

ISSN 2518-7325 (Online), ISSN 2306-1677 (Print)
Land Management, Cadastre and Land Monitoring
Received: 06.05.2026; Accepted: 02.06.2026; Published: 30.06.2026;
<http://dx.doi.org/10.31548/zemleustriy2026.02.01>

Copyright © The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (<https://creativecommons.org/licenses/by-nc-sa/4.0/>)

UDC 004.424.22:528.235

APPLICATION OF THE NESTED ITERATION METHOD FOR AUTOMATED MAPPING

Y. Dorosh, *Doctor of Economic Sciences, Professor, Academician of the NAAS of Ukraine*

E-mail: doroshjosyp@gmail.com

ORCID: 0000-0002-1764-6188

Land Management Institute of the National Academy of Agrarian Sciences of Ukraine

B. Zaiachkivska, *PhD in Economics*

E-mail: b_zayachkivska@nubip.edu.ua

ORCID: 0009-0003-0962-5998

National University of Life and Environmental Sciences of Ukraine

R. Kharytonenko, *PhD in Economics*

E-mail: kharytonenkor@gmail.com

ORCID: 0000-0002-7992-3339

Land Management Institute of the National Academy of Agrarian Sciences of Ukraine

D. Melnyk, *PhD in Economics*

E-mail: melnykdenys@gmail.com

ORCID: 0000-0002-3714-9592

Land Management Institute of the National Academy of Agrarian Sciences of Ukraine

Annotation. *The article considers a method for automating the creation of a series of cartographic materials based on the integration of QGIS Atlas tools and the Python programming language. The study aims to optimize the preparation process of multi-page thematic atlases for large spatial datasets. To achieve the set objective, the authors' approach to integrating nested iteration algorithms has been applied, which allows for flexibly adapting spatial boundaries and individual visual display parameters for each element of the system. A key element is the developed algorithm based on the processing of the featureChanged event, which ensures automatic synchronization of district attributes with the corresponding .qml style files. The model was tested on a dataset exceeding 20,000 contours.*

The testing results confirmed the high efficiency of the method. The speed of map series generation increased approximately 20-fold compared to traditional methods, while completely eliminating subjective visualization errors. The proposed approach represents a universal methodological platform for the automated transformation of complex spatial datasets into high-precision graphical models. This solution provides the capability for rapid presentation of multifactor research results in any field requiring a combination of high-intensity processing of large datasets with adherence to individual visual interpretation parameters for each unique system element.

Keywords: *automated mapping, QGIS Atlas, PyQGIS, agrochemical monitoring, GIS automation, nested iteration method, dynamic styling, thematic atlases, Python API, *.qml style files, layout automation, mapping algorithmization.*

Problem Statement. Thematic mapping is an essential tool for meeting the needs of public administration, defense, spatial planning, and environmental monitoring [1]. It involves the representation of objects and phenomena in specific sectors (economics, ecology, agriculture, etc.), requiring high precision and regular data updates. The traditional creation of thematic maps for a large number of administrative units (districts, communities, settlements, agricultural fields) is a routine process involving

significant time costs. Each map requires individual adjustments for layer filtering, zooming, and the application of appropriate color scales.

QGIS, an open-source platform, is currently one of the most popular geographic information systems due to its free accessibility and constant updates [2]. This software is widely used in land management, agronomy, forestry, water resource management, ecology, urban planning, and public administration for processing large spatial datasets. Due to its cross-platform nature and support for the high-level programming language Python, QGIS has become a popular tool for generating complex cartographic reports, facilitating not only visualization but also the automation of cyclical tasks in land resource monitoring.

The choice of the QGIS software environment for implementing the nested iteration method is based on the findings of S. Steiniger and A. Hunter regarding the functional capabilities of free software. The authors note that modern open-source GIS platforms provide the necessary flexibility for solving specific geoprocessing tasks, allowing the integration of complex automation algorithms without significant capital investment [3]. The necessity of automating the cartographic process is confirmed by current trends in Digital Soil Mapping [4]. The application of intelligent machine learning models to predict the spatial variability of soil characteristics, such as organic carbon content and pH levels, requires adequate dynamic visualization tools [5]. According to R. Espinel et al., the use of artificial intelligence and automated algorithms significantly reduces the time required for spatial data processing while ensuring high efficiency in managerial decision-making [6].

The implementation of automated map generation algorithms ensures the application of defined templates, providing the ability to create thousands of thematic plans in a matter of hours—a task physically impossible with manual layout composition. This enhances the quality of decision-making, as institutions can obtain up-to-date atlases in real-time, allowing for an immediate response to negative changes in the state of land resources. Therefore, the use of GIS tools in land management in Ukraine is not

merely a technical improvement but a fundamental mechanism for the transformation of the entire industry [7]. The development of GIS technologies enables domestic institutions to meet international data management standards and ensure transparency in territorial monitoring.

Analysis of recent research and publications. B. Singh confirms that GIS is utilized by ministries (e.g., USDA) for crop protection, addressing economic issues, and soil condition research. Agricultural GIS is capable of transforming vast amounts of data into visual maps to support decision-making, making QGIS an indispensable tool for monitoring soil cover and agricultural crops due to its technical capabilities and accessibility. Singh evaluates data accuracy at the 90th percentile level; however, the process of visualizing such large-scale datasets remains labor-intensive, necessitating the implementation of automated dynamic mapping algorithms [8].

Most existing works focus either on QGIS Atlas without Python or on full map generation via Python without utilizing the QGIS interface. For instance, Anita Graser describes automation through the use of layouts and Atlas as a fundamental component of serial mapping, but points out its limitations, as the software module itself does not provide for changing style files (.qml) for each individual object [9].

According to G. Sherman, the combination of QGIS and Python (PyQGIS) allows for the expansion of open-source GIS toolsets by creating scripts to perform complex automated tasks that are impossible to implement using standard interface tools. The author regards Python as the primary tool for plugin development and API expansion [10]. Consequently, most automation examples concern data processing in the console, whereas the nested iteration method integrates Python scripts directly into the cycle of the serial layout manager (Atlas) into a single resulting file.

G. Sherman justifies the concept of expanding the basic GIS toolkit by integrating Python 3 with the QGIS API to perform complex automated tasks. The nested iteration method is based on the principle of hierarchical object management within a project. Specifically, when faced with the limitations of direct manipulation of renderer

parameters (QgsRuleBasedRenderer), which often leads to attribute access errors (setClassAttribute), a "peripheral layer management" strategy was applied.

This approach, according to Sherman's research, involves addressing higher-level objects—the Layer Tree and the Layout API. Instead of attempting destructive modification of the internal renderer structure, the algorithm implements: the use of the loadNamedStyle method to load pre-configured .qml files, ensuring visualization stability; the application of the setSubsetString method for direct SQL data filtering at the provider level, allowing for the correct display of only relevant objects within the current atlas frame; and the use of the atlas.featureChanged signal as a trigger to initialize the map update cycle, which is a classic PyQGIS architecture technique for creating complex dynamic systems.

K. Menke, in the context of creating layouts and automation via Atlas, emphasizes the importance of automation for professional cartography to avoid repetitive manual actions [11].

Thus, while most authors focus on static templates, the nested iteration method using PyQGIS implements the automated interception of page-change events for the instantaneous loading of unique style files (.qml) and rigorous attribute filtering. This allows not only for map scaling but for a complete transformation of its analytical content (legend and color scale) without operator intervention.

Purpose of the study: The purpose of the study is to optimize the generation processes of multi-page thematic atlases by implementing automated styling algorithms and iterative geodata processing within the QGIS environment.

Materials and methods of research. Achieving the set objective is ensured by implementing the approach developed by the authors to organizing nested iterations, which is based on a hierarchical two-level architecture for managing the mapping process. At the first level, the primary iteration, performed via the QGIS Atlas module, ensures the sequential processing of spatial objects in the coverage layer, providing automatic scaling, geometry binding, title generation, and atlas page numbering. At the

second level, the secondary iteration is implemented via the Python API (PyQGIS) and is responsible for the semantic management of the cartographic display: dynamic loading of style files (.qml) using the loadNamedStyle method, rigorous filtering of spatial objects at the data provider level using the setSubsetString method with SQL queries, and real-time operational management of mask layers.

Synchronization between the iteration levels is ensured by processing the atlas.featureChanged event, which acts as a trigger to initialize the cartographic display update cycle during transitions between objects. This approach implements the principle of "external layer management," in which the modification of stylistic parameters is carried out by addressing higher-level objects – the Layer Tree and the Layout API. The method was tested on a dataset exceeding 20,000 contours, utilizing a comparative analysis of time costs between manual and automated methods for preparing thematic map series. The efficiency evaluation was performed using a model example of atlas creation.

Presentation of the main research material. Implementing the nested iteration method requires the preparation of input components: styles, visualization data layers, and spatial extent boundaries. One of the most time-consuming tasks is forming the set of QGIS style files (.qml). Unlike static design, these files contain conditional formatting based on the logic of categorizing indicators into classes or groups (e.g., slope steepness, soil quality, crop types, degradation types, suitability levels, etc.), configured color ramps, boundary thickness, object transparency, and rendering rules that allow the script to instantaneously change the layer's appearance when transitioning from one indicator to another.

The attribute-spatial data layers—comprising field contours with geometry and coordinate referencing—encompass the dataset intended for visualization. Each field in the attribute table of the spatial layers can be utilized to visualize specific values. It is assumed that these tables contain a key field for linking with the territorial boundaries.

The territorial boundaries (Coverage Layers) serve as the atlas management layer (e.g., boundaries of administrative districts, settlements, agro-hydrological regions, etc.), providing a spatial filter for geographic objects. Each geometric unit within this layer becomes an individual atlas page. Simultaneously, through spatial overlay, the layer acts as a mask or boundary that defines which specific objects from the primary dataset should be included in the current layout.

By utilizing the input data, the PyQGIS-based script ensures their synergy:

- a district boundary is selected (overlay layer);
- contours are filtered according to the selected boundary (data layer);
- the corresponding classification scale (style file) is applied to the contours designated for visualization.

This approach transforms a process that previously required weeks of manual labor into an automated cycle, the duration of which is limited only by the PDF rendering speed. At the same time, this method makes the automated mapping system dynamic and editable, allowing for the addition of new characteristics to the spatial layer and new styles even after the atlas generation has been configured. To achieve this, one simply needs to add a new style file to the folder without modifying the core program code.

Once the input data is ready, an atlas layout is created, which includes: a title page, a contributors' page, a table of contents, and thematic maps. Cartographic materials are designed according to standard requirements; specifically, the page template includes the map title, a north arrow, the map frame, a legend, a scale bar, page numbering, etc.

Level 3 represents the most critical block, as the Python controller and the proposed nested iteration approach (Processing) provide the core logic and automation for creating and dynamically updating the atlas layout. The implementation of the algorithm is based on the event-driven architecture of PyQGIS, where a system signal indicating a change in the active layout object serves as the trigger for code execution. The script intercepts the event of the Atlas module transitioning to a new element of the coverage layer and automatically initiates a nested procedure, which reads the attribute

data of the current district, performs SQL filtering, and dynamically adapts the graphical styles. At each iteration step, the algorithm first reads the attribute name of the current district, which serves as a key identifier for further semantic processing.

Utilizing the principles of nested logic, the system automatically accesses external storage to locate the corresponding style file in .qml format, the name of which correlates with the current indicator and district (e.g., district_1_slope.qml). Upon identifying the required file, the program initiates the loadNamedStyle() method, which performs an instantaneous update of the symbols and color scales in the project without the need to reload layers. Simultaneously, the setSubsetString() method is applied to perform data filtering via SQL queries, allowing for the localization and display of only those geometries (contours) that spatially belong within the current district boundary. The final stage of each iteration is the forced update of dynamic layout elements, such as text headers and legend content, after which the generated page is sent for rendering, and the cycle automatically proceeds to the next object. This approach ensures that the visual model of the map content is fully identical to the actual state of the input data for each unique mapping unit.

In the final stage of the system's operation, an automated rendering and export process is initiated, converting the generated cartographic materials into high-resolution digital formats, such as PDF or multi-page raster images. The software consolidates all pre-configured dynamic parameters, ensuring the correct visualization of geometries and their compliance with project metadata (Fig. 1).

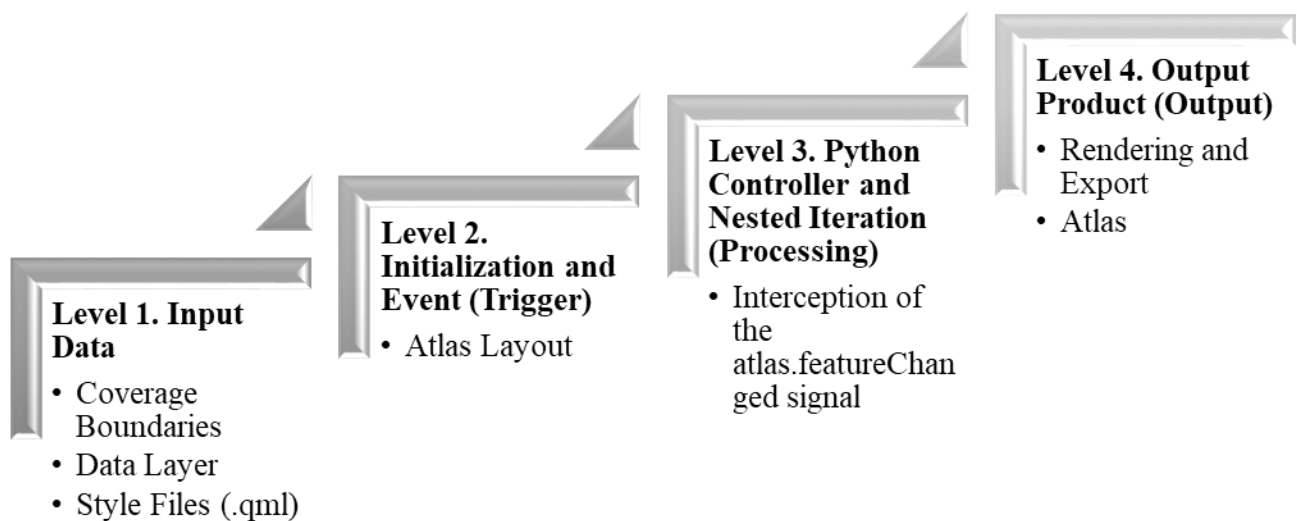


Fig. 1. Conceptual model of automated dynamic mapping using the nested iteration method (via PyQGIS)

The final result of this process is a digital atlas presented as an ordered series of maps. Each page of such an atlas is characterized not only by a unique spatial extent of a specific administrative unit but also by a personalized visual model. The implemented nested iteration technique ensures that each map contains an individual color scale and legends that strictly correspond to the specificity of the indicator (feature) for the given district. This approach ensures a high level of informativeness and clarity in the cartographic materials, minimizing the probability of subjective errors during the interpretation of large spatial datasets (Fig. 2).

The developed functional model reflects the process of transforming input data into the final atlas and consists of three key stages:

- the preparatory stage involves defining content requirements, collecting and spatially aggregating thematic layers into a unified geospatial database, which culminates in the creation of a base graphical layout within the QGIS environment;

- the iterative processing stage (the core of the method) starts from the "Queries" action and implements parallel processing of the thematic and territorial components. The left branch (the thematic component) of the algorithm is responsible for filtering geometry and semantic features, followed by the dynamic application of styles (.qml). It

includes the following actions: formation of thematic layers, combining geometric primitives with thematic variables, as well as a visualization query and a variable style.

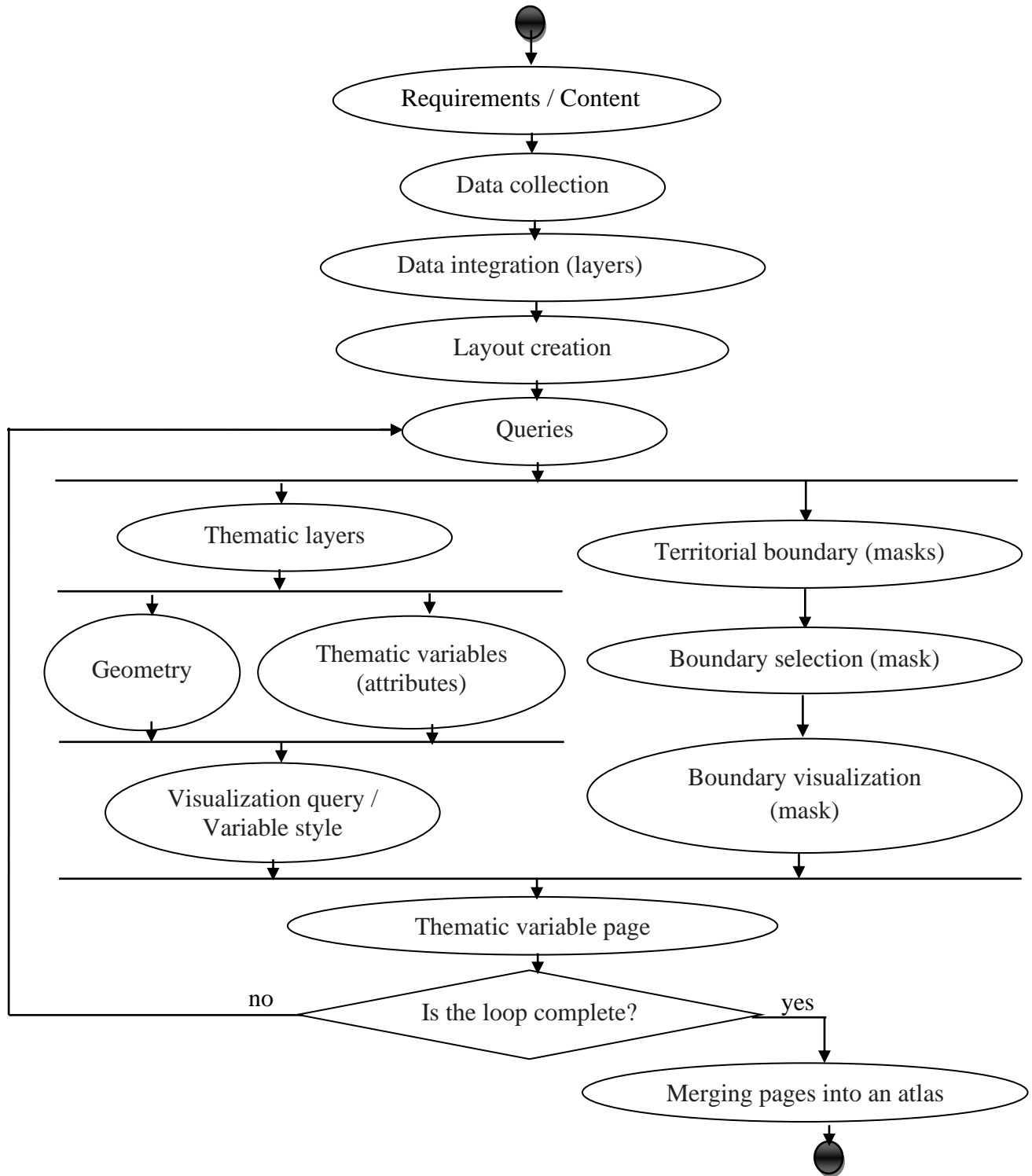


Fig. 2. Functional model of automated dynamic mapping using the nested iteration method (via PyQGIS)

The right branch (the territorial component) ensures the correct visual delimitation of the study area through a boundary masking mechanism and its graphical rendering for a specific page. It includes the following actions: determining the territory boundary, selecting the boundary (mask), and visualizing the boundary;

- at the synthesis stage, the looped structure of the algorithm ensures the automated generation of pages for each thematic variable (mask), which are compiled into a cohesive atlas at the final step. A key control element is the decision-making node – represented on the diagram as a logical diamond for verifying the specified condition: "Is the loop complete?". By checking whether unprocessed geometric contours remain in the database, the process routes back via the conditional transition "no" to a new iteration step at the "Queries" block. If the conditional transition registers a "yes" status (all territorial units and indicators have been processed), the algorithm proceeds to the final action – automated consolidation and merging of all generated pages into a single, comprehensive atlas through the "Merging pages into an atlas" action.

Table 1.

Comparative analysis of time expenditures between manual and automated mapping methods

Work Stage	Manual Method (min/map)	Automated Method
Template Preparation	60 min	120 min (including coding)
Filtering Configuration	5 min	0 min (automated)
Style Application (.qml)	3 min	0 min (automated)
Zooming and Layout	2 min	0 min (automated)
Export and Numbering	2 min	10 min (for the entire project)
Total for 250 maps	~3,000 min (50 hours)	~150 min (2.5 hours)

An assessment of time savings when applying automated dynamic mapping via the nested iteration method compared to the manual method is demonstrated in Table 1 using the example of creating an atlas for 50 administrative units with 5 different indicators for each (a total of 250 maps).

Automation enables a 20-fold acceleration in the process of creating series of thematic maps with consistent styling. Time savings amount to approximately 95% when working with large datasets. The applied methodology allows for processing extensive geodatabases without overloading the RAM, thanks to dynamic SQL filtering.

For more detailed information regarding mapping automation, including the provision of comprehensive methodological recommendations for its application, please contact the authors of the article.

Conclusions. The implementation of the proposed nested iteration into thematic mapping processes represents a qualitative transition from manual layout composition to a fully automated, intelligent environment for geodata generation. The application of the specified methodology for thematic mapping eliminates the human factor (errors during manual style changes), ensures stylistic consistency across all atlas pages, and frees up professional resources for analytical work rather than technical map preparation.

The research results demonstrate that the integration of Python algorithms with QGIS Atlas functionality enables a 20-fold acceleration in atlas preparation. The proposed approach establishes a reliable technological foundation for the digital transformation of institutions, allowing specialists to focus on the strategic analysis of spatial patterns, which is critical for the sustainable management of land and other natural resources in modern conditions.

References

1. Cabinet of Ministers of Ukraine. (2013, September 4). *Pro zatverdzhennia Poriadku zahalnodержavnogo topografichnoho i tematychnoho kartohrafuvannia* [On approval of the Procedure for national topographic and thematic cartography] (Resolution No. 661). Available at: <https://zakon.rada.gov.ua/laws/show/661-2013-п>

2. QGIS Project. (n.d.). *QGIS documentation: The latest guide to using QGIS for geoprocessing and map creation*. Available at: <https://qgis.org>
3. Steiniger, S., & Hunter, A. J. (2013). The 2012 free and open source GIS software map – A guide to help select the right software for GIS projects. *Computers, Environment and Urban Systems*, 39, 136–150. DOI: <https://doi.org/10.1016/j.compenvurbsys.2012.10.003>
4. Hengl, T., Miller, B. A., Aldabaa, A. A., et al. (2017). Digital mapping of soil properties using ensemble machine learning. *Scientific Reports*, 7(1), 1–15. DOI: <https://doi.org/10.1038/s41598-017-18274-6>
5. Adeniyi, O. D., Brenning, A., Bernini, A., Brenna, S., & Maerker, M. (2023). Digital mapping of soil properties using ensemble machine learning approaches in an agricultural lowland area of Lombardy, Italy. *Land*, 12(2), 494. DOI: <https://doi.org/10.3390/land12020494>
6. Espinel, R., Herrera, G., Rivadeneira Garcia, J. L., & Escandón-Panchana, P. C. (2024). Artificial intelligence in agricultural mapping: A review. *Agriculture*, 14(7), 1071. DOI: <https://doi.org/10.3390/agriculture14071071>
7. Dorosh, Y., Dorosh, A., Derkul'skiy, R., & Bratinova, M. (2024). Application of GIS in land management on the example of Ukraine. *Acta Scientiarum Polonorum Administratio Locorum*, 23(1), 31–41.
8. Singh, B. (2018). Geographical information system in agriculture. *International Journal of Scientific Research in Science and Technology*, 4(2), 702–705. Available at: https://academia.edu/37114891/Geographical_Information_System_in_Agriculture
9. Graser, A. (2016). *Learning QGIS* (3rd ed.). Birmingham: Packt Publishing Ltd, 210. Available at: https://books.google.com.ua/books/about/Learning_QGIS_Third_Edition.html?id=rUXicWAAQBAJ

10. Sherman, G. (2018). *The PyQGIS programmer's guide: Extending QGIS 3 with Python 3*. Locate Press, 252. Available at: <https://locatepress.com/book/ppg3>
11. Menke, K. (2019). *Mastering QGIS* (2nd ed.). Birmingham: Packt Publishing, 404.

Й.М. Дорош, Б.Б. Заячківська, Р.А. Харитоненко, Д.М. Мельник
ЗАСТОСУВАННЯ МЕТОДУ ВКЛАДЕНОЇ ІТЕРАЦІЇ ДЛЯ
АВТОМАТИЗОВАНОГО КАРТОГРАФУВАННЯ

Анотація. У статті розглядається метод автоматизації створення серії картографічних матеріалів на основі інтеграції інструментарію QGIS Atlas та мови програмування Python. Метою дослідження є оптимізація процесу підготовки багатосторінкових тематичних атласів для великих масивів просторових даних. Для досягнення поставленої мети застосовано авторський підхід до інтеграції алгоритмів вкладеної ітерації, що дозволяє гнучко адаптувати просторові межі та індивідуальні параметри візуального відображення для кожного елемента системи. Ключовим елементом є розроблений алгоритм, що базується на обробці події `featureChanged`, що забезпечує автоматичну синхронізацію атрибутів району з відповідними файлами стилів `.qml`. Модель апробовано на масиві даних обсягом понад 20 тис. контурів. Результати апробації підтвердили високу ефективність методу. Швидкість генерації серії карт зросла у приблизно 20 разів порівняно з традиційними методами, при повному виключенні суб'єктивних помилок візуалізації. Запропонований підхід становить собою універсальну методологічну платформу для автоматизованої трансформації складних просторових датасетів у високоточні графічні моделі. Дане рішення забезпечує можливість оперативного представлення результатів багатофакторних досліджень у будь-якій галузі, де вимагається поєднання високої інтенсивності обробки великих масивів інформації

з дотриманням індивідуальних параметрів візуальної інтерпретації для кожного окремого елемента системи.

Ключові слова: автоматизоване картографування, QGIS Atlas, PyQGIS, агрохімічний моніторинг, ГІС-автоматизація, метод вкладеної ітерації, динамічна стилізація, тематичні атласи, Python API, файли стилів *.qml, автоматизація макетів, алгоритмізація картографування.