

## ДОДАТОК А

## КОД ФРЕЙМВОРКУ АВТОМАТИЗАЦІЇ

```
from pywinauto import Desktop, Application
import pyautogui
import PIL
import pytest
import time
import allure

@allure.id(1)
@allure.story("Simple test")
@allure.title("test_integration_print_feature")
def test_smoke_main_window():
    app = Application(backend="uia").start("notepad.exe")
    dlg_main = app.UntitledNotepad
    dlg_main.wait('visible')

def test_smoke_open_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Open")
    dlg_open = app.UntitledNotepad.Open
    dlg_open.wait('visible')

def test_smoke_save_as_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Save As")
    dlg_save = app.UntitledNotepad.SaveAs
    dlg_save.wait('visible')

def test_smoke_font_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("Format->Font...")
    dlg_font = app.UntitledNotepad.Font
    dlg_font.wait('visible')

def test_smoke_print_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Print")
    dlg_print = app.UntitledNotepad.Print
    dlg_print.wait('visible')

def test_auto_main_window():
    app = Application(backend="uia").start("notepad.exe")
    dlg_main = app.UntitledNotepad
    dlg_main.wait('visible')
    app.UntitledNotepad.menu_select("Help->About Notepad")
    assert app.UntitledNotepad.AboutNotepad.Static2.texts() == ['Version 21H2
(OS Build 19044.170)']

def test_auto_open_feature():
```

```

app = Application(backend="uia").start("notepad.exe")
app.UntitledNotepad.menu_select("File->Open")
dlg_open = app.UntitledNotepad.Open
dlg_open.Pane.wait('visible')
dlg_open.Static.wait('visible')
dlg_open.ComboBox.wait('visible')
dlg_open.ComboBox2.wait('visible')
dlg_open.Static3.wait('visible')
dlg_open.ComboBox3.wait('visible')
dlg_open.Open.wait('visible')
dlg_open.Cancel.wait('visible')
dlg_open.Pane5.wait('visible')
dlg_open.Pane6.wait('visible')
dlg_open.Pane7.wait('visible')
dlg_open.Pane9.wait('visible')
dlg_open.Pane10.wait('visible')
dlg_open.Pane11.wait('visible')
dlg_open.TitleBar.wait('visible')

def test_auto_save_as_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Save As")
    dlg_save = app.UntitledNotepad.SaveAs
    dlg_save.wait('visible')
    dlg_save.ComboBox.wait('visible')
    dlg_save.Pane6.wait('visible')
    dlg_save.Pane7.wait('visible')
    dlg_save.Pane9.wait('visible')
    dlg_save.Pane10.wait('visible')

def test_auto_font_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("Format->Font...")
    dlg_font = app.UntitledNotepad.Font
    dlg_font.wait('visible')
    dlg_font.Pane10.wait('visible')
    dlg_font.Pane11.wait('visible')
    dlg_font.TitleBar.wait('visible')
    dlg_font.ComboBox2.wait('visible')
    dlg_font.Static3.wait('visible')
    dlg_font.ComboBox3.wait('visible')

def test_auto_print_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Print")
    dlg_print = app.UntitledNotepad.Print
    dlg_print.wait('visible')
    dlg_print.Pane11.wait('visible')
    dlg_print.TitleBar.wait('visible')
    dlg_print.ComboBox2.wait('visible')
    dlg_print.Pane7.wait('visible')
    dlg_print.Pane9.wait('visible')

def test_functional_main_window():
    app = Application(backend="uia").start("notepad.exe")
    pyautogui.write("Test22")
    pic = pyautogui.screenshot()
    pic.save('Screenshot3.png')

```

```

def test_functional_open_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Open")
    dlg_open = app.UntitledNotepad.Open
    dlg_open.wait('visible')
    app.UntitledNotepad.Open.Cancel.click()
    pyautogui.write('Test1.txt')
    pyautogui.press('enter')
    dlg_new_open = app.window(title='Test1.txt - Notepad')
    dlg_new_open.wait('visible')

def test_functional_save_as_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.type_keys("Anton")
    app.UntitledNotepad.menu_select("File->Save As")
    dlg_save = app.UntitledNotepad.SaveAs
    dlg_save.wait('visible')
    pyautogui.write("Testtest2")
    pyautogui.press("enter")
    dlg_new_save = app.window(title='Testtest2.txt - Notepad')
    dlg_new_save.wait('visible')

def test_functional_font_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.type_keys("Anton")
    pic = pyautogui.screenshot()
    pic.save('Screenshot1.png')
    for i in "Anton":
        pyautogui.press('backspace')
    app.UntitledNotepad.menu_select("Format->Font...")
    pyautogui.press('backspace')
    time.sleep(12)
    pyautogui.write('Algerian')
    pyautogui.press('enter')
    app.UntitledNotepad.type_keys("Anton")
    pic2 = pyautogui.screenshot()
    pic2.save('Screenshot2.png')

def test_functional_print_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Print")
    dlg_print = app.UntitledNotepad.Print
    dlg_print.wait('visible')
    app.UntitledNotepad.Print.Cancel.click()
    pyautogui.hotkey('ctrl', 'p')
    pyautogui.press('enter')
    pyautogui.write("pdfctest0")
    pyautogui.press('enter')
    app.UntitledNotepad.menu_select("File->Open")
    dlg_open = app.UntitledNotepad.Ope
    dlg_open.wait('visible')
    pyautogui.write('pdfctest0.pdf')
    pyautogui.press('enter')
    dlg_new_open = app.window(title='pdfctest0.pdf - Notepad')
    dlg_new_open.wait('visible')

def test_integration_main_window():
    app = Application(backend="uia").start("notepad.exe")
    dlg_main = app.UntitledNotepad
    dlg_main.StatusBar.wait('visible')

```

```
dlg_main.Edit.wait('visible')
dlg_main.TitleBar.wait('visible')
dlg_main.ApplicationMenu.wait('visible')

def test_integration_open_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Open")
    dlg_open = app.UntitledNotepad.Open
    dlg_open.wait('visible')
    app.UntitledNotepad.Open.Cancel.click()
    pyautogui.hotkey('ctrl', 'o')
    dlg_open.wait('visible')

def test_integration_save_as_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Save As")
    dlg_save = app.UntitledNotepad.SaveAs
    dlg_save.wait('visible')
    app.UntitledNotepad.SaveAs.Cancel.click()
    pyautogui.hotkey('ctrl', 's')
    dlg_save.wait('visible')

def test_integration_print_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("File->Print")
    dlg_print = app.UntitledNotepad.Print
    dlg_print.wait('visible')
    app.UntitledNotepad.Print.Cancel.click()
    pyautogui.hotkey('ctrl', 'p')
    dlg_print.wait('visible')

def test_integration_font_feature():
    app = Application(backend="uia").start("notepad.exe")
    app.UntitledNotepad.menu_select("Format->Font...")
    dlg_font = app.UntitledNotepad.Font
    dlg_font.wait('visible')
    app.UntitledNotepad.Font.Cancel.click()
    pyautogui.hotkey('ctrl', 'p')
    dlg_font.wait('visible')
```





**International Science Group**

**ISG-KONF.COM**

**XXXVI  
INTERNATIONAL SCIENTIFIC  
AND PRACTICAL CONFERENCE  
"THE MAIN PROSPECTS FOR THE DEVELOPMENT OF  
SCIENCE IN MODERN LIFE"**

**Warsaw, Poland  
September 13 - 16, 2022**

**ISBN 979-8-88796-808-7**

**DOI 10.46299/ISG.2022.1.36**

# **THE MAIN PROSPECTS FOR THE DEVELOPMENT OF SCIENCE IN MODERN LIFE**

Proceedings of the XXXVI International Scientific and Practical Conference

Warsaw, Poland  
September 13 – 16, 2022

**UDC 01.1**

The XXXVI International Scientific and Practical Conference «The main prospects for the development of science in modern life», September 13 – 16, 2022, Warsaw, Poland. 392 p.

**ISBN – 979-8-88796-808-7**

**DOI – 10.46299/ISG.2022.1.36**

**EDITORIAL BOARD**

<u>Pluzhnik Elena</u>	Professor of the Department of Criminal Law and Criminology Odessa State University of Internal Affairs Candidate of Law, Associate Professor
<u>Liudmyla Polyvana</u>	Department of Accounting and Auditing Kharkiv National Technical University of Agriculture named after Petr Vasilenko, Ukraine
<u>Mushenyk Iryna</u>	Candidate of Economic Sciences, Associate Professor of Mathematical Disciplines, Informatics and Modeling. Podolsk State Agrarian Technical University
<u>Prudka Liudmyla</u>	Odessa State University of Internal Affairs, Associate Professor of Criminology and Psychology Department
<u>Marchenko Dmytro</u>	PhD, Associate Professor, Lecturer, Deputy Dean on Academic Affairs Faculty of Engineering and Energy
<u>Harchenko Roman</u>	Candidate of Technical Sciences, specialty 05.22.20 - operation and repair of vehicles.
<u>Belei Svitlana</u>	Ph.D., Associate Professor, Department of Economics and Security of Enterprise
<u>Lidiya Parashchuk</u>	PhD in specialty 05.17.11 "Technology of refractory non-metallic materials"
<u>Levon Mariia</u>	Candidate of Medical Sciences, Associate Professor, Scientific direction - morphology of the human digestive system
<u>Hubal Halyna Mykolaiivna</u>	Ph.D. in Physical and Mathematical Sciences, Associate Professor



80.	Гончаров Н.А., Лесная Ю.Е., Семёнов А.С., Малахов С.В. МОДЕЛИРОВАНИЕ АТАКИ СТЕГАНОКОНТЕНТА НА КОРОТКОМ СТЕКЕ ВЫБОРКИ ПАРАМЕТРОВ СЕРИЙ ПРИ ГРУБЫХ ОЦЕНКАХ ПОДОБИЯ ИСХОДНЫХ ДАННЫХ	344
81.	Дьоміна В.М., Чаговець В.В., Івах А.В. МЕТОДИКА ОЦІНКИ РИСКІВ SHADOW IT	348
82.	Матківський С.В., Матішин Л.І. ДОСЛІДЖЕННЯ ВПЛИВУ КОНДЕНСАТОНАСИЧЕНОСТІ ПРИВИБІЙНОЇ ЗОНИ ПЛАСТА НА ДЕБІТ СВЕРДЛОВИНИ	353
83.	Михайлов А.О. СІЛЬСЬКОГОСПОДАРСЬКІ ТРАКТОРИ ДЛЯ СУЧАСНИХ АГРОТЕХНОЛОГІЙ АПК УКРАЇНИ	358
84.	Полубехін А. ВИВЧЕННЯ МЕТОДІВ ТЕСТУВАННЯ ДЕСКТОПНИХ ЗАСТОСУНКІВ	362
85.	Темчур К. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗПІЗНАВАННЯ ДОРОЖНІХ ЗНАКІВ	367
86.	Шалигін А., Єрилкін А., Марченко О., Кудрявцев А., Шевченко Ю. АНАЛІЗ СВІТОВОГО ДОСВІДУ СТВОРЕННЯ ГІРОКОПТЕРІВ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ЇХНЬОГО ЗАСТОСУВАННЯ ВІЙСЬКАМИ	371
87.	Шишацький А.В., Журавський Ю.В., Гурський Т.Г., Возняк Р.М., Прис Г.П. МАТЕМАТИЧНА МОДЕЛЬ УПРАВЛІННЯ РАДІОРЕСУРСОМ СИСТЕМ РАДІОЗВ'ЯЗКУ СПЕЦІАЛЬНОГО ПРИЗНАЧЕННЯ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ	376
TOURISM		
88.	Бойко В.О. ВИННИЙ ТУРИЗМ ЯК ОДИН ІЗ НАПРЯМІВ ЕНОГАСТРОНОМІЧНОЇ КУЛЬТУРИ	385

## ВИВЧЕННЯ МЕТОДІВ ТЕСТУВАННЯ ДЕСКТОПНИХ ЗАСТОСУНКІВ

**Полубехін Антон,**  
Магістр з інформатики  
Харківський національний університет радіоелектроніки

У сучасному світі розробляється все більше різних інформаційних систем, веб та десктопних застосунків. Дані вказують, що у 2020 році було завантажено більше, ніж 204 млрд. мобільних програмних засобів, що на 45% більше, ніж було у 2016 році. Статистичні показники доводять, що люди почали частіше використовувати різні застосунки у повсякденному житті [1–5].

Тестування програмного забезпечення – це метод перевірки того, чи фактичний програмний продукт відповідає очікуваним вимогам, а також переконатися, що програмний продукт без дефектів. Метою тестування програмного забезпечення є виявлення помилок, прогалин або вимог [6–11].

Існує багато видів тестування, які мають різні функції і для виконання яких потрібно мати спеціальні знання. Прийнято класифікувати види тестування. Зазвичай, вони розподіляються на нефункціональні (рис. 1) та функціональні.

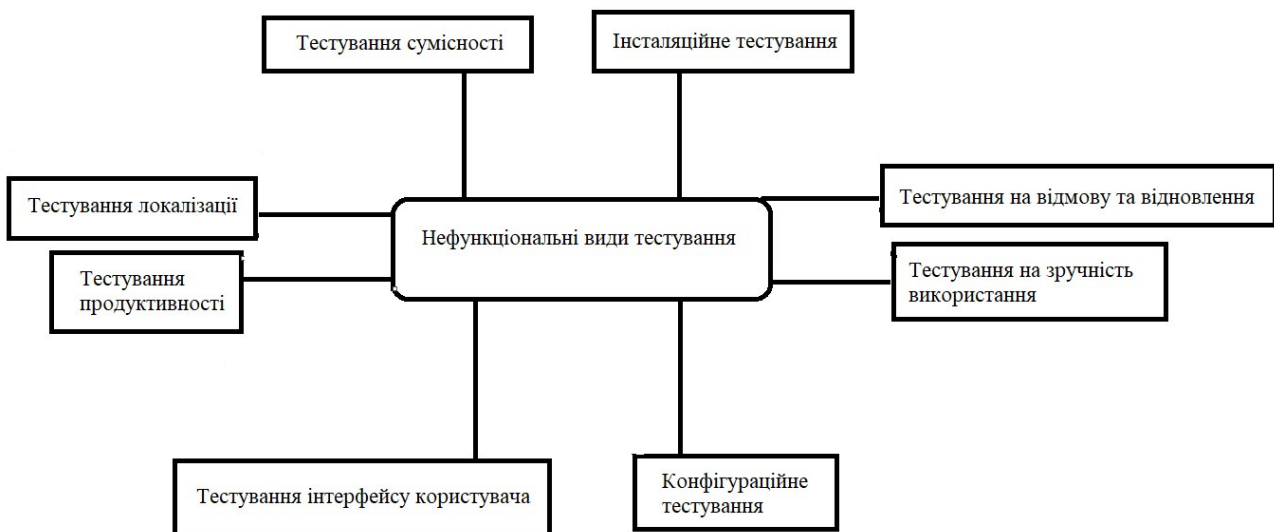


Рисунок 1 – Нefункціональні види тестування

Нефункціональне тестування має на меті таку перевірку: як працює програмне забезпечення і наскільки добре.

Дане тестування буває таким [12–18]:

- тестування локалізації;
- тестування на відмову та відновлення;
- сумісності;
- конфігураційне;
- інсталяційне;
- тестування на зручність використання;

- тестування інтерфейсу користувача;
- тестування продуктивності.

Функціональне тестування перевіряє, що робить програмне забезпечення, воно поділяється на [1–3, 9, 11, 19]:

- функціональне;
- тестування безпеки;
- тестування взаємодії.

Нефункціональне тестування так само важливе, як і функціональне. Обидва гарантують, що програмний продукт працює належним чином.

Однак, нефункціональне тестування перевіряє деталі, які не розглядаються у функціональних тестах.

Насправді класифікацій видів тестування більше, їх можна класифікувати таким чином:

- за часом виконання;
- за формальністю;
- за виконанням коду;
- за рівнями тестування;
- за виконавцем тестування;
- за ступенем автоматизації;
- позитивне і негативне тестування;
- за знанням системи;
- за розробкою тестової документації.

Найчастіше під час тестування десктопних застосунків використовуються такі функціональні види тестування: функціональне тестування і тестування безпеки.

Функціональне тестування – це тип тестування програмного забезпечення, який перевіряє систему програмного забезпечення на відповідність функціональним вимогам або специфікаціям. Метою функціональних тестів є тестування кожної функції програмної платформи шляхом надання відповідних вхідних даних і перевірки вихідних даних на відповідність функціональним вимогам. Функціональне тестування в основному включає тестування чорного ящика, і воно не стосується вихідного коду програми. Це тестування перевіряє інтерфейс користувача, API, базу даних, безпеку, зв'язок між клієнтом і сервером та інші функції програми, що тестується. Тестування можна проводити як вручну, так і за допомогою автоматизації.

Тестування безпеки – це тип тестування програмного забезпечення, який виявляє вразливі місця, загрози, ризики в програмному застосунку та запобігає атаки зловмисників.

Метою тестів безпеки є виявлення всіх можливих слабких місць програмної системи, які можуть призвести до втрати інформації, доходу, репутації з боку співробітників або сторонніх осіб організації.

З нефункціональних видів тестів використовують: усі види тестування продуктивності, локалізації, інсталяційне, конфігураційне, сумісності, тестування на зручність використання, тестування інтерфейсу користувача,

тестування на відмову та відновлення, а також димове тестування. Особливу увагу звертають на продуктивність і тестування інсталяції.

Тестування продуктивності – це техніка нефункціонального тестування програмного забезпечення, яка визначає, наскільки стабільність, швидкість, масштабованість і швидкість реагування програми зберігаються при заданому робочому навантаженні. Це ключовий крок у забезпеченні якості програмного забезпечення, але, на жаль, часто розглядається ізольовано, і розпочинається після завершення функціонального тестування, а в більшості випадків – після того, як код буде готовий до випуску.

Тестування продуктивності включає оцінку вихідних даних програми, швидкості обробки, швидкості передачі даних, використання пропускну здатності мережі, максимальної кількості одночасних користувачів, використання пам'яті, ефективності робочого навантаження та часу відповіді на команди.

Тестування навантаження – перевіряє здатність програми працювати в умовах очікуваного навантаження користувача.

Стрес-тестування – передбачає тестування програми за екстремальних робочих навантажень, щоб побачити, як вона справляється з великим обсягом або обробкою даних.

Тестування на довговічність – проводиться, щоб переконатися, що програмне забезпечення може витримати очікуване навантаження протягом тривалого періоду часу.

Тестування стрибків – перевіряє реакцію програмного забезпечення на раптові великі стрибки навантаження, створювані користувачами.

Об'ємне тестування – перевіряє продуктивність програми в різних обсягах бази даних.

Тестування масштабованості – визначає ефективність програмного застосунку в «розширенні» для підтримки збільшення навантаження на користувачів.

Тестування локалізації – це техніка тестування програмного забезпечення, яка перевіряє, чи програма або вебсайт пропонує повну функціональність і зручність використання в певному регіоні. Якщо продукт налаштовано для цільової мови чи регіону, то перевірка локалізації перевіряє точність і придатність вмісту.

Тестування інсталяції виконується, щоб перевірити, чи програмне забезпечення правильно встановлено з усіма властивими функціями та чи продукт працює відповідно до очікувань. Воно відоме як тестування реалізації, виконується на останній фазі тестування перед першою взаємодією кінцевого користувача з продуктом.

Тестування конфігурації – це тип тестування програмного забезпечення, який перевіряє продуктивність системи, що розробляється, порівняно з різними комбінаціями програмного та апаратного забезпечення, щоб знайти найкращу конфігурацію, за якої система може працювати без будь-яких недоліків або проблем, відповідаючи функціональним вимогам (рис. 2).



Рисунок 2 – Процес конфігураційного тестування

Тестування GUI – це тип тестування програмного забезпечення, який перевіряє графічний інтерфейс користувача програмного забезпечення. Мета тестування графічного інтерфейсу користувача (GUI) полягає в тому, щоб переконатися, що функціональні можливості програмного забезпечення працюють відповідно до специфікацій, перевіряючи екрани та елементи керування.

Тестування відновлення – це техніка тестування програмного забезпечення, яка перевіряє здатність програмного забезпечення відновлюватися після збоїв, таких як програмного або апаратного забезпечення, мережі тощо. Мета тестування відновлення – визначити, чи можна продовжувати роботу програмного забезпечення після аварії або втрати цілісності. Тестування відновлення передбачає повернення програмного забезпечення до точки, коли була відома цілісність, а також повторну обробку транзакцій до точки збою.

#### Список літератури:

1. Кучеренко Є.І., Творошенко І.С. (2011) Оперативне оцінювання простору станів складних розподілених об'єктів з використанням нечіткої інтервальної логіки, *Штучний інтелект*, № 3, С. 382–387.
2. Кучеренко Е.И., Корниловский А.В., Творошенко И.С. (2010) О методах настройки функций принадлежности в нечетких системах, *Системы управления, навигации и связи*, Т. 1, № 13, С. 94–98.
3. Кучеренко Є.І., Творошенко І.С., Анопрієнко Т.В. (2016) Моделювання та оцінювання станів складних об'єктів із застосуванням формальної логіки, *Системи обробки інформації*, № 2, С. 76–82.
4. Tvoroshenko I.S. (2004) Structure and functions of intelligent decision-making tools in complex systems, *Artificial Intelligence*, № 4, С. 462–470.
5. Кучеренко Е.И., Творошенко И.С. (2003) Процессы принятия решений в сложных системах на основе нечетких интервальных представлений, *Вісник Національного технічного університету «ХПІ». Тематичний випуск: Системний аналіз, управління та інформаційні технології, Х.: НТУ «ХПІ»*, 1(7), С. 79–86.
6. Tvoroshenko I. (2019) Development of models of spatial analysis of status of interactive processes of complex systems.
7. Tvoroshenko I., and Tkachenko D. (2020) Mechanisms of image classification based on descriptors of local features, *Abstracts of IV International Scientific and*

*Practical Conference «Integration of scientific bases into practice» (October 12-16, 2020). Stockholm, Sweden, pp. 443–448.*

8. Tvoroshenko I., and Dziubenko M. (2020) Modern methods of analysis of the movement scheme using video detection of vehicles, *Abstracts of V International Scientific and Practical Conference «Study of modern problems of civilization» (October 19-23, 2020). Oslo, Norway, pp. 422–428.*

9. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785–1797.

10. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40–48.

11. Творошенко І.С., Табашник В.А. (2018) Розробка просторової моделі геоінформаційної підтримки людей з обмеженими можливостями, що пересуваються на інвалідних колясках, у місті Харків, *Збірник наукових праць Харківського національного університету Повітряних Сил*, 1(55), С. 122–128.

12. Tvoroshenko I., and Zarivchatskyi R. (2020) Analysis of existing methods for searching object in the video stream, *Abstracts of VI International Scientific and Practical Conference «About the problems of science and practice, tasks and ways to solve them» (October 26-30, 2020). Milan, Italy, pp. 500–505.*

13. M. Ayaz Ahmad, Irina Tvoroshenko, Jalal Hasan Baker, Liubov Kochura, and Vyacheslav Lyashenko (2020) Interactive Geoinformation Three-Dimensional Model of a Landscape Park Using Geoinformatics Tools, *International Journal on Advanced Science, Engineering and Information Technology*, 10(5), pp. 2005–2013.

14. Творошенко І.С. (2018) Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем, *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland*, pp. 118–121.

15. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069–6084.

16. Творошенко І.С., Дехтярь А.П. (2005) Информационные технологии в задачах компьютерной диагностики с использованием интеллектуальных систем. *Клиническая информатика и Телемедицина. Компьютерная Медицина–2005: материалы междунар. научн.-технич. конф., Харьков, р. 138.*

17. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.

18. Кучеренко Е.И., Творошенко И.С. (2010) Прикладные аспекты моделирования нечетких процессов в сложных системах, *Збірник наукових праць Харківського університету Повітряних сил*, 1(123), С. 127–131.

19. Кучеренко Е.И., Филатов В.А., Творошенко И.С., Байдан Р.Н. (2005) Интеллектуальные технологии в задачах принятия решений технологических комплексов на основе нечеткой интервальной логики, *Восточно-Европейский журнал передовых технологий*, № 2. С. 92–96.



# CERTIFICATE



INTERNATIONAL  
SCIENCE GROUP

is awarded to



ПОЛУБЕХІН АНТОН

for active participation

XXXVI International Scientific and Practical Conference  
“THE MAIN PROSPECTS FOR THE DEVELOPMENT  
OF SCIENCE IN MODERN LIFE”  
13-16 September 2022, Warsaw, Poland  
24 Hours of Participation  
(0,8 ECTS credits)

Organizing committee



Ekaterina Zvereva