Abstract. According to the difficulties of learning the object-oriented concept of Java language for beginners, the paper mainly discuss the understanding and using of object-oriented respectively through three methods. The first method is to create an object through a no-parameter constructor, and then assign the object through a variety of ways. The second method is to create and initialize objects by parameterized constructor. The third method is same to the first method when creating objects, but their assignment methods are different. Choose which one method or combined methods, according to the specific situation of specific analysis. The experiments have proved that the comprehensive application and comparison of the above three methods can greatly improve the students' ability to understand object-oriented.

Introduction

Nowadays, along with the extensive use application of the Java language [1], all universities have opened Java courses. The main of the Java language programming course is object-oriented, and understanding the concept of object-oriented is the key issue for us to use Java language programming [2]. In general circumstances, university courses, especially the students major in computing, first learn the C language and then learn the Java language, but the C language is process-oriented, different from the Java language which is object-oriented completely. Therefore, for beginners, the basic part is easy to understand, but the object-oriented part is notoriously impenetrable, it is difficult to use Java language programming. If the students cannot profoundly understand the use of the class and object in the Java language, they cannot write the programs independently [3].

In order to improve the student's hands-on practical ability and capacity of the object-oriented part of Java language, this paper mainly studies the understanding and use of the object, and the code of the specific student example is given to show the process of generating object cases in detail.

Code instance and Parsing

Firstly, Define a Simple Student Class

```java
package studentManager; //define a package
public class Student {  // define a Student class
    private long no;  // define a student no
    private String name; // define a student name
    private char sex;  // define a student sex
    private String major; // define a student major

    /* Defines a constructor without parameter, do nothing in generally, or perform certain actions unconditionally when creating objects. */
    public Student() {
    }
}
```
public Student(){} 
/* Defines a constructor with parameters, which are used to assign member variables in generally. */
public Student(long no,String name,char sex,String major){
/* Because the parameter name is the same as the member variable name, the name of the member variable needs to be invoked through this. 
this: references to themselves objects 
If the parameter name is different from the member variable name, it does not need to be invoked through this. */
this.no=no;
this.name=name;
this.sex=sex;
this.major=major;
}
/* Defines a member method 
The method is similar to the constructor Student(long no,String name,char sex,String major) with parameters, assigns all of the member variables 
Difference: the constructor with parameters initializes the member variables when creating the object, but the method creates the object through the no-parameter constructor first, and then assigns the member variables. */
public void setStudent(long no, String name, char sex, String major){
this.no=no;
this.name=name;
this.sex=sex;
this.major=major;
}
//Setters of the member variables 
public void setNo(long no) { this.no = no; } 
public void setName(String name) { this.name = name; } 
public void setSex(char sex) { this.sex = sex; } 
public void setMajor(String major) { this.major = major; } 
//Getters of the member variables 
public long getNo() { return no; } 
public String getName() { return name; } 
public char getSex() { return sex; } 
public String getMajor() { return major; } 
/* overrides the toString() method in the parent Object class, print all instance objects generated by the class according to some format and require */
public String toString(){
return "XueHao: " + no + "XingMing: " + name + "XingBie: " + sex +"ZhuanYe: " + major;
}
}

Secondly, Define a TestStudent Class

Print the value of the object through the different object creation method and assigned method.
package exam
public class TestStudent {
public static void main(String[] args){
    /* The first method:
    Create an object stu1 of the Student class through the no-parameter constructor
    */
    Student stu1 = new Student();
    stu1.setStudent(2018001, "ZhangSan", 'M', "computer");
    System.out.println("XueHao:" + stu1.getNo() + "\nXingMing:" + stu1.getName()
                      + "\nXingBie:" + stu1.getSex() + "\nZhuanYe:" + stu1.getMajor());
    /* The execution results of the upper codes:
    XueHao:2018001
    XingMing:ZhangSan
    XingBie:M
    ZhuanYe:computer
    We can assign all of the member variables by invoking setStudent() method of the object stu1, and
    acquire the values of the member variables by invoking Getters methods of the object stu1
    */
    System.out.println(stu1);
    /* The execution results of the above line Code is same to the last one.
    It can be found that we can achieve the purpose of obtaining the value of the member variable of the
    object stu1 through printing the object stu1 and calling the Getters methods of the object stu1.
    Printing the object stu1, the system calls the overload method toString() in Student class
    automatically to print the value of the member variables of the object stu1.
    In the overload method toString(), a member variable name can be directly referenced to get the
    value of the member variable in the object stu1, because the definitions of the overload method
    toString() and the member variables names is in the same class of Student;
    In this code, we need to get the values of all member variables of the object stu1 by Getters method,
    because the member variables names is private variables, and the object stu1 and the definitions of the
    member variables names are not in the same class, so the member variables names cannot be
    referenced directly, it must be acquired by calling the public Getters methods in the Student class
    through the object stu1.
    */
    /*
    The second method:
    Create an object stu2 of the Student class through the constructor with parameters
    */
    Student stu2 = new Student(2018002, "LiSi", 'M', "English");
    System.out.println("XueHao: " + stu2.getNo() + "\nXingMing: " + stu2.getName()
                      + "\nXingBie:" + stu2.getSex() + "\nZhuanYe: " + stu2.getMajor());
    /* The execution results of the upper codes:
    XueHao: 2018002
    XingMing: LiSi
    XingBie: M
    ZhuanYe: English
    In the above codes, we can assign all member variables of the object stu2 uniformly through the
    constructor Student with parameters, and acquire the values of all member variables of the object stu2
    through the Getters methods.
    */
    System.out.println(stu2); // the print results same as above
    /* The third method:
    Create an object stu3 of the Student class through the no-parameter constructor Student(), and
    assign all member variables of the object stu3 through the public Setters methods of Student class,
    and acquire the values of all member variables of the object stu3 through the Getters methods.
    */
star
Student stu3=new Student();
stu3.setNo(2018003);
stu3.setName("WangWu");
stu3.setSex('F');
stu3.setMajor("Chinese");
System.out.println("XueHao:"+stu3.getNo() + ", XingMing:"+ stu3.getName() + ", XingBie:" + stu3.getSex() + ", ZhuanYe:" + stu3.getMajor());
/* The execution results of the upper codes is similar to the second method. */
System.out.println(stu3); // the print results same as above
}
}

In the program, the function of the setStudent method and the constructor Student with parameter is similar, but the calling methods are different, because the constructor is a member method used when creating an object of this class, its main function is to initialize variables, and the constructor name same as the class name, and has no return type, and so can be created directly by the keyword new, however, the setStudent method is an ordinary member method, its name is different from the class name, it has a return type, and needs to be invoked by the object created by the class.

The methods of Setters and Getters is respectively to assign or acquire the values of the object's member variables of the class, that is to say, we can assign or acquire the value of some object's any member variable alone through this method, or modify the value of the object's one member variable. It is no possible to realize for the setStudent method and the parameterized constructor Student().

Of course, a class may have many constructors, but their parameter list is different, so that you can create and initialize the object through parameters of different formats, the member variables which have not been initialized, can be assigned alone by the Setters methods.

From this experiment, it can be found that we can obtain the same printing results of the object through different object creation methods and assignment methods.

Experimental Results and Summary

The paper mainly discusses three methods of understanding and using of object-oriented. The first method is one of the most commonly used methods, which create an object through a no-parameter constructor, and then assign the object through a variety of ways. The second method is to create and initialize objects by parameterized constructor, which is used when the specific value of an object is explicit. The third method is same to the first method when creating objects, but their assignment methods are different, the first method assigns all member variables of the object unified through the set method, but the third method is respectively to assign some or all member variables of the object independently by the Setters methods. And then, the printing ways of the objects can be different too, the one is unifying the printout format by the override method toString(), the other one is customized. Choose which one method or combined methods, according to the specific situation of specific analysis.

According to the integrated use of the above three methods, through the practice of teaching reform on the sophomore of two classes who majored in the E-commerce and two classes who majored in the electronic information in the department of Computer Science of ZhiXing College of HuBei University., the “Java language programming course” is their required courses, which has achieved good teaching effect, most of the students are able to compile the student management system, and the similar system of the books management system to improve themselves, according to the example of the student management system program to compile the books management system program [4].

According to the students’ use these three methods compile varies of different management systems [5], most of the students have learned to draw inferences about other cases from one instance, and have a further understanding of object-oriented programming methods.
Through the teaching practice, it is time to summarize the experimental data as the table 1.

<table>
<thead>
<tr>
<th>methods</th>
<th>Class</th>
<th>electronic information class 1</th>
<th>electronic information class 2</th>
<th>E-commerce class 1</th>
<th>E-commerce class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>total numbers</td>
<td>45</td>
<td>49</td>
<td>30</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>classroom</td>
<td>teaching effectiveness</td>
<td>excellent</td>
<td>good</td>
<td>good</td>
<td>excellent</td>
</tr>
<tr>
<td>raised number</td>
<td>41</td>
<td>20</td>
<td>18</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Ratio (%)</td>
<td>83.67</td>
<td>66.67</td>
<td>47.36</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>computer room</td>
<td>teaching effectiveness</td>
<td>excellent</td>
<td>good</td>
<td>good</td>
<td>excellent</td>
</tr>
<tr>
<td>raised number</td>
<td>40</td>
<td>21</td>
<td>19</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Ratio (%)</td>
<td>81.63</td>
<td>70.00</td>
<td>50.00</td>
<td>94.44</td>
<td></td>
</tr>
</tbody>
</table>

It can be seen from table 1:
1. It has a great relationship with the class management on improving the ability of the students;
2. The learning basis and the learning consciousness of the undergraduate are stronger than the specialized students obviously, so the effect on curriculum reform of the undergraduate (electronic information class 1 and 2) is superior to the specialized students (E-commerce class 1 and 2);

Therefore, the comprehensive application of the above three methods can deepen the students’ comprehension of the concept of object-oriented obviously, and further improve the students' ability to learn the "Java language programming course", so that the teaching quality can be guaranteed and further improved.

Acknowledgement
This research was financially supported by the Education Department of Hubei Province Science and Technology Research Project—the recognition research of the iris biologic characteristics based on LBP mode (No: B2018406).

References