Teaching Design and Implementation of Design Mode in the Course of Java Programming in Higher Vocational Colleges

Jun Zhang
Sichuan Judicial and Police Officers Professional College DeYang, Sichuan, China 618000
2014326215@dlvtc.edu.cn

Abstract—Java programming is one of the important courses for computer majors such as software technology in higher vocational colleges. The course has strong logic and abstraction, and is rich in content and practical. When studying this course, students should not only master the Java language and object-oriented programming ideas and methods, and cultivate programming habits, but also learn to use design patterns to efficiently solve real-world problems. In order to achieve the expected teaching goals in the limited class hours, universities must carry out teaching design work. This article will put forward some views on how to optimize the software design model teaching design and teaching activities based on the analysis of related concepts and teaching status.

1.INTRODUCTION
Java programming is a course based on Java language learning, the basic ideas, concepts and programming methods of object-oriented design. It includes basic knowledge of Java language, object-oriented programming, network programming and database programming and other learning content. Before studying this course, students have completed the C language and C++ predictive programming, and they have a certain understanding of programming. The teaching purpose of this course is to enable students to master the object-oriented programming technology, ideas and methods, and cultivate the ability to participate in project development and design. Students will develop the habit of programming to lay a good foundation for other courses in the future.

2.CURRENT TEACHING STATUS OF JAVA PROGRAMMING COURSE
In order to solve the gap between the quality of talent training in higher vocational colleges and the demand for talents in the market, all higher vocational colleges have carried out optimization of professional settings and reforms in curriculum teaching. Teaching methods such as action-oriented and project-based teaching have emerged in universities and colleges, emphasizing the transformation of teaching concepts that are teacher-led and student-oriented. But there are still many problems in the teaching of design patterns in Java programming courses.

First of all, software design patterns are an important part of the Java programming curriculum. It has the characteristics of abstraction and strong practicality, but there are few teaching hours according to the teaching plan. This makes many teachers compress teaching content and practice time to complete tasks. Although most teachers adopt an integrated teaching and learning approach, they basically attach importance to the explanation of basic grammar, concepts and example verification,
and students have very little time to design by themselves. Besides, many higher vocational students have poor professional foundation and weak learning initiative, which may cause students to lose their enthusiasm for learning in the early stage, and they cannot solve the problems encountered in the later practice. In addition, in learning evaluation, usually a combination of usual assessment and final assessment. However, the average scores and practical assessment scores account for a very small proportion, which cannot accurately reflect the knowledge and skills of students. Especially in the experimental class, the teacher can not go to the site to check and accept, but only evaluate the test results. In this way, not only will there be a problem of cheating, but also because of ignoring the inspection of the process, it cannot truly reflect the individual situation of the students and cannot provide a basis for teaching optimization. In practical teaching methods, teachers often verify many cases or use case questions in textbooks. However, there is no connection between the cases, and the students lack the actual project development experience and the necessary skills for development, such as program debugging and training in the use of development tools. This will not really improve students' skills, and it will not cultivate students' good programming habits.

3. DESIGN PATTERNS IN THE COURSE OF JAVA PROGRAMMING IN HIGHER VOCATIONAL COLLEGES

3.1. Types of Software Design Patterns
The software design pattern refers to a set of classification and code design experience that can be used repeatedly and known by many people. That is, the software design experience summarized by the predecessors has the characteristics of reusability, maintainability, readability and safety. Software design modes include creation mode, structure mode and behavior mode. Among them, the creation model includes abstract factory model, creator model, prototype model and singleton model. Structural modes include adapter mode, decorator mode, proxy mode, bridge mode, combination mode, etc. Behavioral modes include strategy mode, command mode, iterator mode, observer mode, state mode, interpreter mode, etc. [1].

3.2. Application Principles of Software Design Patterns
The design pattern uses the characteristics of object-oriented encapsulation, inheritance and polymorphism, and summarizes the corresponding software design methods based on the principle of object-oriented design. The related principles of design patterns include the principle of opening and closing, the principle of single responsibility, the principle of interface isolation, the principle of dependency inversion, the principle of composition and aggregation and reuse.

3.3. The Purpose and Difficulties of Learning Software Design Mode for Vocational College Students
High vocational college students learn the software design mode in the Java programming course for the purpose of using the corresponding design mode reasonably when solving practical problems based on learning the language and algorithms. This can help students quickly solve most of the problems with messy code and difficult dimensions, and focus more on business processing. The use of design patterns makes the code not only reusable, but also easier to understand by others, and it also has higher reliability and security. As the basis of software engineering, software design mode makes software project code engineered, which is conducive to maintaining the code and improving the software life cycle. Various design patterns are applications of object-oriented programming. At the same time, following the design principles, engineers will use certain design patterns after the amount of code reaches a certain level. This will make them more conducive to coordination and communication with the team.

For students in higher vocational colleges, the course of Java programming is very difficult to learn, and students need to learn a lot. Otherwise, there are dozens of software design patterns, and it is much more difficult for students to learn the content without a foundation. In addition, some colleges and universities are restricted by class hours, and they do not even offer special courses related to software
design patterns. This makes it difficult for students to truly master the principles and methods of software design patterns. Students can understand some common design patterns in the process of learning OOD and OOP when object-oriented programming, and use these design patterns in future programming work. For teachers, it is necessary to do a good job in teaching design and improvement and optimization [2].

4. Teaching Design Patterns in Java Programming Courses in Higher Vocational Colleges

4.1. Choose a Design Mode
Because the class hours of Java programming courses in higher vocational colleges are very limited, and there is a lot of programming-related content that needs to be learned in this course, it is simply unrealistic for students to fully understand dozens of design patterns. As a result, teachers should reasonably choose representative and frequently used design patterns to carry out teaching work according to the students' learning situation and future development needs. Teachers can choose singleton mode, adapter mode, factory mode and observer mode to explain. Afterwards, teachers can learn design patterns through API to make students' understanding of these several software design patterns more in place and in-depth.

4.2. Design Mode Teaching Design and Implementation
In the actual implementation of design pattern teaching, we should first briefly introduce the design pattern, and then use UML class diagrams to show the relationship between APIs. Considering that it is very difficult for students to understand many streams and classes when learning I/O streams, teachers can use the decorator pattern and adapter mode commonly used in I/O stream related classes as examples to carry out teaching work.

4.2.1. Brief Introduction to Design Patterns
We take the adapter mode and the decorator mode as examples to briefly introduce the meaning, types, and characteristics of these two software design modes. The adapter software design pattern is to transform a certain type of excuse into another interface representation of the client. Its function is to eliminate the compatibility problem of the class due to the unsuitable interface. The adapter pattern includes the adapter pattern of the class, the adapter pattern of the object and the adapter pattern of the interface. among them. The adapter pattern of a class is when one class is transformed into another class with a new interface that is adapted. We can use the class adapter pattern to create a new class, which inherits the characteristics of the original class. The adapter mode of an object is that when an object is transformed into an object that satisfies another new interface, it will create a wrapper class and hold the characteristics of the original class. The adapter mode of the interface means that when you do not want to implement all the methods in an interface, all methods are realized through the establishment of abstract wrapper classes, and the abstract classes are inherited when writing other classes. The decorator pattern refers to adding new functions to an object, and it has dynamic characteristics, requiring the decorated object and the decorated object to implement the same interface. Compared with inheriting the original class, the decorator mode is more flexible when it needs to extend the functionality of a class or dynamically add functionality to an object. This not only avoids a bloated inheritance system, but also reduces the coupling between classes.

4.2.2. UML Class Diagram
In the adapter mode, the UML class diagram is shown in Figure 1. Among them, InputStream is a byte stream, which will be read according to bytes. It does not have the function of reading character streams according to characters. The InputStreamReader class inherits the reader class but holds the original class, that is, an instance of the InputStream class, which will expand the functions of the InputStream class. In the decorator mode, the UML class diagram is shown in Figure 2 below. It will use BufferedReader and InputStream...
to add buffered input and support mark and reset methods to the inputstream byte stream to decorate the inputstream stream. At this time, inputstream becomes a super class, fileinputstream is a decoration class, filterinputstream is a decorator, datainputstream, buffered, and inputstream are a decoration class [3].

4.2.3.API Application
The JavaI/O package includes four abstract classes. Among them, inputstream and outputstream are classes for processing byte streams, and reader and writer are classes for character stream processing. Most of the I/O stream classes are decorator software design patterns, but inputstreamreader and outputstreamreader are adapter patterns.

5.ON THE OPTIMIZATION OF TEACHING DESIGN OF SOFTWARE DESIGN MODE COURSE IN HIGHER VOCATIONAL COLLEGES

5.1.General Mode of Instructional Design
The main purpose of teaching design is to effectively solve design teaching problems and improve teaching efficiency and quality. Instructional design mode is not only the visualization of instructional design theory, but also the summary of instructional design practice activities. The teaching design model is composed of teaching objects, teaching goals, teaching strategies, and teaching evaluation. It can be seen from this that, in order to make the teaching design effective, the teaching object must first be analyzed. Teachers need to understand the students' learning ability and learning situation, and on this basis, formulate scientific and reasonable teaching goals, design teaching strategies and teaching evaluations.
5.2. Analysis of Learning Needs in Instructional Design

Before developing instructional design, teachers must first understand the learning needs of students. Teachers can analyze the deficiencies in the current software design model teaching through investigation and research, analyze the nature and main reasons of the problem, and judge whether the current teaching system design can solve actual problems. In the meantime, teachers need to analyze the existing teaching resources and constraints, and demonstrate the possibility and difficulty of problem solving. As far as design pattern teaching is concerned, it is a simple and efficient solution to a specific problem in the process of object-oriented software design. It can include dozens of design patterns in the object-oriented software architecture. The function of these design patterns is to make the software reusable. Students should understand the basic concepts, principles and methods of commonly used software design patterns, and make the courses vivid through examples. Through theoretical lectures, teachers enable students to clearly understand the relevant concepts, types, characteristics, and principles of design patterns. This will enable students to use appropriate design software patterns to write codes and solve practical problems [4].

Since every student in a higher vocational college has different basic abilities and learning levels, the current learning situation and learning goals of students are different. The analysis of students is to understand their actual situation and learning goals. These analyses should include the analysis of psychological characteristics and the analysis of learning styles and acquired abilities. The results of the analysis will serve as the main basis for teaching design.

5.3. Teaching Goal Design

The study of the content of design patterns in the Java programming course is mainly to impart basic knowledge about software design and methods to the students of computer technology majors in vocational colleges, and cultivate their scientific thinking ability and software development ability. This requires students to master the general process of software design, software design principles, and understand the concepts, types, and methods of design patterns in common use. Students need to strengthen their object-oriented software design and development ideas and improve their software system architecture design capabilities.

Teachers must clarify the above-mentioned teaching objectives before teaching design, and make the objectives precise, so as to avoid any ambiguity problems. Teachers should use the learning result classification method to analyze the teaching target components when describing the goals. Teachers can analyze the computer information skills, cognitive strategies, and motor skills that students should acquire, and describe learning objects, learning behaviors, conditions and standards. This will help to achieve a precise description of the teaching objectives and provide standards for later teaching evaluation. When students in higher vocational colleges study the course of Java programming, some have a certain foundation, some need in-depth and systematic study, and some have to deal with the pressure of employment. At this time, students need to strengthen practical teaching. Teachers can rationally design course teaching content according to society's demand for talents, so that students can learn more basic skills and improve their operational capabilities. The clearer the learning goal, the more the students' practical application ability can be improved, which can also lay a good foundation for the future work. In teaching design, teachers should abandon the traditional rote memorization of the past, but should link theory with practice. In the teaching process, teachers should adopt different teaching strategies according to different classes. Students with a good foundation can improve their practical design application ability on the basis of their systematic mastery of theoretical knowledge. For students who are about to face employment, they should focus on students as the center, and strengthen their ability to analyze and solve problems based on their mastery of basic theoretical knowledge and methods. This not only pays attention to the effect of actual learning, but also makes every teaching activity effective.
5.4. Teaching Strategy Design
The teaching strategy is designed to solve the problem of how to teach. Only by adopting a reasonable teaching strategy can the teaching goal of the design mode be successfully completed. Traditional teaching strategy is based on teacher teaching as the center to carry out teaching design, which focuses on how to impart knowledge and skills to students, or in what form. This kind of teaching strategy design focuses on the transmission of knowledge, but ignores the actual situation of students. In order to ensure the effectiveness of teaching, teachers must adopt multiple teaching methods. For example, teachers need to combine group teaching and group cooperative learning, and guide students to adopt personalized learning methods according to their actual conditions. Teachers should adopt multiple teaching methods. For example, in addition to teaching knowledge, teachers can also use collaborative discussions. This can deepen the impression through interaction in the classroom, and enhance the ability of solving practical problems through cooperation among students. Furthermore, teachers can combine case analysis method, project method, etc., and modern media to improve the effectiveness of teaching [5].

5.5. Teaching Media Design
In the selection of teaching media, in addition to books and classroom blackboard explanations, it also includes various online learning resources, learning tools and learning spaces. Teachers can provide students with self-study materials, electronic teaching plans and multimedia courseware, etc., and students can realize long-distance interactive learning through computer networks. Teachers can also influence students' learning activities through interpersonal communication, and teachers become participants in group learning. With the emergence of various new media, teachers can use existing resources to expand teaching media resources in terms of teaching resource construction. Teachers can deploy FTP for resource sharing among students of the school, and students outside the school should create network disk files with sharing attributes for resource sharing.

5.6. Teaching Process Design
The design of the teaching process of software design mode includes specific lesson preparation, content innovation, method innovation, etc. The purpose is to improve teaching efficiency and quality. First of all, teachers should let students clarify the purpose of learning software design patterns and stimulate their desire for learning. Simultaneously, teachers need to inform students of the difficulties of learning and draw their attention. Secondly, before teaching new content, teachers must make a pre-class introduction design, by asking questions, students' self-study research and collaborative discussions, recalling and consolidating existing knowledge and skills, and preparing for the next new content learning. In classroom teaching, teachers need to combine knowledge concept explanation, questioning, group cooperation, teacher-student interaction, practical operation, project cooperation research, etc. Using multimedia courseware to operate teaching content on-site can quickly and systematically transmit new learning content to students. In order to get feedback from students in time, teachers can intersperse questions in the classroom to understand their learning situation, and then arrange corresponding design and practice activities according to the actual situation. At the same time, teachers need to provide abundant teaching resources to guide students to learn independently outside of class. In addition, teachers should listen more to students' suggestions after class. Afterwards, teachers need to do a good job postscript according to the actual situation of the classroom, and make timely improvements to the original teaching plan and teaching methods. Finally, teachers should pay more attention to practical teaching to improve students' practical design skills. Teachers need to design experiments, provide corresponding guidance for students' experiments, guide them to complete tasks independently, and cultivate their innovative and practical ability. In terms of experimental teaching, teachers should appropriately broaden the teaching content, and explain the content, process and requirements of the experiment before the experiment. In the experiment, teachers should check the projects completed by students in time, urge them to actually do it and analyze the results. The evaluation of the experiment should be based on the process, and comprehensively examine the attitude,
skills, proficiency and completion of students. Teachers should increase designed experiments and put forward requirements and specific experimental results in the form of projects. This requires students to find relevant information first, and then complete the project based on the knowledge and skills they have learned to improve their comprehensive application ability.

6. CONCLUSION
In summary, object-oriented computer programming languages will use design patterns in their software design. There are also many types of design patterns, as a set of experience, a kind of thought, but also a kind of technology. Choosing a suitable design pattern can quickly solve real-world problems and solve coding chaotic and unstable problems, which helps to complete the work quickly and efficiently. Each design pattern has its scope of application. For students in higher vocational colleges, when studying design patterns, the main purpose is to understand and master object-oriented software design ideas, and be able to choose appropriate design patterns to solve practical engineering problems in future work. For teachers, their main task is to carry out a reasonable design of design mode courses, and teachers should adopt various teaching methods in actual implementation. Teachers can improve the quality of teaching through the optimization of curriculum teaching design, so that students can truly master the concept and method of design mode.

REFERENCES
[1] Yin Ziming, Zhou Lei, Zheng Jianli. Use "software design mode" to promote the study of "object-oriented programming" course teaching method [J]. Software, 2019(8): 216-219.
[2] Gou Jianping, Ke Jia, Zhu Huijuan. On the Teaching of "Design Mode" [J]. Educational Modernization, 2019(95): 169-170.
[3] Zheng Xiuqin. Teaching design and implementation of design mode in Java programming courses in higher vocational colleges [J]. Information and Computer, 2020(9): 254-256.
[4] He Lifu. Analysis on the Teaching Reform of "Java Programming" in Higher Vocational Colleges [J]. Computer Knowledge and Technology, 2016(32): 107-113.
[5] Hou Xiangru. The application of project teaching method in JAVA programming course [J]. Electronic Testing, 2016(2): 112.