Reflection word

Introduction

CA3 will complete the Booking application, is divided into four classes Classroon, Lecturer, Booking, ClassroomBookingApp, corresponding reflection document will be divided into three parts, respectively is learned, The challenges and unsolved screenshots are explained as follows.

1. Things you have learnt through doing the assignment.
2. Most programming will basically carry out the variable definition first, the variable name is the unique name of the variable, and the variable value is referenced in JAVA, but there will be some rules such as not starting with a number, you can use underscores, letters and numbers, and the same word has different meanings.
3. Data types (such as integers, floating-point numbers, strings, characters, booleans, etc.) determine the range of data that a variable can store, numeric types are divided into integer types (byte,short,int,long) and floating-point types (float,double), character types are char and String, escape characters, and so on. Boolean type is judging right or wrong as a table.

|  |  |  |
| --- | --- | --- |
| Basic data type |  |  |
| Numerical type | Integer type （byte,short,int,long) | Symbol type(float,double) |
| Character type | char(single character）  String(string） | Escape character(\’,\\,\n,\t,\r) |
| Boolean type | boolean(true or false) |  |

1. Most variables will have an initial value, depending on the situation (e.g. int age = 10).
2. Modifiers In this operation, the most commonly used are private and public, both of which are access control modifiers. The difference is that private, which restricts access to class members, can only be accessed by other members inside the class and cannot be accessed externally. public allows broad access to class members. The details are shown in Table below.

|  |  |  |
| --- | --- | --- |
|  | private | public |
| Access range | Only inside the class | Unlimited access from anywhere |
| Example | private int age(age is only accessible inside the class) | public void display( ){...}(The display method can be called anywhere) |

1. The declaration of a constant, which can only be assigned once in the entire program, is qualified by the "final" keyword (for example: final int LENGTH =6;).
2. This group of code also uses the constructor method, which is mainly used to initialize the state of the object when creating the object, and also uses the keyword "this" to distinguish. The constructor method is divided into parameterless constructors (default initialization) and parameterized constructors (providing specific initialization values). The difference between parameterized constructors and parameterless constructors is shown in Table .

|  |  |  |
| --- | --- | --- |
| Parameter construction method | public Classroom(int classroomNum, String classroomName) {  this.classroomNum = classroomNum;  this.classroomName = classroomName;  } | Initializes the member variable with the parameters passed in |
| No parameter construction method | this,make = “Unkown”;  this.year = 0; | Provides default initialization values |

1. The CA3 code also uses the get & set method for many times. The following table is the concept and difference between get and set. get is used to obtain the attribute value of the object, and getName() is used to obtain the attribute value named name. set is used to set the property value of an object, and setNanme(String name)- sets the property value named name, which will be used in the UML diagram

|  |  |  |
| --- | --- | --- |
|  | get | set |
| concept | Used to get the attribute value of an object | Used to set the property value of an object |
| Naming convention | getName( )--used to get the value of an attribute named name | setNanme(String name)--Sets the value of an attribute named name |
| Example | public int getClassroomNum() {  return classroomNum;  } | public void setClassroomName(String classroomName) {  this.classroomName = classroomName;  } |

1. Overloading allows multiple methods with the same name to be defined in the same class, while overwriting is a process in which subclasses rewrite methods inherited from the parent class, as shown in Table .

|  |  |  |
| --- | --- | --- |
|  | overloading | Rewrite |
| Scope | It happens in the same class | Occurs between a child class and a parent class |
| Example | Overloaded addition method, two int numbersPublic clas Result{  public int add(int a,int b){  return a+b;  }  Overloaded addition method, two double numbers public double add(double a,double b){  return a+b;  } | The Dog class does this by overriding the set method of the Aninal classclass Animal{  void eat( ){  System.out.println(“Animals need to eat“）；  }  }  class Dog extends Animal{  @override  void eat( ){  System.out.println(“Dog need to eat”);  }  } |

1. Selection statement

|  |  |  |
| --- | --- | --- |
| The if statement, | The if…else statement | The switch statement |
| either performs (selects) an action, if a condition is true, or skips it, if the condition is false. | performs an action if a condition is true and performs a different action if the condition is false. | performs one of many different actions, depending on the value of an expression. |

The if statement is a single-selection statement because it selects or ignores a single action (a single group of actions) .

The if…else statement is called a double-selection statement because it selects between two different actions (or groups of actions).

The switch statement is called a multiple-selection statement because it selects among many different actions (or groups of actions).

1. Iteration statements also called repetition statements or looping statements, that enable programs to perform statements repeatedly as long as a condition (called the loop-continuation condition) remains true.The while and for statements perform the action (or group of actions) in their bodies zero or more times.If the loop-continuation condition is initially false, the action (or group of actions) will not execute.

|  |  |  |
| --- | --- | --- |
| iteration statements |  |  |
| while | If the condition is true, the body of the loop is executed; otherwise, the loop is exited | while(condition){  //Body of loop  } |
| do…while | Execute the loop at least once, and then check the condition | do{  //Body of loop  }while (condition); |
| for | Similar to the while loop, the number of iterations is known | for (initialization; loopContinualtionCondition; increment){  //Body of loop  } |
| enhanced for | (The for--each )loop simplifies iteration | for (type element: arrayOrCollection){  //Body of loop  } |

1. Declaring and Creating a Scanner to obtain user input from the keyword import java.util.Scanner;

Scanner input = new

Scanner(System.in);

int number1 = input.nextInt();

1. The method of output

System.out.print( )--No line wrap

System.out.println( )--line wrap

System.out.printf( )--Format the output, using the format specifier

1. Constructor call

Those parentheses “( )” in combination with a class name represent a call to a constructor.

// create an Account object and assign it to myAccount

Account myAccount = new Account( );

1. Instance variables exist before methods are called on an object, while the methods are executing and after the methods complete execution.Instance varivables are declareed inside a class declaration but outside the bodies of the class’s methods.
2. A class normally contains one or more methods that manipulate the instance variables belonging to particular objects of the class.
3. The break statement--Interrupt loop

The break statement, when executed in a while, for, do…while or switch, causes immediate exit from that statement.

1. The continue statement--Execute next loop

Instead of jumping out of the loop immediately, skip the loop.Code comments are split into single-line "//" comments, such as private String classroomNumber. // Define a String to hold classroomNumber; And the multiline comment "/\* \*/", everything between the symbols "/\*" and "\*/" is comment content.

1. An array is a collection of data with the same data type (for example, the collection of ball games - football, basketball, badminton, etc.). The array can be divided into one-dimensional array and two-dimensional array. The one-dimensional array is essentially a linear collection of data of the same type. Array element type Array Name []; Array element type [] Array name; After declaring an array, you must specify the length of the array (e.g., int arr[]). arr= new int[5]; The index of the array starts at 0. There are also two ways to express two-dimensional arrays: type of array element array name [] []; Type of array element [] [] Array name; For example, to create a two-dimensional array with two rows and four columns, int a=new[2][4].
2. The Scanner class reads the value that the user types.Instead of a single line of code that you can use directly, it uses the Scanner class.System.out means output to the console, and System.in means input from the console.To use Scanner, you have to import it: import java.util.Scanner; For example, the common method name nextInt(), which returns an int, and the method name nextLine(), which returns a String, use Scanner to scan the code in the console as :Scanner sc=new Scanner(System.in);
3. primitive types and reference types.All nonprimitive types are reference types.So classes, which specify the types of objects, are reference types.1. primitive types A primitive-type variable can hold exactly one value of its declared type at a time.2. reference types Programs use variables of reference types (normally called references) to store the locations of objects.Such a variable is said to refer to an object in the program.
4. Conditions in if statements --TWO (2) types of Operators.​Equality operators (== and !=)andRelational operators (>, <, >= and <=)

|  |  |
| --- | --- |
| Algebraic Oprator | Java equality or relational operator |
| equality operators |  |
| = | == |
| ≠ | != |
| relational operators |  |
| > | > |
| < | < |
| ≥ | >= |
| ≤ | <= |

1. Challenges faced in programming or group work.
2. Understand what the user thinks about the input

The user input phase will be designed in the code, and the challenge this time is to imagine what the user is typing, to respond clearly to different operations, and to guide the user to enter correctly, such as the input does not match the error prompt. The assumption of user input cannot be limited to one possibility. Errors can result in poor user experience, functional errors and even time spent verifying data. For text input, you may need to use string handling functions to parse the input data, such as splitting, finding, replacing, and so on, and to convert the input strings to data types used internally by the program (such as integers, floating point numbers, dates, and so on). Second, you can limit the range of user input by providing predefined options (such as pull menus, radio buttons, etc.), making it easier to understand user intent to solve such problems.

1. Teamwork ability

Teamwork can face individual differences, unclear assignment of tasks (team members may have challenges such as poor communication, duplication or omission of work), and technical mismatches (everyone is good at different parts and has different ability levels and some writing challenges). Such problems can be solved from the following aspects: 1. Clarify the communication mechanism, establish clear communication channels, and ensure that team members can communicate and share information in real time. Hold regular meetings to clarify project goals and everyone's responsibilities, synchronize information, solve problems, and plan for future work. Team leaders encourage team members to actively express their ideas and opinions to reduce misunderstandings and information lag. 2. Define the work process and role, formulate a clear work process, and work in accordance with the prescribed process, covering the details of task allocation, execution, inspection and approval, so as to improve work efficiency and convenience. Clarify the division of labor and tasks of each member of the team to avoid duplication of work and unclear responsibilities. Reasonable division of labor according to the ability level of each member in the group to ensure that the task can be efficiently completed.

(3) The code is not uniform

Different systems and software may have inconsistent codes, such as windows computers and Apple computers. Differences in system configuration and hardware of different computers will affect the running results of codes. Some code will depend on a specific file or directory structure, and there will be a situation where the code synthesis cannot be unified. 1. At the beginning of the project, the team should agree on a system architecture to ensure that all systems and software adhere to this architecture. This helps reduce the problem of code inconsistency due to architectural differences. 2. Work requirements should be clearly defined and followed, and detailed coding specifications should be formulated, including naming specifications, code formats, comment styles, etc., to ensure that all members of the group use the same specifications for code operations. Improve code readability and maintainability. 3. Ensure that environment variables and configuration files for different systems and software are properly managed. Use configuration files or environment variables to store and pass system-specific configuration information to reduce code inconsistencies due to environment differences.

(4) How to optimize the arrangement in a limited time

Team work requires a reasonable arrangement of time, and it is necessary to ensure that the task is completed on time within a limited time. However, in actual work, improper time management and uneven distribution will affect the progress. In group work: 1. Define goals and tasks, make a time schedule, evaluate the time and resources required for each task, and ensure that the arrangement is practical and feasible. 2. Set priorities and tackle problems step by step. 3. The group leader urged and checked in time, and the team members responded actively to avoid ineffective operations. 4. Stay flexible and adjust your plans to new situations in a timely and timely manner.At the same time, I would like to thank the offline teachers for their patient tutoring and the online teachers for their careful teaching.

1. Screenshot and explanation on unresolved/unachievable requirements.

None.