Region-Educational Research and Reviews, 2025,Volume 7, Issue 6
DOI:10.12238/rerr.v7i6.4178
ISSN Online:2661-4634
ISSN Print:2661-4626

Exploring the Path of Cultivating Computational Thinking Ability in Primary and Secondary School Students

Daiwen Wu

College of Education Science, Weinan Normal University, Weinan, Shaanxi Province, 714099, China

Abstract: This paper explores in depth the cultivation path of computational thinking ability for primary and secondary school students. At the theoretical level, it systematically sorts out and analyzes the definition and connotation of computational thinking. Clear computational thinking refers to the use of fundamental concepts in computer science, covering key elements such as decomposition, abstraction, modeling, algorithm design, evaluation, and optimization, to engage in thinking activities related to human problem-solving, system design, and behavioral understanding. In terms of exploring the cultivation path, comprehensive and specific strategies have been proposed from multiple dimensions such as curriculum design, teaching method innovation, and faculty team building. In terms of curriculum design, the article particularly emphasizes the importance of courses and the integration of disciplines. In terms of innovative teaching methods, the project-based learning teaching approach was explained in detail. In terms of teacher team building, a detailed construction plan for teacher training has been proposed, and through multidimensional research methods and rich practical cases, some achievements with certain theoretical and practical value have been achieved.

Keywords: primary and secondary school students, computational thinking, curriculum design, path exploration

1. Introduction

Computational thinking is an important and gradually gaining widespread attention as a way of thinking. It is not just limited to the field of computer science, but a series of cognitive activities[1]. For example, using basic concepts of computer science to solve problems, design systems, and understand human behavior. Computational thinking emphasizes the decomposition of complex problems into manageable sub problems through abstraction and modeling, design of algorithms and steps, evaluation and optimization of solutions, in order to discover the essence and laws of the problem and solve it accordingly. This way of thinking not only plays a core role in computer programming, software development, and other fields, but also has a wide range of applications in many disciplines such as mathematics, physics, engineering, economics, and practical life.

This article aims to deeply explore the effective paths for cultivating computational thinking abilities among primary and secondary school students. Through a systematic analysis of the current educational situation, students' characteristics, and problems in teaching practice, combined with educational theories and practical experiences, it proposes targeted and highly operational strategies for cultivating computational thinking abilities, with the goal of effectively enhancing the computational thinking abilities of primary and secondary school students.

2. Definition and Connotation of Computational Thinking

2.1 Definition and Explanation

The concept of computational thinking has attracted extensive attention and in-depth discussion in the fields of education and academia since its introduction. Professor Zhou Yizhen, the director of the Computer Science Department at Carnegie Mellon University in the United States, first proposed the concept of computational thinking in the authoritative American computer journal "Communications of the ACM" in March 2006. It is defined as "a series of thinking activities covering the breadth of computer science, such as problem-solving, system design, and understanding human behavior, which utilize the basic concepts of computer science"[2]. This definition laid a solid foundation for the research on computational thinking and enabled people to start viewing the application and value of computational thinking in different fields from a broader perspective.

Professor Zhou Yizhen further explained that computational thinking is a thinking process that simplifies, embeds, transforms, and simulates problems through various methods, breaking down a seemingly complex problem into a clear and easily solvable one. For example, when solving complex mathematical problems, the reduction method can be used to simplify the complex problem into several simple sub-problems, and solve the sub-problems one by one to achieve the solution of the original problem; when designing a computer system, through the methods of transformation and simulation, the actual requirements are converted into models that the computer can understand and process, thereby constructing a system that meets the requirements. This way of thinking emphasizes starting from the perspective of computer science, using its unique concepts and methods to solve various problems, not only limited to the traditional field of computer programming, but also widely applied in problem-solving, system design, and understanding human behavior, etc.

2.2 In-depth Analysis of the Connotation

The connotation of computational thinking is rich and diverse, covering multiple key elements. These elements are interrelated and together form the core content of computational thinking.

Decomposition is one of the important connotations of computational thinking. Decomposition refers to dividing a large and complex problem into several sub-problems that are easier to handle[3]. When dealing with the development of large-scale management information system software projects, it can be decomposed into several sub-problems such as user interface and interaction design, core functional module implementation, database design, etc. Each sub-problem can be further divided. For example, the implementation of the core functional module can be divided into the writing of specific function functions, the design of interfaces between modules, and database access, etc. Through this top-down and gradually decomposing problem from the macroscopic to the microscopic approach, complex problems are gradually refined, thereby reducing the difficulty of problem-solving. Then, independently analyze and solve each sub-problem, and finally achieve the effective solution of the entire complex problem.

Abstraction is another key element of computational thinking. It requires people to filter out unnecessary details when thinking about problems, quickly extracting the essential features and key elements of the problem, thereby forming a concise and accurate understanding of the problem[4]. For example, when designing a graphics drawing program, one can focus on the basic shapes and positional relationships of the graphics, which are the essential features of the graphics, while temporarily ignoring the specific details and non-key information such as the color and thickness of the lines. This way, when solving problems, one can avoid being disturbed by the complicated details, thus focusing more on the core content and ultimately improving the efficiency and accuracy of problem-solving.

Modeling is one of the core components of computational thinking. It involves constructing models that can describe problems and their solutions by using mathematical and logical methods on an abstract basis[5]. When studying physical phenomena, such as the laws of object motion, a mathematical model can be established to describe the relationships between physical quantities such as the position, velocity, and acceleration of the object using formulas and equations. Through the analysis and solution of the model, one can deeply understand the essence of the problem, predict its development trend, and provide effective guidance for solving the problem.

Algorithm design is an important manifestation of computational thinking. It refers to a series of clear and orderly operational steps designed for a specific problem[6]. When writing sorting programs, different algorithms can be designed, such as bubble sort, quick sort, etc. The bubble sort algorithm gradually "bubbles" the largest (or smallest) element to the end of the array by repeatedly comparing adjacent elements and swapping positions; the quick sort algorithm adopts the divide-and-conquer approach, dividing the array into two parts, and recursively sorting each part to ultimately achieve the ordered arrangement of the entire array. Reasonable algorithm design can improve the efficiency and quality of problem-solving and is a key link of computational thinking in practical applications.

3. Analysis of the Problems and Causes of Computational Thinking Skills among Primary and Secondary School Students

3.1 Existing Problems

At present, there are many deficiencies in the computational thinking abilities of primary and secondary school students. Firstly, the computational thinking skills of students need to be improved. Although students perform relatively well in the decomposition dimension, their scores in the abstract, modeling, algorithm design, evaluation and optimization dimensions are all at a medium-low level, indicating that there are obvious shortcomings in the development of these key abilities[7]. When solving mathematical problems, students can break down the problem into known conditions and solution steps, but they have difficulties in abstracting the essential features of the problem, establishing mathematical models, and designing efficient algorithms, resulting in low problem-solving efficiency.

Secondly, students lack the ability to apply computational thinking. In class, although many students have learned and mastered some concepts and methods of computational thinking, they have difficulty applying these knowledge to solve practical problems in real life and study. When facing daily problems, such as scheduling study time and personal shopping budget, students rarely use the methods of computational thinking, such as breaking down problems and designing algorithms, to formulate reasonable solutions. This clearly indicates that there is a significant gap between students' ability to convert theoretical knowledge into practical application skills. The awareness and ability to combine computational thinking with actual situations need to be strengthened.

Furthermore, the weak awareness of computational thinking among students is also a significant issue. Some students have a poor understanding of the concept and importance of computational thinking, and do not have a deep comprehension of its role in the problem-solving process. They fail to fully grasp it during their learning[8]. During the learning process, students place more emphasis on memorizing knowledge and mastering test-taking skills, while neglecting the training of their thinking abilities. In classroom teaching, when teachers ask students to solve problems using computational thinking, some students appear at a loss and unable to start, which fully reflects that students have not yet developed the awareness and habit of actively applying computational thinking to solve problems.

3.2 Analysis of Reasons

The reasons for the insufficient computational thinking ability of primary and secondary school students are diverse, mainly including factors such as curriculum design, teaching methods and teaching staff.

In terms of curriculum design, currently, the curriculum system of primary and secondary schools has relatively few courses specifically focused on the cultivation of computational thinking, and lacks systematicness and coherence[9]. Although the information technology courses do cover some elements of computational thinking, in actual teaching, they often focus on the training of computer operation skills rather than providing in-depth cultivation of computational thinking. Mathematics, science and other subjects contain rich elements of computational thinking, but due to the limitations of course goals and teaching priorities, these resources have not been fully explored and utilized, resulting in the lack of effective carriers for the cultivation of computational thinking in curriculum implementation. Moreover, the integration of courses among different disciplines is insufficient, and there are no cross-disciplinary courses for the cultivation of computational thinking, making it difficult for students to combine the knowledge and methods of different disciplines and apply computational thinking to solve complex problems.

In terms of teaching methods, traditional teaching approaches are still predominant in primary and secondary school classrooms. This teaching method, which focuses on imparting knowledge and neglects the students' central role, is unable to effectively cultivate students' computational thinking abilities[10]. During the teaching process, teachers often directly provide solutions to problems, while students merely passively accept and imitate, lacking opportunities for independent thinking and exploration. Such teaching methods are not conducive to fostering students' innovative thinking and practical skills, nor do they help stimulate their interest and initiative in learning. Moreover, the lack of practical activities and project-based learning in the teaching process means that students rarely have the chance to apply the computational thinking knowledge they have learned to real-world problems, resulting in their application abilities not being effectively exercised and improved.

The cultivation of students' computational thinking ability is also an important factor related to the teaching staff. On one hand, some teachers themselves do not have a deep understanding and mastery of computational thinking, lack relevant teaching experience and skills, and thus find it difficult to effectively incorporate the cultivation of computational thinking into their teaching[11]. On the other hand, there is relatively little training for computational thinking teaching, and teachers lack opportunities for learning and improvement, making it difficult for them to master and adopt advanced teaching methods and strategies, and unable to meet the actual needs of cultivating students' computational thinking ability.

4. Exploration of Training Paths

4.1 Establish specialized courses

4.1.1 Course objective setting

The formulation of course objectives should closely align with the cognitive development patterns of students and the core essence of computational thinking. For students in the lower grades of primary school (grades 1-3), their cognition mainly transitions from intuitive image thinking to abstract logical thinking. This type of thinking prefers to learn through specific objects and activities, and is full of curiosity about the world. The course objectives for this stage should focus on stimulating students' interest in computational thinking and guiding them to initially understand the basic concepts and methods of computational thinking. For example, through simple and interesting games and activities, students can understand what problem decomposition is, such as breaking down a jigsaw puzzle game into steps like finding the puzzle pieces and attempting to assemble them; initially experiencing abstract concepts, such as abstracting the basic features of shapes from different-shaped objects; and learning some simple pattern recognition, such as identifying the arrangement patterns of shapes. Through the implementation of these activities, students' observation skills and initial logical thinking abilities are exercised, laying a foundation for future learning.

The students' ability to understand some complex concepts and relationships through abstract logical thinking has already developed to a certain extent in the upper grades of primary school (grades 4-6). The curriculum objectives at this stage should be further deepened to cultivate students' ability to solve simple practical problems using computational thinking[12]. At this stage, students can learn how to use the decomposition method to break down complex problems in mathematics or life into several sub-problems, and find solutions for each sub-problem; they can extract key information from practical problems and master more complex abstract skills to establish simple mathematical models or logical models; they can learn to design simple algorithms, such as implementing functions like sorting sequences and text processing through writing simple programs; and cultivate students' awareness of making preliminary evaluations and reflections on problem-solving strategies, being able to identify their own shortcomings during the problem-solving process and attempt to improve them.

At the junior high school stage, students' thinking abilities further improve, and they begin to possess strong abstract thinking and logical reasoning skills[13]. The course objectives should focus on the cultivation of students' comprehensive application abilities and innovative thinking. Students need to deeply study algorithm design, master the principles of common algorithms, be able to flexibly choose algorithms for different problems, and be able to optimize and improve algorithms; learn to use computational thinking for system design, such as designing small management information systems and simple website architectures; cultivate students' critical thinking and innovative abilities, encourage them to question existing solutions and propose improvement suggestions, try to think from different perspectives, and propose innovative solutions; at the same time, guide students to apply computational thinking in the study of other subjects, achieve the integration of knowledge in different disciplines, and improve students' ability to solve complex problems.

4.1.2 Course Content Design

The design of the course content should be comprehensive and systematic, covering key areas such as programming fundamentals, algorithm design, and data analysis, in order to establish a complete system for cultivating computational thinking.

The important cornerstone for cultivating computational thinking - programming foundation. In the primary school stage, graphical programming tools such as Scratch can be introduced. It features an intuitive graphical interface and simple, understandable instruction modules, enabling students to easily get started. They can drag and combine instruction modules to write programs and achieve various interesting functions, such as animation creation and game development[14]. The Scratch graphical programming is vivid and appealing, in line with the cognitive characteristics of primary school students, reducing the programming barrier and stimulating students' interest in learning, allowing them to initially understand the basic concepts and processes of programming in a relaxed and enjoyable atmosphere. As the learning progresses, in the junior high school stage, it can gradually transition to code programming, such as the Python language. The Python language is concise and clear, has powerful functions, and has rich libraries and modules, which can meet the programming needs of different fields. When students learn Python programming, they not only need to master the basic grammar and data types, but also need to learn to use functions for modular programming to improve the readability, reusability, and maintainability of the code.

One of the core elements of computational thinking is algorithm design. The course content should include basic concepts of algorithms, the design and implementation of common algorithms, optimization and analysis of algorithms, etc.[15]. In the primary school stage, through simple examples such as finding the largest number among three numbers, calculating the greatest common divisor and least common multiple of two numbers, students can understand the concept and function of algorithms and learn to describe the steps of algorithms in natural language. In the junior high school stage, students will delve deeper into the principles and implementation of common algorithms, such as sorting algorithms (bubble sort, quick sort, insertion sort, etc.) and search algorithms (sequential search, binary search, etc.). Students need to understand the basic ideas and implementation steps of each sorting algorithm and be able to analyze the time complexity and space complexity of the algorithm. Taking bubble sort as an example, students should understand that it is a process of repeatedly comparing adjacent elements and exchanging positions to gradually "bubble" the largest (or smallest) element to the end of the array. Through the actual programming implementation of these algorithms, students can deepen their understanding and mastery of algorithms, and improve their logical thinking ability and programming skills.

In the digital age, data analysis is an indispensable ability and an important part of cultivating computational thinking[16]. In the design of course content, it should include data collection, organization, analysis, and visualization. At the primary school level, students can use simple charts (such as bar charts and pie charts) to learn to collect and organize data through simple survey activities such as statistics on students' interests, heights, and weights in the class. At the junior high school level, they will further learn methods and tools for data analysis, such as using Excel for data processing and analysis, learning to use functions and data pivot tables for data calculation and statistical analysis; learning the basic concepts and methods of data analysis, such as the central tendency of data (mean, median, mode), and the dispersion degree (variance, standard deviation); through the analysis of housing price trends in a certain area and the distribution of students' academic performance and other practical cases, students can master the process and methods of data analysis, be able to extract valuable information, and make reasonable decisions.

4.2 Project-Based Learning

In project-based learning, project design is a crucial aspect that directly affects students' learning outcomes and the cultivation of their computational thinking skills. Project design should be carried out based on the principles of being close to life, being challenging, and having a certain degree of openness[17].

Projects closely related to daily life enable students to truly understand the practical significance of learning, and also stimulate their interest and enthusiasm in studying. For example, the "Campus Water Conservation Plan" project allows students to conduct on-site investigations of campus water facilities, calculate water usage, analyze the causes of water waste, and propose specific water conservation measures[18]. During this process, students apply the decomposition strategy in computational thinking, breaking down the water conservation issue into multiple sub-problems, such as understanding the current water usage situation, analyzing the causes of waste, and formulating water conservation measures, etc. By solving these sub-problems, they ultimately achieve the goal of campus water conservation. Such projects closely related to life enable students to apply the knowledge they have learned in real life, improve their ability to solve practical problems, and at the same time enhance their environmental awareness and social responsibility.

Challenging projects can stimulate students' curiosity and exploratory spirit, enabling them to fully utilize their thinking abilities and try new methods and strategies. For instance, the "Intelligent Robot Competition Project" requires students to design and build an intelligent robot capable of completing specific tasks, such as robot football competitions or robot treasure hunts[19]. During the project implementation, students need to apply programming knowledge, electronic circuit knowledge, and mechanical design knowledge to solve problems related to the robot's motion control, sensor application, and algorithm optimization. These tasks have a certain degree of difficulty and challenge, and students need to continuously think, try, and innovate in order to cultivate their innovative thinking and practical hands-on skills. At the same time, challenging projects can also help students develop their ability to withstand setbacks and solve problems. Through continuous efforts and attempts, students can ultimately overcome difficulties when encountering problems in the project, enhancing their self-confidence and sense of achievement.

Open-ended projects can provide students with a broad space for thinking, encouraging them to consider problems from different perspectives and propose diverse solutions. This is an open educational approach. For example, in the "Future City Planning Project", students can design the layout, transportation system, energy supply, etc. of the future city based on their imagination and understanding[20]. In this project, there are no fixed answers or standards. Students can fully exert their creativity and imagination and propose various novel ideas and solutions. Open-ended projects can also cultivate students' critical thinking and teamwork skills. During discussions and exchanges, students need to analyze and evaluate different solutions, express their own opinions and suggestions, and jointly improve the project plan through teamwork, thereby enhancing the quality and feasibility of the project.

4.3 Staff Team Building

Teacher training is extremely crucial and necessary, and it holds significant importance for enhancing the quality of cultivating students' computational thinking skills in primary and secondary schools. The training content should cover theoretical knowledge of computational thinking, enabling teachers to deeply understand the connotation, core elements, and significance of computational thinking. Through systematic learning, teachers can accurately grasp the differences between computational thinking and traditional thinking methods, as well as the important role of computational thinking in students' learning and future development[21]. For example, organizing teachers to attend special lectures on "Computational Thinking Theory", inviting experts and scholars to elaborate on the concept, development history, and application fields of computational thinking, so that teachers can comprehensively and thoroughly understand computational thinking, integrate it, and apply it in practice.

In teacher training, programming skills training is also an important part. In today's era of continuous development of information technology, programming has become an important means to cultivate students' computational thinking. Teachers need to master certain programming skills in order to better guide students in programming learning and cultivate their computational thinking abilities. The training can be targeted at teachers of different levels, providing courses ranging from basic programming knowledge to advanced programming applications. For teachers with weak programming foundation, they can first receive introductory training in programming languages such as Python and Scratch, allowing them to master basic programming syntax, data types, control structures, etc., while learning simple program design. For teachers with certain programming foundation, advanced programming courses such as algorithm design, data structures, and artificial intelligence programming can be offered to improve their programming skills and application abilities. Through the development and practice of actual projects, teachers can integrate programming knowledge with teaching practice, thereby better guiding students' programming learning.

Teaching method training is also indispensable. Teachers need to master advanced teaching methods that can stimulate students' interest in learning and improve teaching effectiveness. Training can be conducted through case analysis, simulation teaching, etc., allowing teachers to understand the implementation steps, precautions, and evaluation methods of these teaching methods. Taking project-based learning as an example, some successful project-based learning cases can be selected during the training, such as "Campus Culture Construction Project" and "Environmental Protection Theme Project", to analyze the design ideas, implementation processes, and students' learning outcomes of these projects. Teachers can learn how to design projects, how to organize students to carry out project practices, how to guide students to solve problems using computational thinking, and how to evaluate project outcomes, etc. At the same time, the training can also set up simulation teaching sections, allowing teachers to conduct simulated teaching of project-based learning in groups, observe and evaluate each other, and improve teaching ability in practice.

The training methods should be diverse to meet the learning needs of different teachers. Online training has the advantages of convenience and flexibility. The learning content and time can be chosen by the teachers themselves. An online learning platform can be set up to provide rich training course resources, including video lectures, online documents, interactive discussion areas, etc. By watching video lectures, teachers can learn theoretical knowledge while also acquiring computational thinking and programming skills; by consulting the relevant information in online documents, they can deepen their understanding of the knowledge points; in the interactive discussion area, they can exchange learning experiences and teaching methods with other teachers and solve problems encountered in the learning process. Offline training can provide face-to-face communication and practical opportunities, enhancing the effectiveness of the training. Concentrated training can be organized, inviting experts and outstanding teachers from the front line to give on-site lectures and guidance. During the training, practical operation sections can be set up, allowing teachers to apply the knowledge they have learned to practice through actual programming, teaching design, and other activities, improving their teaching ability. At the same time, teachers can also be organized for field trips to visit the teaching practices of other schools on computational thinking, learning advanced teaching experiences and teaching models.

5. Conclusion and Outlook

This article conducts an in-depth exploration of the cultivation paths for computational thinking skills among primary and secondary school students. Through multi-dimensional research methods and a wealth of practical cases, it has achieved a series of significant theoretical and practical achievements.

Looking to the future, the cultivation of computational thinking skills among primary and secondary school students will remain a continuously evolving research field. With the rapid development of information technology and the continuous emergence of new technologies, the connotation and application scenarios of computational thinking will also continue to expand and deepen. Future research should closely follow technological trends, promptly update and improve the content and methods of computational thinking cultivation, integrate emerging technologies into teaching practice, and cultivate students' computational thinking abilities to adapt to the development of future society. At the same time, cross-disciplinary research should be strengthened to further explore the integration models and methods of computational thinking with other disciplines, promoting the all-round development of students. Additionally, international exchanges and cooperation should be enhanced to draw on advanced computational thinking education experiences from abroad, combined with the actual situation of our country's education, to promote the continuous improvement of the cultivation level of computational thinking skills for primary and secondary school students in our country.

References

[1] Liu Yongyan. Exploration of Python Teaching Process Based on Computational Thinking Cultivation - Taking the Project-based Learning "Calculating the Total Deposit Amount" as an Example [J]. Anhui Education Research, 2025, (03): 56-58.

[2] Gao Hongyu, Wang Yanxi, Shen Weili. Key Development Indicators of Preschool Children's Computational Thinking in Programming Activities [J]. Fujian Education, 2024, (43): 20-22.

[3] Liu Jing. "Computational Thinking + Artificial Intelligence" Empowering University Computer Course Teaching Reform and Innovation [J]. Modern Vocational Education, 2025, (09): 137-140.

[4] Xie Lingpeng. DBL Learning Model Oriented towards Computational Thinking [J]. Primary Science, 2025, (06): 109-111.

[5] Wang Yangyu. Assessment Research on High School Mathematical Modeling with Intelligent Computational Thinking Orientation [J]. Middle School Mathematics Monthly, 2025, (03): 65-67.

[6] Pan Zhihong. Teaching Design and Evaluation of High School Information Technology Based on Computational Thinking Cultivation [J]. Primary and Secondary School Information Technology Education, 2025, (03): 63-64.

[7] Han Han. Research on the Cultivation of Computational Thinking Ability of Junior High School Students Based on Smart Classrooms [D]. Shandong Normal University, 2021.

[8] Sun Shuang, Wei Wenqing, Zhao Rui, et al. Investigation and Analysis of the Current Situation of Computational Thinking Levels of High School Students [J]. Primary and Secondary School Electrification, 2022, (10): 14-17.

[9] Liang Chen. Construction and Optimization of STEM Curriculum Development Model [D]. Zhejiang Normal University, 2023.

[10] Zhou Meiyun. Teaching Strategies for Primary School Mathematics Unit Whole Teaching Oriented towards Core Competencies [J]. Tianjin Education, 2025, (06): 26-28.

[11] Wang Hongmei. Research on the Design and Practice of Primary School Programming Club Activities under the "Double Reduction" Policy - Taking S Primary School in Chongqing as an Example [J]. Modern Primary and Secondary School Education, 2025, 41(03): 74-78.

[12] Zhang Xuan. Development Characteristics and Teaching Strategies of Mathematical Thinking Ability of Primary School Students of Different Ages [J]. Readings and Writing Calculation, 2025, (02): 37-39.

[13] Lv Yuanchun. Thoughts on the Cultivation of Mathematical Thinking Ability in Junior High School Mathematics Class under the New Curriculum Standards [J]. Mathematical World (Junior High School Edition), 2025, (06): 104-106.

[14] Dong Yawei. Construction and Practice Research on Teaching Model for Cultivating Computational Thinking of Primary School Students [D]. Qingdao University, 2024.

[15] Deng Jie. Research on Three-Dimensional Teaching Model for Computational Thinking Cultivation [D]. Yan'an University, 2024.

[16] Xie Lingpeng. Teaching Model for Computational Thinking [J]. Primary Science, 2025, (06): 109-111.

[17] Wang Xiaoting, Li Ruiping. Implementing Maker Education with Project-Based Learning Method [J]. Primary Science Education, 2025, (07): 10-12.

[18] Chen En. Design and Application of Cross-disciplinary Teaching Activities in Primary School Mathematics [J]. Reading and Writing Arithmetic, 2025, (06): 10-12.

[19] Huang Chaolong, Li Chengzheng. Analysis of the Project-Based Learning Model of VEX IQ Robot in Primary School [J]. Journal of Yuzhang Normal University, 2024, 39(05): 89-94.

[20] Yang Qingyuan. Construction of Interactive Teaching Mode in High School Geography Class [J]. College Entrance Examination, 2024, (36): 79-81.

[21] Zhang Chunhua. Research on the Promotion of Digital Literacy of University Teachers in the Digital Age: Foundation, Obstacles and Breakthroughs [J]. Journal of Hubei Open Vocational University, 2025, 38(04): 158-160.

Acknowledgments

Scientific research program funded by Shaanxi Provincial Education Department (Program No.23JS024); SPOC blended teaching reform and practice based on OBE concept (SGH23Y2625); Construction of Intangible Cultural Heritage Database and Information System in Weinan City (2024GFY03); Application Research on Comprehensive Management System for University Scientific Research Business (2024QCY-YJ04); Investigation and Countermeasure Research on the Current Situation and Basis for Education Services of Local University Teachers in the Digital Age for Primary and Secondary Education (SGH24Y2346); Comprehensive Management System for Science and Technology Business and Science Project Management in Weinan City (2021HXXM329).

Author Profile

Daiwen WU, male, born in 1979, of Han ethnicity, from Hengyang, Hunan Province. He is an associate professor at Weinan Normal University and holds a master's degree. His main research field is educational software development