Didactic potential of neuropedagogical exercises in teaching informatics

NISHONOV AKMAL

Associate Professor, Department of Applied Sciences and Information Technologies,
Fergana Regional Center for Pedagogical Excellence.
ORCID: https://orcid.org/0009-0009-9524-5863

Abstract. *This article examines the didactic potential of the neuropedagogical approach in teaching informatics, the exercises that serve to develop students' cognitive activity, and the methodology for applying them in the educational process. In particular, it analyses the role in informatics education of methods such as retrieval practice, spaced repetition, step-by-step instruction, algorithmic thinking, error analysis, and metacognitive reflection. Sample neuropedagogical exercises suited to the topics of programming, algorithm design, and information processing are also developed. Drawing on scientific research in the psychology of learning and education, the article offers methodological recommendations for using these exercises in the classroom.*

Keywords: *neuropedagogy, informatics education, cognitive processes, working memory, attention, algorithmic thinking, retrieval practice, spaced repetition, metacognition, programming, digital education.*

Introduction

One of the priority tasks of the modern education system is not only to build solid theoretical knowledge in students, but also to develop their competencies in independent and critical thinking, analysing problem situations, working out optimal solutions, and effectively applying acquired knowledge in new conditions. In this process, increasing students' cognitive activity, conscious processing of information, and the formation of skills for independently managing their own learning are of particular importance. In teaching informatics especially, these competencies are closely tied to developing algorithmic thinking, logical analysis, step-by-step problem solving, and the ability to use digital technologies effectively.

When informatics is taught, some students may have difficulty understanding the order in which algorithms are executed, identifying errors in program code, or connecting new concepts with prior knowledge. In such situations, it is necessary not to limit instruction to traditional explanation and repetitive exercises, but to use methods that take into account how students perceive, remember, and process information and how they monitor their own activity.

Neuropedagogy is an interdisciplinary approach that seeks to link scientific knowledge about brain function and learning processes with pedagogical practice. In this field, it is important to draw on findings from cognitive psychology, educational psychology, and neuroscience. Caution is required when turning scientific evidence about the brain into pedagogical recommendations, because the effectiveness of any teaching method should be judged not only by its neurobiological explanation but also by empirical results in

educational settings. From this perspective, the use of neuropedagogical exercises in teaching informatics is a promising direction for improving how students acquire knowledge, developing their algorithmic thinking, and forming independent learning skills.

Aim and objectives of the study

The aim of the study is to analyse the theoretical foundations of using neuropedagogical exercises in teaching informatics and to develop practical and methodological recommendations for applying them in the classroom.

To achieve this aim, the following objectives were set:

To analyse the scientific foundations of the neuropedagogical approach as they relate to education.

To highlight the importance of attention, working memory, and metacognitive activity in informatics.

To develop exercises aimed at developing students' algorithmic thinking.

To show the didactic potential of retrieval practice, spaced repetition, and error analysis.

To develop methodological recommendations for integrating neuropedagogical exercises into the lesson process.

Methods

The article is theoretical and analytical in nature. The study used analysis of the scientific literature, comparative analysis, pedagogical generalisation, and methodological modelling. The principles of retrieval practice, metacognition, and the management of learning activity described in scientific sources on the psychology of learning and education were examined in relation to the content of the informatics subject.

The exercises presented in the article were developed as methodological proposals. There is no information that they have been piloted in a particular school or group of students. For this reason, the article makes no claims of new empirical results or statistical indicators regarding the effectiveness of these exercises.

Results

Theoretical foundations of neuropedagogy in informatics education

In the process of learning informatics, students perform several cognitive tasks at once: understanding the task statement, breaking the problem into parts, designing an algorithm, writing code, checking the result, and correcting errors. Attention, working memory, long-term memory, and executive functions play an important role in such activity.

Working memory makes it possible to actively hold and process information for a short time. When building a complex algorithm or program, a student must keep variables, the sequence of operations, and conditions in mind together. If the task is overly complex or the explanation is disorganised, the load on working memory may increase. Therefore, it is important to explain new concepts step by step, show worked examples, and then move on to tasks completed independently.

In consolidating memory, an important role is played not only by re-reading knowledge but also by independently retrieving it from memory. When a student recalls a previously learned concept by answering a question, explaining code, or reconstructing an algorithm, an opportunity is created to actively process the knowledge. A systematic review covering studies conducted in school and classroom settings has shown that retrieval practice benefits learning.

In addition, metacognition – the ability to be aware of, monitor, and regulate one's own thinking and learning process – is of great importance in informatics education. By making a plan before writing a program, checking results during the work, and analysing

their own errors at the end, a student develops independent problem-solving skills. The guidance report updated by the Education Endowment Foundation in 2025 recommends teaching metacognitive strategies explicitly and embedding them in subject content. Thus, neuropedagogical exercises in informatics should not remain merely a set of engaging tasks; they should serve to purposefully organise the student's processes of acquiring, processing, and applying knowledge and evaluating their own activity.

Neuropedagogical exercises used in informatics

Retrieval practice exercises

Retrieval practice is a method based on recalling previously learned knowledge from memory without looking at the source. In informatics, it can be used to start a new lesson, consolidate a previous topic, and determine students' level of knowledge.

Sample exercise: "Recall without looking at the code".

Topic: Variables in the Python programming language.

Students are asked the following questions:

1. What is a variable?
2. How is a value assigned to a variable?
3. How do an integer and a text value differ?
4. Which command is used to display the values of two variables on the screen?

Students write their answers independently for 3–4 minutes. The answers are then checked, and misunderstood points are discussed. This exercise serves to activate students' prior knowledge, identify gaps in understanding, and prepare them for learning a new topic. It can be used in a short form in every lesson.

Spaced repetition exercises

Spaced repetition involves reviewing knowledge at different time intervals. Rather than limiting review of a topic to the day it was learned, recalling it again in subsequent lessons can support learning.

Sample exercise: "Reconstruct the algorithm".

Topic: Linear algorithms.

Students are asked to build an algorithm for performing an everyday task – for example, calculating the sum of two numbers.

In the first lesson, students build the algorithm with the teacher's help. In the next lesson, they rewrite the algorithm from memory. A week later, they adapt the algorithm to another conditional problem. The interval between reviews is set according to students' results. If the core concept has not been mastered, additional explanation and guiding support should be provided first.

Step-by-step algorithmic thinking exercises

Algorithmic thinking includes analysing a problem, breaking it into smaller tasks, determining the sequence of operations, and checking the solution. Dividing complex tasks into stages helps manage the load on the student's working memory.

Sample exercise: "Break a big problem into small steps".

Topic: Finding the larger of two numbers.

The task is completed in the following stages:

5. Read the problem statement.
6. Identify the input data.
7. Define the comparison condition.
8. Write the algorithm.
9. Test the result with several values.

For example, in Python:

```
a=int(input("Enter the first number: "))
b=int(input("Enter the second number: "))
      if a > b:
          print(a)
      elif b > a:
          print(b)
      else:
          print("The numbers are equal")
```

Students are required not simply to copy the code, but to explain the purpose of each condition and also to test the case of equal numbers. This exercise aims to develop skills in

algorithmic analysis, understanding conditions, and testing a solution.

Error detection and correction exercises

Working with errors has significant didactic value in learning programming. In addition to finding an error in the code, the student should explain its cause and verify the correctness of the corrected solution.

Sample exercise: "Find and explain the error".

The following program should output the sum of two numbers:

```
a = 7
b = 5
print(a - b)
```

The student is asked the following questions:

- What result does the program produce?
- What is the difference between the purpose of the task and the result?
- On which line is the error?
- How can the error be corrected?
- How can the corrected code be checked?

Correct solution:

```
a = 7
b = 5
print(a + b)
```

The main purpose of the exercise is not to penalise the error, but to create an opportunity to analyse it and improve one's knowledge. The teacher should teach students to predict the program's output in advance, run the code, and compare the result obtained.

Metacognitive exercises

Metacognitive exercises help students monitor and regulate their own thinking process. In informatics lessons, this is done by planning the task, checking the process of completing it, and evaluating the final result.

Sample exercise: "Plan – Do – Check".

The student is given a task to create a particular program and completes the following three stages:

Stage one – planning. The student analyses the problem statement and writes down the steps needed for the solution.

Stage two – doing and monitoring. The student writes the code, checks the program's output, and modifies the solution if necessary.

Stage three – evaluation. The student checks whether the task was completed correctly and explains at which step they encountered difficulty.

At the end of the session, students answer the following questions:

- What did I start with when solving the problem?
- Which step was difficult for me?
- How did I identify the error?
- What approach will I use next time?

The metacognitive approach aims to shape the student not merely as a recipient of ready-made solutions, but as an active participant who manages their own learning.

Methodology for integrating neuropedagogical exercises into the lesson

To apply neuropedagogical exercises effectively, they need to be linked to the lesson's aim, content, and expected outcomes. It is advisable to organise the exercises not as separate, disconnected activities, but as coherent components of the learning process.

The following sequence can be used in an informatics lesson:

Activating prior knowledge. At the start of the lesson, short questions, code-explanation, or algorithm-recall exercises are carried out.

Explaining the new concept. The teacher explains the topic by dividing it into small, logically connected parts.

Showing a worked example. The teacher demonstrates how they analyse the problem, plan the solution, and check the result.

Independent practical activity. Students complete similar tasks, then slightly more complex ones.

Error analysis. Students identify incorrect

solutions and explain the reasons for correcting them.

Reflection and review. At the end of the lesson, students evaluate their results; in subsequent lessons, the key concepts are recalled again.

Such a sequence creates conditions not only for students to remember knowledge, but also to understand it, apply it in practice, and monitor their own activity.

Criteria for evaluating the effectiveness of neuropedagogical exercises

When introducing neuropedagogical exercises, it is not correct to accept their effectiveness as a guaranteed outcome in advance. They need to be evaluated pedagogically by applying them in a particular class and systematically observing the results.

The following criteria can be used:

Evaluation criterion	Observed indicator
Knowledge retrieval	The student answers questions on the previous topic independently
Algorithmic thinking	Is able to break a problem into sequential stages
Programming skill	Writes code, runs it, and checks the result
Working with errors	Finds an error, explains its cause, and corrects it
Independent learning	Plans their own work and evaluates its outcome
Knowledge retention	Is able to complete a similar task after some time has passed

If the teacher wishes to investigate the effect of the exercises, they should compare pre- and post-assessment results and keep task difficulty and assessment criteria as consistent as possible. Where possible, using a comparison group and longer-term follow-up increases the reliability of conclusions. Knowledge growth cannot be firmly confirmed solely on the basis of students' activity in class or the positive opinions they express. In assessment, it is also important to take into account practical tasks, knowledge-retrieval results, and the ability to apply the learned material later.

Discussion

The main advantage of using neuropedagogical exercises in teaching informatics is the opportunity to organise learning activity on the principles of active knowledge processing and self-regulation. Retrieval practice exercises allow the student to reconstruct prior knowledge, while spaced repetition serves to reactivate knowledge over time. Step-by-step instruction divides complex tasks into manageable parts, and error analysis helps students understand cause-and-effect relationships in the programming process.

At the same time, the neuropedagogical approach cannot be regarded as a universal method that produces the same results for all students. The benefit of the exercises depends on students' prior knowledge, task complexity, the teacher's methodological skill, and the learning conditions. Moreover, not all views spread under the name of neuropedagogy are scientifically grounded. For example, the advantage of dividing students into visual, auditory, or kinaesthetic learning styles and adapting instruction to these categories has not been confirmed by reliable empirical evidence. Therefore, in informatics lessons, exercises should be chosen not according to a student's supposed "brain type", but according to the learning objective, the nature of the task, and evidence-based pedagogical principles. In future, important research tasks in this area include organising experimental studies in school settings, comparing different exercises against learning outcomes, and studying their potential across different age groups.

Conclusion

The use of neuropedagogical exercises in teaching informatics offers methodological opportunities aimed at developing students' skills in acquiring knowledge, algorithmic thinking, analysing programming errors, and managing their own learning. The retrieval practice, spaced repetition, step-by-step instruction, error analysis, and metacognitive reflection exercises analysed in this article can

be adapted to different stages of an informatics lesson. Their effective use requires the teacher to set clear goals, adapt tasks to students' level of preparation, and systematically monitor learning outcomes. In applying neuropedagogical exercises, it is also necessary to rely on scientifically confirmed learning principles and to avoid oversimplified and unsupported views about brain function. To determine the effectiveness of the exercises in real educational settings, it is advisable to test them through pedagogical experiments and analyse the results using appropriate evaluation criteria. Overall, the neuropedagogical approach can serve as a useful methodological basis for orienting informatics education towards students' active thinking, independent inquiry, and monitoring of their own knowledge.

References

1. Karimov I. Informatika va axborot texnologiyalari: O‘quv qo‘llanma [Informatics and Information Technologies: A Textbook]. – Tashkent: TDPU Publishing, 2022. – 240 p.
2. Тункун Я.А. Основы нейропедагогики: история, теория и практика [Fundamentals of Neuropedagogy: History, Theory and Practice] / Ж.: Киберленинка, 73-2, – 203. – С. 204.
3. Клемантович И. П., Степанов В. Г. Нейропедагогика: предмет исследования [Neuropedagogy: The Subject of Research] // Фундаментальные исследования. – 2015. – № 2-11. – С. 2464–2468.
4. Казаченок В. В. Применение нейронных сетей в обучении [The Use of Neural Networks in Teaching] // Информатика и образование. 2020. № 2. С. 41–47.
5. Эгамбердиева, Ф. (2025). Нейропедагогика как наука: концепции, задачи и перспективы [Neuropedagogy as a Science: Concepts, Tasks and Prospects]. Oriental Renaissance: Innovative, educational, natural and social sciences, 5(20), 171–176. https://doi.org/10.5281/zenodo.14876978
6. Сорочинский, М. А., Адамов, В. С. Нейропедагогика как направление развития современного образования [Neuropedagogy as a Direction in the Development of Modern Education] // Современное образование: традиции и инновации. – 2022. – № 2. – С. 37–39.
7. Agarwal, P. K., Nunes, L. D., & Blunt, J. R. (2021). Retrieval Practice Consistently Benefits Student Learning: A Systematic Review of Applied Research in Schools and Classrooms. Educational Psychology Review, 33, 1409–1453. https://doi.org/10.1007/s10648-021-09595-9
8. Education Endowment Foundation. (2025). Metacognition and Self-Regulated Learning: Guidance Report. Second Edition. https://educationendowmentfoundation.org.uk/education-evidence/guidance-reports/metacognition
9. Education Endowment Foundation. Metacognition and Self-Regulation: Evidence Review. https://educationendowmentfoundation.org.uk/education-evidence/evidence-reviews/metacognition-and-self-regulation-guidance-report
10. Rousseau, L. (2024). Dispelling Educational Neuromyths: A Review of In-Service Teacher Professional Development Interventions. Mind, Brain, and Education, 18(3), 270–287. https://doi.org/10.1111/mbe.12414
11. Newton, P. M., & Salvi, A. (2020). How Common Is Belief in the Learning Styles Neuromyth, and Does It Matter? Frontiers in Education, 5, 602451. https://doi.org/10.3389/feduc.2020.602451
12. Sweller, J., Ayres, P., & Kalyuga, S. (2011). Cognitive Load Theory. Springer. https://doi.org/10.1007/978-1-4419-8126-4
13. Dunlosky, J., Rawson, K. A., Marsh, E. J., Nathan, M. J., & Willingham, D. T. (2013). Improving Students' Learning With Effective Learning Techniques: Promising Directions From Cognitive and Educational Psychology. Psychological Science in the Public Interest, 14(1), 4–58. https://doi.org/10.1177/1529100612453266
14. Brown, S. B. R. E. (2023). The persistence of matching teaching and learning styles: A review of the ubiquity of this neuromyth, predictors of its endorsement, and recommendations to end it. Frontiers in Education, 8, 1147498. https://doi.org/10.3389/feduc.2023.1147498