 113

Радевич В. О., Артеменко А. В.

Експортно-імпортна діяльність українських
підприємств: проблеми та перспективи розвитку 117

Висновки. Штучний інтелект є невід’ємною частинною нашого життя. Зустріти його прояви ми можемо на кожному кроці, починаючи із наших мобільних телефонів і закінчуючи комп’ютерами які управляють фабриками, заводами та великими компаніями. В нашому випадку було створено систему редактору розкладу, який буде оптимізувати роботу та зменшувати ризик на появу нових проблем.

Список використаних джерел

1. Шершньова З. Є., Оборська С. В., Ратушний Ю. М. Стратегічне управління : навч.-метод. посіб. – Київ : КНЕУ, 2001. – 232 с.
2. Сироджа И. Б. Квантовые модели и методы искусственного интеллекта для принятия решений и управления. – Киев : Наукова думка, 2002. – 490 с.
3. Глібовець М. М, Олецький О. В. Штучний інтелект : підруч. для студ. вищ. навч. закладів. – Київ : Видавничий дім «КМ Академія», 2002. – 336 с.
4. Зайченко Ю. П. Основи проектування інтелектуальних систем : навч. посіб. – Київ : Видавничий дім «Слово», 2004. – 352 с.

УДК 519.681

АЛГОРИТМ ТРЕНАЖЕРУ З ТЕМИ «ВЕРИФІКАЦІЯ ПРОГРАМ» ДЛЯ АНГЛОМОВНОГО ДИСТАНСІЙНОГО КУРСУ «ТЕОРІЯ ПРОГРАМУВАННЯ»

Сімперович М. М., магістр спеціальності Комп’ютерні науки освітня програма «Комп’ютерні науки»

Черненко О. О., к. ф.-м. н., доцент, доцент кафедри комп’ютерних наук та інформаційних технологій

Анотація. В статті розглядається проблема верифікації програмного забезпечення та опис розробки алгоритму для навчального тренажеру з теми «Верифікація програм» для англomовного дистанційного курсу «Теорія програмування».

Ключові слова: верифікація, тестування, алгоритм, тренажер.

Abstract. The article deals with the problem of software verification and a description of the development of an algorithm for

a training simulator on the topic “Verification of programs” for the English distance course “Programming Theory”.

Keywords: verification, testing, algorithm, simulator.

Постановка проблеми. Проблеми верифікації (тобто доказ правильності) програм займають центральне місце при розробки програмного забезпечення. Під правильністю програм розуміється їх відповідність різним умовам коректності, безпеки, стійкості у разі непередбаченої поведінки оточення, ефективності використання ресурсів, часу та пам’яті, оптимальності реалізованих у програмі алгоритмів тощо. Як правило, для перевірки коректності роботи програми її тестують, тобто аналізують її поведінку на деяких вхідних даних. Проте тестування має свій недолік: якщо його можливо провести не для всіх допустимих вхідних даних, а тільки для невеликої частини (що має місце майже завжди), воно не може служити гарантованим обґрунтуванням того, що програма володіє властивостями, що перевіряються. Як зазначив Дейкстра [2], тестування може лише допомогти виявити деякі помилки, але аж ніяк не довести їхню відсутність. Але у багатьох програмах наявність навіть незначних помилок категорично неприпустима.

Аналіз основних досліджень і публікацій. Наведемо приклад, що ілюструє наявність помилок навіть у дуже простих програм, правильність яких на перший погляд не викликає жодного сумніву. Розглянемо програму P , завдання якої полягає у зарахуванні грошей на рахунок клієнта банку. Кількість грошей на рахунку цього клієнта зберігається у базі даних банку у змінній P . Коли P виконує дії із зарахування суми s з цього приводу, вона виконує такі дії:

- копіює у свою внутрішню пам’ять значення змінної x ;
- обчислює нове значення, яке має мати змінна x , воно дорівнює сумі поточного значення x і сумі, що зараховується s , і заносить до змінної x це нове значення.

Навіть якщо програма P виконує всі свої дії правильно, це не гарантує коректності обслуговування клієнта у разі, коли стан його рахунка може змінюватися кількома такими програмами.

Розглянемо ситуацію, коли стан рахунку клієнта змінюють дві програми P_1 та P_2 описаного вище типу. Можливий наступний варіант спільного виконання цих програм:

- спочатку програма P_1 виконує свою першу дію, вона починається в момент часу t_1 , а заключна дія P_1 (оновлення значення x) відбувається в момент часу t_2 , та

- в момент часу, що лежить в інтервалі між t_1 та t_2 , програма P_2 починає свою операцію зарахування грошей на рахунок клієнта, причому виконання першої дії програми P_2 (копіювання значення змінної x) проводиться до виконання заключної дії програми P_1 .

Після завершення роботи обох програм ті гроші, які зарахувала на рахунок клієнта одна з програм P_1 , P_2 просто пропадуть.

Помилку описаного вище типу не можливо знайти шляхом тестування, так як операція зарахування грошей на рахунок клієнта виконується практично миттєво, і тому серед тестів, якими можна аналізувати програми такого типу, з великою ймовірністю можуть відсутні такі тести, в яких дві різні програми, що обслуговують рахунки клієнтів, майже одночасно звертаються до одного та того ж ресурсу пам'яті [5].

Формулювання мети. Метою статті є виявлення проблем при верифікації програмного забезпечення та розробка алгоритму навчального тренажеру з теми «Верифікація програм» для англomовного дистанційного курсу «Теорія програмування».

Виклад основного матеріалу дослідження. Верифікація програм дозволяє за допомогою строгих правил встановити відповідність програми своєму призначенню, для поліпшення процесу навчання розроблено алгоритм тренажеру з теми «Верифікація програм» для англomовного дистанційного курсу «Теорія програмування».

Тренажер складатиметься з двох вправ. Перша вправа складатиметься з трьох питань з кількома варіантами відповідей, серед яких правильних може бути декілька, друга вправа складатиметься також з трьох питань з кількома відповідями, серед яких правильна тільки одна. Розглянемо алгоритм тренажеру на прикладі вправи 1.

Крок 1. Умова завдання «Допустимий інтервал значень для числа **a** та **b** – від 0 до 10 включно. Для тестування функції порівняння двох чисел на рівність **a == b** необхідні такі приклади»:

- **a = 3, b = 5, значення функції = false (1)**
- **a = 3, b = 3, значення функції = true (2)**
- **a = 0, b = 0, значення функції = true (3)**
- **a = 10, b = 10, значення функції = true (4)**
- **a = 0, b = 10, значення функції = false (5)**
- **a = 10, b = 0, значення функції = false (6)»**

Якщо користувач вибрав 1, 2, 5 та 6 приклади, відповідь зараховується і користувач може перейти на наступний крок, якщо ж відповідь не вірна, вибрані варіанти очищаються і на екран відображаються вірні відповіді, причому користувач залишається на цьому ж вікні для повторного вибору відповіді.

Крок 2. Умова завдання «Допустимий інтервал значень для числа **a** та **b** – від 0 до 10 включно. Для тестування функції порівняння двох чисел на рівність **a < b** необхідні такі приклади:

- **a = 5, b = 6, значення функції = true (1)**
- **a = 5, b = 4, значення функції = false (2)**
- **a = 6, b = 6, значення функції = false (3)**
- **a = 0, b = 0, значення функції = false (4)**
- **a = 0, b = 10, значення функції = true (5)**
- **a = 10, b = 0, значення функції = false (6)**
- **a = 10, b = 10, значення функції = false (7)»**

Якщо вибрана 1, 2, 3, 5 та 6 приклади, відповідь зараховується і користувач може перейти на наступний крок, якщо ж відповідь не вірна, вибрані варіанти очищаються і на екран відображаються вірні відповіді, причому користувач залишається на цьому ж вікні для повторного вибору відповіді.

Крок 3. Умова завдання «Допустимий інтервал значень для числа **a** та **b** – від 0 до 10 включно. Для тестування функції порівняння двох чисел на рівність **a >= b** необхідні такі приклади:

- **a = 5, b = 6, значення функції = false (1)**
- **a = 5, b = 4, значення функції = true (2)**
- **a = 6, b = 6, значення функції = true (3)**

- $a = 0, b = 0$, значення функції = true (4)
- **$a = 0, b = 10$, значення функції = false (5)**
- **$a = 10, b = 0$, значення функції = true (6)**
- $a = 10, b = 10$, значення функції = true (7)»

Якщо користувач вибрав 1, 2, 3, 5 та 6 приклади, відповідь зараховується і користувач може перейти на наступний крок, якщо ж відповідь не вірна, вибрані варіанти очищаються і на екран відображаються вірні відповіді, причому користувач залишається на цьому ж вікні для повторного вибору відповіді.

Після проходження вправи 1 відображається повідомлення, що завершено виконання вправи 1 та пропонується перейти до наступної вправи або завершити проходження тренажера.

Висновок. У цій статті розглянута проблема верифікації програм та описано декілька кроків реалізації алгоритму тренажера з теми «Верифікація програм» для англomовного дистанційного курсу «Теорія програмування», який необхідний для покращення процесу навчання з вищі вказаної теми.

Список використаних джерел

1. Миронов А. М. Верификация программ методом инвариантов, Интеллектуальные системы. Теория и приложения. Т. 21, Вып. 4: 2017. С. 31–49.
2. Лекции лауреатов премии Тьюринга. Москва : Мир, 1993.
3. Андерсон Р. Доказательство правильности программ / Р. Андерсон : пер. с англ. Б. Н. Зобниной. Москва : Мир, 1982. 168 с.
4. Лавріщева К. М. Програмна інженерія. Національна академія наук України. Київ. 2008. С. 145–181.
5. Навчальні матеріали для студентів. Верифікація програм. Семантики мов програмування. 2018 [Електронний ресурс]. URL: <https://vunivere.ru/work23796>.
6. Інформатика та системні науки (ІСН-2017) : матеріали VIII Всеукр. наук.-практ. конф. за міжнародною участю (м. Полтава, 16–18 березня 2017 р.) / за ред. Ємця О. О. Полтава : ПУЕТ, 2017. С. 146–149.
7. Комп'ютерні науки і прикладна математика (КНіПМ-2020) : матеріали наук.-практ. семінару. Вип. 5 / за ред. Ємця О. О. – Полтава : кафедра ММСІ ПУЕТ, 2020. 63 с.