



Ірина БЛІНДАР,
вчителька інформатики Грузевицького ліцею
Чорноострівської селищної ради

STEM-освіта та проєктна технологія: комплексна реалізація математичних концепцій засобами мови програмування Python

У сучасну інформаційну епоху, що характеризується стрімким нагромадженням великих обсягів цифрових даних, здатність до коректної обробки та представлення інформації стає базовою компетентністю. Візуалізація даних сьогодні виступає не просто допоміжним елементом, а критично важливим інструментом для виявлення прихованих закономірностей, які важко зафіксувати у текстовому чи табличному форматах.

Метою цих методичних рекомендацій є надання педагогам концептуальної та практичної підтримки у виборі інструментарію для проєктування програмних засобів обробки та візуалізації математичних даних. Використання Python у STEM-освіті дозволяє реалізувати кросплатформові рішення, що забезпечують автоматизацію обчислень та високу наочність, необхідну для ухвалення точних і надійних рішень у навчальному та дослідницькому процесах.

Ефективність впровадження графічних методів навчання обґрунтована особливостями опрацювання інформації людським мозком.

Ключові факти про візуальне сприйняття:

- Домінування зорового каналу: людина сприймає близько 90% всієї інформації через зір.
- Ефективність праці: використання візуальних засобів підвищує продуктивність роботи в середньому на 17%.
- Швидкість обробки: мозок ідентифікує та опрацьовує візуальні образи у 60 000 разів швидше, ніж текстові структури. Це дозволяє аудиторії миттєво вловлювати суть алгоритмів та закономірностей.

Статистика зростання візуального компонента (1985–2007 рр.):

- необхідність адаптації методики викладання до потреб "цифрового покоління" підтверджується вибуховим зростанням наочності в медіа-просторі: кількість візуальних елементів у друкованих виданнях (газетах) зросла на 142%;
- обсяг ілюстрацій у науковій та навчальній літературі збільшився на 400%;
- в інформаційному просторі мережі Інтернет з 2007

року обсяг візуального контенту продемонстрував шалений ріст — на 9900%.

Побудова графіків є важливим етапом переходу від абстрактних математичних залежностей до їх візуального представлення. Доцільно виокремити шість ключових аспектів, що визначають значущість цього процесу:

1. Аналіз властивостей функцій: дослідження форми, періодичності та точок екстремуму сприяє глибшому розумінню поведінки математичних об'єктів.
2. Пошук розв'язків рівнянь: графічні методи забезпечують візуальне визначення точок перетину, що дає змогу знаходити розв'язки рівнянь і їх систем.
3. Моделювання фізичних явищ: візуалізація є ефективним засобом відтворення траєкторій руху тіл, динаміки систем і фізичних полів у задачах фізики та інженерії.
4. Виявлення аномалій та помилок: графічне подання дає змогу оперативно ідентифікувати нетипові значення (викиди) у наборах даних.
5. Розвиток графічної інтуїції: педагогічна цінність полягає у формуванні здатності прогнозувати результати зміни параметрів. Наприклад, зміна коефіцієнта у лінійній функції ілюструє варіацію кута нахилу прямої, що сприяє встановленню зв'язку між числовими параметрами та геометричною інтерпретацією.
6. Оптимізація комунікації результатів: використання графіків зменшує комунікаційні бар'єри, спрощуючи інтерпретацію та передачу складних ідей.

У сучасному освітньому просторі вибір мови програмування для реалізації проєктів має вирішальне значення. Серед різноманіття технологій Python впевнено посідає перше місце як найбільш адаптивний інструмент для учнів. Python не обмежується лише інформатикою, його можна інтегрувати в будь-який шкільний предмет: моделювання процесів, розрахунки та побудова графіків за допомогою готових бібліотек.

Для розробки навчальних проєктів рекомендується поєднання стандартних засобів GUI та спеціалізованих математичних пакетів.

Бібліотека	Призначення	Методичні особливості та обмеження
Tkinter	Створення графічного інтерфейсу (GUI)	Стандартна бібліотека для побудови вікон, кнопок та полів введення. Проста в освоєнні, але обмежена в аналітиці
Matplotlib	Професійна візуалізація даних	Пакет для 2D та 3D графіки. Забезпечує автоматичну побудову сіток, осей та масштабування

Важливим зауваженням щодо модуля Turtle (Черепаха) є те, що попри популярність у початковому навчанні, модуль Turtle є недостатнім для наукової візуалізації. Його ключові слабкі місця – неможливість автоматизованої побудови точної координатної сітки, складність у розстановці цифрових позначок на осях та відсутність динамічного масштабування. Перехід до Matplotlib дозволяє автоматизувати ці процеси за допомогою функцій `plt.grid()` та автоматичних маркерів осей.

Функціональні можливості Matplotlib: бібліотека підтримує 9 основних типів візуалізації:

- лінійні графіки (line plot);
- діаграми розсіювання (scatter plot);
- стовпчасті діаграми (bar chart);
- гістограми (histogram);
- секторні (кругові) діаграми (pie chart);
- дискретні графіки (stem plot);
- контурні графіки (contour plot);
- векторні поля (quiver);
- спектрограми (spectrogram).



Розроблення проекту доцільно організувати як процес інженерного проєктування, що складається з двох основних етапів:

1. Проєктування графічного інтерфейсу користувача (GUI): створення візуальної форми (вікна), а також раціональне розміщення елементів інтерфейсу (кнопок, полів введення `tk.Text`, міток) з метою забезпечення зручної та ефективної взаємодії з користувачем.

2. Реалізація програмної логіки та обчислень: розроблення функцій оброблення даних, а також забезпечення відображення результатів шляхом виклику відповідних вікон візуалізації.

У процесі розроблення зазначених проєктів досягаються такі навчальні цілі та формуються відповідні компетентності:

- Опанування принципів об'єктно-орієнтованого програмування (ООП): засвоєння концепцій інкапсуляції логіки інтерфейсу в межах класу `Main (Frame)`, а також набуття навичок роботи з об'єктами та механізмами спадкування.
- Оброблення вхідних даних: перетворення рядкових

даних, отриманих із текстових полів, у числові структури даних (зокрема типу `float`).

- Математичне моделювання: алгоритмічна реалізація математичних залежностей (зокрема, лінійних і квадратичних функцій).

Чек-лист для вчителя (підготовка та реалізація):

- Налаштування середовища розроблення: використання середовища `IDLE` для розв'язання базових завдань або `Visual Studio` — для реалізації складніших проєктів.
- Створення та активація віртуального середовища (`venv`): забезпечення ізоляції залежностей проєкту.
- Встановлення сторонніх бібліотек: використання менеджера пакетів `pip` для керування залежностями.
- Робота з методами вбудованих типів даних: застосування методів (наприклад, `.split()`, `.get()`) для оброблення даних.
- Оброблення винятків: використання конструкцій `try-except` для оброблення помилок введення.
- Тестування кросплатформності коду: перевірка коректності роботи програмного забезпечення на різних операційних системах.

Використання Python разом із бібліотеками `Tkinter` та `Matplotlib` дозволяє створювати потужні дидактичні інструменти, які є маловимогливими до системних ресурсів та кросплатформовими. Такі проєкти стимулюють перехід від "процедурного малювання" (як у `Turtle`) до наукової візуалізації на основі реальних даних.

Шляхи розвитку проєкту для учнів:

- інтеграція тригонометричних, степеневих та логарифмічних функцій;
- розв'язання проблеми блокування інтерфейсу (побудова графіків у окремих незалежних вікнах без зупинки головного циклу програми);
- додавання функціоналу збереження отриманих графіків у форматі `PNG/PDF`.

Використані джерела

1. Бежанова М. М., Москвіна Л. А., Поттосін І. В. Практичне програмування: структури даних і алгоритми: навч. для вузів. — М.: Логос, 2001. — 223 с.
2. Гловацька А. П. Методи та алгоритми обчислювальної математики: навч. посібник для вузів. — М.: Радіо і зв'язок, 1999. — 408 с.
3. Кнут Д. Е. Мистецтво програмування: навч. посібник: у 3 т. Т. 1: Основні алгоритми. — 3-є вид. — М.: Вільямс, 2000. — 720 с.
4. `Beginner'sGuidetoPython` [Електронний ресурс]. — Режим доступу: <https://wiki.python.org/moin/BeginnersGuide> (дата звернення: 04.02.2024).
5. `Matplotlib: VisualizationwithPython` [Електронний ресурс]. — Режим доступу: <https://matplotlib.org/> (дата звернення: 04.02.2024).
6. `ThePython Standard Library` [Електронний ресурс]. — Режим доступу: <https://docs.python.org/3/library/index.html> (дата звернення: 01.03.2024).
7. `Why your Brain Craves Infographics` [Електронний ресурс]. — Режим доступу: <https://neomam.com/interactive/13reasons/> (дата звер-