

Іншим важливим аспектом при оцінці ефективності реклами у соціальних мережах є можливість застосувати до інформації про дані користувачів методів BigData для побудови закономірностей поведінки користувачів [4].

Тут варто сказати про те, що між базами користувачів і рекламною аудиторією кожної з платформ може спостерігатися значне збігання. Тому рекламодавець, реалізуючи рекламну кампанію засобами соціальних мереж, може їх поєднувати.

#### Список використаних джерел

1. Воронько-Невіднича Т. В., Калюжна Ю. П., Хурдей В. Д. Реклама і рекламна діяльність : навч. посіб. Полтава, РВВ ПДАА, 2018. 230 с.
2. "The world's most used social platforms 2022" [Електронний ресурс]. Режим доступу: <https://www.hootsuite.com/resources/digital-trends>
3. "What Is an Advertising Campaign? Definition, Strategy, and Examples," Cyberclick. [Електронний ресурс]. Режим доступу: <https://www.cyberclick.net/advertising>
4. Casaca J. Marketing in the Era of Big Data [Електронний ресурс]. Режим доступу: [https://www.academia.edu/34354137/Marketing\\_in\\_the\\_Era\\_of\\_Big\\_Data](https://www.academia.edu/34354137/Marketing_in_the_Era_of_Big_Data)

УДК 004.428 : 004.514

Баркалов О. О., д.т.н., професор,  
професор кафедри інформаційних  
технологій

Бабаков Р. М., д.т.н., доцент,  
доцент кафедри інформаційних  
технологій

### ДОСЛІДЖЕННЯ ПРОДУКТИВНОСТІ КРОСПЛАТФОРМНИХ БІБЛІОТЕК ДЛЯ ПОБУДОВИ ГРАФІЧНОГО КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Наявність в сучасному комп'ютерному світі кількох домінуючих операційних систем (ОС) вимагає розробників програмного забезпечення (ПЗ) адаптувати власні продукти до різних ОС та їх версій. Це потребує від розробників ПЗ додаткових витрат ресурсів і часу, що, в свою чергу, відбивається на кінцевих користувачах. Одним зі шляхів вирішення цієї проблеми є розробка і використання кросплатформного ПЗ [1]. Це дозволяє розробникам зосередитись на одній версії програмного продукту, але вимагає використання кросплатформних мов програмування.

До кросплатформних мов програмування належить мова Python. Її значне поширення в останні роки обумовлене, в тому числі, наявністю великої кількості бібліотек класів (так званих модулів), використання яких переважно є безкоштовним. Так, для вирішення задач проєктування графічних користувацьких інтерфейсів (GUI) програміст має вибір з цілого ряду бібліотек, що пропонують приблизно однакові можливості: Tkinter, Flexx, Dabo, Kivy, PyQt, PySide, wxPython, PyGUI, та інші (усього більше 30) [2]. Оскільки більшість програмних продуктів мають графічний користувацький інтерфейс, перед Python-розробником постає питання вибору відповідної бібліотеки. Розгляду саме цього питання присвячена дана робота.

Недоліком Python, як відомо, є відносно низька швидкодія, що є наслідком ряду особливостей даної мови програмування, зокрема, динамічної типізації даних. Питання швидкодії розроблюваних Python-додатків часто буває критичним і обмежує їхню область застосування [3]. Загальна низька швидкодія мови Python, безумовно, має вплив на швидкість роботи графічного інтерфейсу. Використання вікон з великою кількістю різноманітних віджетів, а також багатовіконного інтерфейсу може забирати відчутну частину і так невисокої продуктивності Python-додатку. З огляду на це розробнику ПЗ доцільно обирати для побудови GUI таку бібліотеку, яка у порівнянні з іншими бібліотеками забезпечує найбільшу швидкодію користувацького інтерфейсу.

Метою даної роботи є дослідження швидкодії різних бібліотек, призначених для побудови графічного користувацького інтерфейсу додатків, написаних мовою Python. Отримані результати можуть бути використані розробниками ПЗ при виборі GUI-бібліотеки з огляду на її відносну продуктивність.

Пропонована авторами методика дослідження полягає в наступному.

Як відомо, графічний користувацький інтерфейс складається із певного набору віджетів. Хоча кожна GUI-бібліотека містить свій набір віджетів, можна виділити певний набір віджетів, загальних для більшості GUI-бібліотек. Найбільш поширеними, на думку авторів, є наступні п'ять віджетів: кнопка, радіокнопка (забезпечує вибір одного із декількох пунктів), позначка (може бути ввімкнена або вимкнута), мітка (область з текстовим написом), однорядкове текстове поле.

На створення кожного окремого віджету GUI-бібліотека витрачає певний час. Було обрано наступну ознаку продуктивності: чим більше часу витрачається на створення окремого віджету, тим менш продуктивною є відповідна бібліотека. Оскільки час на створення одного екземпляру віджета є надто малим, було розроблено додаток, в якому в циклі створюється велика кількість однакових віджетів (від 500 і більше). Усі віджети створюються і розташовуються в межах одного GUI-вікна, хоча й не можуть бути відображені на екрані одночасно. Час, що витрачається на даний процес, залежить від багатьох факторів, але в цілому він демонструє ефективність роботи GUI-бібліотеки з даним віджетом. При цьому нескладно виміряти час створення графічного вікна додатку, для чого можуть бути використані функції Python-бібліотеки Time.

Суть проведених експериментів полягала в наступному:

1. Для кожного з п'яти наведених вище типів віджетів визначався час створення графічного вікна з великою кількістю віджетів (500, 1000, 1500).

2. Експеримент повторювався декілька разів для визначення середнього часу створення вікна програми (з метою зменшення впливу зовнішніх факторів).

3. Експеримент проводився для трьох GUI-бібліотек, що містять аналогічні віджети: бібліотеки PyQt, бібліотеки wxPython та стандартної бібліотеки Tkinter.

Були використані наступні умови проведення експерименту: персональний комп'ютер з процесором i5-4670k, 16 Гб оперативної пам'яті, ОС Windows 10, Python 3.8.5. Узагальнені результати експериментів наведені в табл. 1, що містить витрати часу на створення різної кількості віджетів за допомогою кожної з обраних GUI-бібліотек. Відзначимо, що для віджета «однорядкове текстове поле» засобами бібліотеки wxPython не вдалося створити зазначену кількість віджетів, внаслідок чого відповідні результати в табл. 1 відсутні.

Аналіз таблиці 1 дозволяє зробити наступні висновки:

1. Бібліотека Tkinter має найменші показники витрат часу, тож серед досліджених бібліотек має найбільшу швидкодію. Недоліком Tkinter, на думку багатьох розробників GUI-додатків, є трохи застарілий дизайн віджетів, що однак жодним чином не впливає на їх функціональність.

2. Бібліотека wxPython має найбільші витрати часу. При використанні 1500 віджетів витрати часу зростають багаторазово, що може говорити про непристосованість даної бібліотеки до роботи з великою кількістю віджетів.

3. Бібліотека PyQt має проміжні результати витрат часу, що характеризує її як бібліотеку із середньою продуктивністю серед досліджених бібліотек. При цьому перевагою PyQt є наявність великої кількості різноманітних віджетів і їх сучасний дизайн.

На рис. 1 графічно показаний вміст табл. 1 для 1000 віджетів.

Таблиця 1 – Результати експериментальних досліджень

Тип віджету	Кількість створюваних віджетів	Середній час створення вікна програми, мс		
		Бібліотека Tkinter	Бібліотека wxPython	Бібліотека PyQt5
Кнопка	500	0,206	0,608	0,423
	1000	0,246	1,467	0,438
	1500	0,282	36,246	0,468
Радіокнопка	500	0,201	0,484	0,405
	1000	0,245	0,952	0,437
	1500	0,293	22,481	0,483
Мітка	500	0,188	0,534	0,407
	1000	0,235	1,030	0,470
	1500	0,272	22,702	0,500
Позначка	500	0,203	0,483	0,406
	1000	0,249	0,900	0,452

	1500	0,296	22,368	0,468
Однорядкове текстове поле	500	0,187	—	0,453
	1000	0,203	—	0,531
	1500	0,243	—	0,609

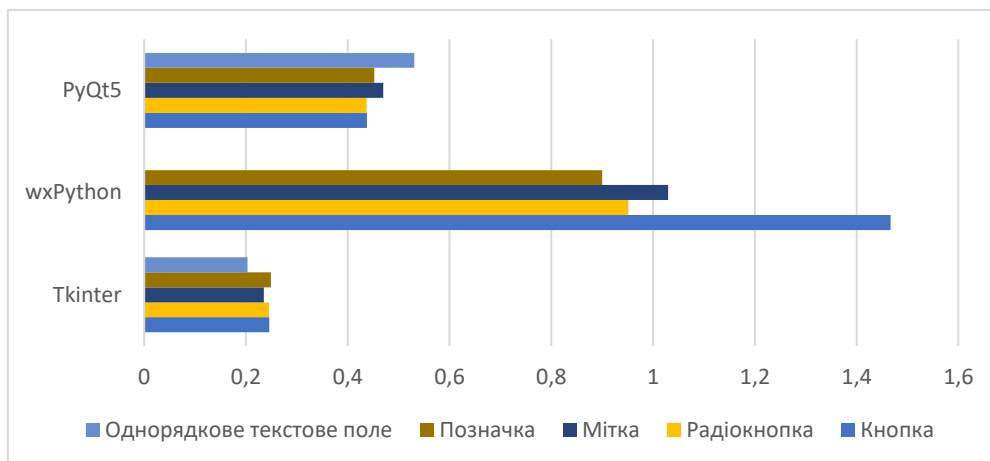


Рисунок 1 – Графічне представлення вмісту таблиці 1 для 1000 віджетів

Отримані авторами результати, безумовно, не є всеохоплюючими і вичерпними. Дослідження інших віджетів, інших бібліотек та їх версій під різними ОС дозволить отримати більш повну картину та спростити розробнику ПЗ вибір GUI-бібліотеки за критерієм швидкодії графічного користувацького інтерфейсу.

### Список використаних джерел

1. Крохмалюк В.В., Римар П.В. Використання крос-платформного програмного інтерфейсу OpenGL в комп'ютерній графіці Матеріали 2-ї всеукраїнської науково-практичної конференції Комп'ютерні технології обробки даних 2021 URL: <https://jktod.donnu.edu.ua/article/view/11654>
2. Alan D. Moore. *Mastering GUI Programming with Python*. – Birmingham: Packt Publishing Ltd, 2019. – 509 p.
3. Васильєв О. *Програмування мовою Python*. – Тернопіль: «Навчальна книга – Богдан», 2019. – 504 с.