

УДК 004

Букатка Ю. С.  
аспірант, Івано-Франківський національний технічний університет  
нафти і газу, м. Івано-Франківськ, Україна

Копей В. Б.  
д-р техн. наук, професор, Івано-Франківський національний  
технічний університет нафти і газу,  
м. Івано-Франківськ, Україна

### **ДОСВІД ВИКОРИСТАННЯ PYTHON ТА ЇЇ ПАКЕТІВ ДЛЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ЖИТТЄВОГО ЦИКЛУ ВИРОБУ**

Python сьогодні є найбільш популярною і перспективною мовою прикладного програмування загального призначення. Основна її особливість – це простота розуміння і складання програмного коду. Це дало поштовх бурхливому розвитку пакетів для розв'язування різних прикладних задач, у тому числі задач інформаційної підтримки життєвого циклу (ЖЦ) виробу. З використанням вільних Python-пакетів сьогодні є можливим розроблення якісної PLM-системи для підтримки етапів проектування, виготовлення та експлуатації виробу. Проекти [1-7] свідчать про можливість простого об'єднання різнотипних пакетів в єдину ефективну PLM-систему. Як правило базовими пакетами в таких системах повинні бути пакети для роботи з масивами, чисельними методами і візуалізацією даних (NumPy, SciPy, Matplotlib). Для забезпечення автоматизованого проектування виробу (CAD, CAM) можуть бути використані пакети PythonOCC, FreeCAD, CadQuery, які ґрунтуються на потужному вільному геометричному ядрі OpenCASCADE [2, 3]. Ці пакети можуть бути доповнені пакетами FEA (PyCalculix, scikit-fem, Gmsh), симуляції динаміки жорстких тіл (PyBullet, PyODE), моделювання і оптимізації складних мультифізичних динамічних систем (Gekko, OMPython, rumpso) [2-4]. Аналітичні задачі можуть бути розв'язані за допомогою пакету символічної математики SymPy [4, 5]. Його використовують такі спеціальні пакети як Orptlang (оптимізаційні задачі), SimuPy (динамічні системи), PyDu (динаміка багатьох тіл), Scikit-fdiff (метод скінченних різниць), SfePy (метод скінченних елементів). В проекті [4] SymPy використано для геометричного моделювання гвинтових ліній різнотипних різьбових з'єднань. На основі SymPy та SciPy розроблено Python-пакет для компонентно-орієнтованого моделювання складних динамічних систем [1]. Інтерактивна робота з PLM-системою може бути забезпечена засобами: візуалізації

## Підтримка та організація виробництва

тривимірних даних (pyglet, VTK, Mayavi, PyVista), створення графічного інтерфейсу користувача (Tkinter, wxPython, PyQt) та інтерактивних документів (IPython, Jupyter, Google Colab, JupyterLite). Jupyter є основою інтерфейсу користувача в PLM-системі різьбових з'єднань [4]. На етапах виробництва і експлуатації виробу можуть бути використані системи управління виробничими процесами, CNC-системи, системи автоматизованого технічного контролю та SCADA-системи, які основані на мікроконтролерах та використовують пакети PySerial, PyFirmata, PyScada, інтерпретатор Місроpython [4, 7]. На різних етапах ЖЦ виробу розв'язуються складні задачі, які потребують застосування методів системного аналізу, штучного інтелекту, інженерії знань, аналізу даних та машинного навчання. Для цього можна використати пакети pydatalog, python-constraint, NetworkX, Pandas, scikit-learn, keras та інші [3, 5, 6]. Ефективна робота багатьох компонентів PLM-системи потребує інтеграції в ній засобів паралельних обчислень (Dask, Celery, Ray). На основі концепції акторів та пакету Ray була забезпечена паралельна робота компонентів мультиагентної PLM-системи [4]. Досягнення ефекту синергії потребує не простого поєднання різнотипних пакетів, а створення їхньої системи з оптимальними зв'язками. Для цього слід опиратись на теоретико-системні принципи, зокрема принципи самоорганізації складних систем. Це було враховано під час розроблення мультиагентної PLM-системи [4].

### **Список літератури**

1. Pycodyn. URL: <https://github.com/vkopey/pycodyn> (accessed 9.11.24).
2. ThreadsOCC. URL: <https://github.com/vkopey/ThreadsOCC> (accessed 9.11.24).
3. ThreadsPLM. URL: <https://github.com/vkopey/ThreadsPLM> (accessed 9.11.24).
4. ThreadsPLM-MAS-. URL: <https://github.com/vkopey/ThreadsPLM-MAS-> (accessed 9.11.24).
5. ThreadES. URL: <https://github.com/vkopey/ThreadES> (accessed 9.11.24).
6. IRclassify. URL: <https://github.com/vkopey/IRclassify> (accessed 9.11.24).
7. CNC3018Pro-and-Robot-Arm. URL: <https://github.com/vkopey/CNC3018Pro-and-Robot-Arm> (accessed 9.11.24).