

2021/12/10 上午2:29	Address.java	AssociationDemo.java
<pre> 1 package ex4; 2 3 /** 4 * 5 * @author vanting 6 */ 7 public class AssociationDemo { 8 9 public static void main(String[] args) { 10 11 Student peter = new Student("Peter"); 12 Student mary = new Student("Mary"); 13 Student john = new Student("John"); 14 15 Course chinese = new Course("Chinese"); 16 Course english = new Course("English"); 17 Course maths = new Course("Maths"); 18 19 enroll(maths, peter); 20 enroll(maths, john); 21 enroll(chinese, peter); 22 enroll(chinese, mary); 23 enroll(english, mary); 24 enroll(english, john); 25 26 chinese.showList(); 27 english.showList(); 28 maths.showList(); 29 john.showList(); 30 mary.showList(); 31 peter.showList(); 32 } 33 34 static void enroll(Course course, Student student) { 35 // associate objects here 36 course.addStudent(student); 37 student.addCourse(course); 38 } 39 } 40 41 class Student { 42 private static final int MAX = 3; 43 private int count = 0; 44 private Course[] courseList; 45 private String name; 46 47 Student(String name) { 48 this.name = name; 49 courseList = new Course[MAX]; 50 } 51 52 public void addCourse(Course course) { 53 if(count < MAX) 54 courseList[count++] = course; 55 } 56 57 @Override 58 public String toString() { 59 return name; </pre>	<pre> 1 package ex4; 2 3 public final class Address { 4 5 private String street; 6 private String city; 7 private String state; 8 private String zip; 9 10 /** Create an address with street, city, state, and zip */ 11 public Address(String street, String city, String state, String zip) { 12 this.street = street; 13 this.city = city; 14 this.state = state; 15 this.zip = zip; 16 } 17 18 /** Return street */ 19 public String getStreet() { 20 return street; 21 } 22 23 /** Return city */ 24 public String getCity() { 25 return city; 26 } 27 28 /** Return state */ 29 public String getState() { 30 return state; 31 } 32 33 /** Return zip */ 34 public String getZip() { 35 return zip; 36 } 37 38 /** Get full address */ 39 public String getFullAddress() { 40 return street + '\n' + city + ", " + state + ' ' + zip + '\n'; 41 } 42 } 43 </pre>	<pre> 1 package ex4; 2 3 /** 4 * 5 * @author vanting 6 */ 7 public class AssociationDemo { 8 9 public static void main(String[] args) { 10 11 Student peter = new Student("Peter"); 12 Student mary = new Student("Mary"); 13 Student john = new Student("John"); 14 15 Course chinese = new Course("Chinese"); 16 Course english = new Course("English"); 17 Course maths = new Course("Maths"); 18 19 enroll(maths, peter); 20 enroll(maths, john); 21 enroll(chinese, peter); 22 enroll(chinese, mary); 23 enroll(english, mary); 24 enroll(english, john); 25 26 chinese.showList(); 27 english.showList(); 28 maths.showList(); 29 john.showList(); 30 mary.showList(); 31 peter.showList(); 32 } 33 34 static void enroll(Course course, Student student) { 35 // associate objects here 36 course.addStudent(student); 37 student.addCourse(course); 38 } 39 } 40 41 class Student { 42 private static final int MAX = 3; 43 private int count = 0; 44 private Course[] courseList; 45 private String name; 46 47 Student(String name) { 48 this.name = name; 49 courseList = new Course[MAX]; 50 } 51 52 public void addCourse(Course course) { 53 if(count < MAX) 54 courseList[count++] = course; 55 } 56 57 @Override 58 public String toString() { 59 return name; </pre>
2021/12/10 上午2:29	localhost:4649/?mode=click	localhost:4649/?mode=click
1/1		1/2

```
60 }
61
62 public void showList() {
63     System.out.println(name + ": ");
64     for(int i=0; i<count; i++)
65         System.out.print(courseList[i] + " / ");
66     System.out.println("");
67 }
68 }
69
70 class Course {
71     private static final int MAX = 10;
72     private int count = 0;
73     private Student[] studentList;
74     private String name;
75
76     Course(String name) {
77         this.name = name;
78         studentList = new Student[MAX];
79     }
80
81     public void addStudent(Student student) {
82         if(count < MAX)
83             studentList[count++] = student;
84     }
85
86     @Override
87     public String toString() {
88         return name;
89     }
90
91     public void showList() {
92         System.out.println(name + ": ");
93         for(int i=0; i<count; i++)
94             System.out.print(studentList[i] + " / ");
95         System.out.println("");
96     }
97 }
```

```
1 package ex4;
2
3 public class Borrower extends Person {
4
5     private Loan loan;
6
7     /** Construct a borrower with default properties */
8     public Borrower() {
9         super();
10    }
11
12    /** Create a borrower with specified name and address */
13    public Borrower(Name name, Address address) {
14        super(name, address);
15    }
16
17    /** Return loan */
18    public Loan getLoan() {
19        return loan;
20    }
21
22    /** Set a new loan */
23    public void setLoan(Loan loan) {
24        this.loan = loan;
25    }
26
27    /** String representation for borrower */
28    public String toString() {
29        return super.toString() +
30            "Monthly payment is " + loan.getMonthlyPayment() + '\n' +
31            "Total payment is " + loan.getTotalPayment();
32    }
33 }
34
```

```
1 package ex4;
2
3 import javax.swing.JOptionPane;
4
5 public class BorrowLoan {
6
7     /** Main method */
8     public static void main(String[] args) {
9         // Create a name
10        Name name = new Name("John", 'D', "Smith");
11
12        // Create an address
13        Address address = new Address("100 Main Street", "Savannah",
14        "GA", "31419");
15
16        // Create a loan
17        Loan loan = new Loan(5.5, 15, 250000);
18
19        // Create a borrower
20        Borrower borrower = new Borrower(name, address);
21
22        borrower.setLoan(loan);
23
24        // Display loan information
25        JOptionPane.showMessageDialog(null, borrower.toString());
26    }
27 }
28
```

```
1 package ex4;
2
3 /**
4  *
5  * @author wanting
6  */
7 public class Calculator {
8
9     private FactorialInterface factorialPlugin;
10
11     public void addFactorialPlugin(FactorialInterface f) {
12         factorialPlugin = f;
13     }
14
15     public int findFactorial(int n) {
16         return factorialPlugin.factorial(n);
17     }
18
19 }
20
```

```
1 package ex4;
2
3 /**
4  *
5  * @author vanting
6  */
7 public interface Colorable {
8
9     abstract public void color();
10
11 }
12
```

```
1 package ex4;
2
3 /**
4  *
5  * @author vanting
6  */
7 public class DependenceManager {
8
9     public static void main(String[] args) {
10         Calculator cal = new Calculator();
11         cal.addFactorialPlugin(new FactorialImpl1());
12         int n = cal.findFactorial(3);
13         System.out.println(n);
14     }
15 }
16
```

```
1  / *
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package ex4;
7
8 /**
9  * @author wanting
10 */
11
12 public interface Edible {
13
14     public abstract void eat();
15 }
16
```

```
1 package ex4;
2
3 /**
4  *
5  * @author wanting
6  */
7 public class FactorialImpl1 implements FactorialInterface {
8
9
10     public int factorial(int n) {
11         if(n == 0)
12             return 1;
13         else
14             return n * factorial(n-1);
15     }
16 }
17
```

```
1 package ex4;
2 /**
3  *
4  * @author vanting
5  */
6 public class FactorialImpl2 implements FactorialInterface {
7
8     public int factorial(int n) {
9         if (n == 0) {
10             return 1;
11         } else {
12             int product = 1;
13             for (int i = 2; i <= n; i++) {
14                 product *= i;
15             }
16             return product;
17         }
18     }
19 }
20
21 }
```

```
1 package ex4;
2 /**
3  *
4  * @author vanting
5  */
6 public interface FactorialInterface {
7
8     public abstract int factorial(int n);
9
10 }
11
12 }
```

```
1 package ex4;
2
3 public class InterfaceDemo {
4
5     public static void main(String[] args) {
6         Colorable[] fruits = {new Apple(), new Banana()};
7         findEdibleFruitColor(fruits);
8     }
9
10    public static void findEdibleFruitColor(Colorable[] fruits) {
11        for(int i=0; i<fruits.length; i++)
12            if(fruits[i] instanceof Edible)
13                fruits[i].color();
14    }
15 }
16
17 class Apple implements Colorable, Edible {
18
19     @Override
20     public void color() {
21         System.out.println("Red");
22     }
23
24     @Override
25     public void eat() {
26         System.out.println("I'm juicy");
27     }
28 }
29
30
31 class Banana implements Colorable {
32
33     @Override
34     public void color() {
35         System.out.println("Yellow");
36     }
37 }
38
```

```
1 package ex4;
2
3 public class Loan {
4
5     private double annualInterestRate;
6     private int numberOfYears;
7     private double loanAmount;
8     private java.util.Date loanDate;
9
10    /** Default constructor */
11    public Loan() {
12        this(7.5, 30, 100000);
13    }
14
15    public Loan(double annualInterestRate, int numberOfYears, double loanAmount) {
16        this.annualInterestRate = annualInterestRate;
17        this.numberOfYears = numberOfYears;
18        this.loanAmount = loanAmount;
19        loanDate = new java.util.Date();
20    }
21
22    /** Return annualInterestRate */
23    public double getAnnualInterestRate() {
24        return annualInterestRate;
25    }
26
27    /** Set a new annualInterestRate */
28    public void setAnnualInterestRate(double annualInterestRate) {
29        this.annualInterestRate = annualInterestRate;
30    }
31
32    /** Return numberOfYears */
33    public int getNumberOfYears() {
34        return numberOfYears;
35    }
36
37    /** Set a new numberOfYears */
38    public void setNumberOfYears(int numberOfYears) {
39        this.numberOfYears = numberOfYears;
40    }
41
42    /** Return loanAmount */
43    public double getLoanAmount() {
44        return loanAmount;
45    }
46
47    /** Set a newLoanAmount */
48    public void setLoanAmount(double loanAmount) {
49        this.loanAmount = loanAmount;
50    }
51
52    /** Find monthly payment */
53    public double getMonthlyPayment() {
54        double monthlyInterestRate = annualInterestRate / 1200;
55        return loanAmount * monthlyInterestRate / (1 - (Math.pow(1 / (1 + monthlyInterestRate), numberOfYears * 12)));
56    }
57
58    /** Find total payment */
59
```

2021/12/10 上午2:30

Loan.java

```
59 public double getTotalPayment() {
60     return getMonthlyPayment() * numberOfYears * 12;
61 }
62
63 /** Return loan date */
64 public java.util.Date getLoanDate() {
65     return loanDate;
66 }
67 }
68
```

2021/12/10 上午2:30

Name.java

```
1 package ex4;
2
3 public final class Name implements Comparable {
4
5     private String firstName;
6     private char mi;
7     private String lastName;
8
9     /** Construct a name with firstName, mi, and lastName */
10    public Name(String firstName, char mi, String lastName) {
11        this.firstName = firstName;
12        this.mi = mi;
13        this.lastName = lastName;
14    }
15
16    /** Return firstName */
17    public String getFirstName() {
18        return firstName;
19    }
20
21    /** Return middle name initial */
22    public char getMi() {
23        return mi;
24    }
25
26    /** Return lastName */
27    public String getLastName() {
28        return lastName;
29    }
30
31    /** Obtain full name */
32    public String getFullName() {
33        return firstName + ' ' + mi + ' ' + lastName;
34    }
35
36    /** Implement compareTo in the Comparable interface */
37    public int compareTo(Object o) {
38        if (!lastName.equals(((Name) o).lastName)) {
39            return lastName.compareTo(((Name) o).lastName);
40        } else if (!firstName.equals(((Name) o).firstName)) {
41            return firstName.compareTo(((Name) o).firstName);
42        } else {
43            return mi - ((Name) o).mi;
44        }
45    }
46 }
47
```



```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package ex4;
6
7 /**
8  *
9  * @author wanting
10 */
11 public class Outer {
12
13     public static void main(String[] args) {
14         Outer o = new Outer();
15         Inner i = o.new Inner();
16     }
17
18     // non-static nested class
19     public class Inner {
20
21     }
22 }
23
```

```
1 package ex4;
2
3 public class Person implements Comparable {
4
5     private Name name;
6     private Address address;
7
8     /** Construct a person with default properties */
9     public Person() {
10         this(new Name("Jill", 'S', "Barr"),
11              new Address("100 Main", "Savannah", "GA", "31411"));
12     }
13
14     /** Construct a person with specified name and address */
15     public Person(Name name, Address address) {
16         this.name = name;
17         this.address = address;
18     }
19
20     /** Return name */
21     public Name getName() {
22         return name;
23     }
24
25     /** Set a new name */
26     public void setName(Name name) {
27         this.name = name;
28     }
29
30     /** Return address */
31     public Address getAddress() {
32         return address;
33     }
34
35     /** Set a new address */
36     public void setAddress(Address address) {
37         this.address = address;
38     }
39
40     /** Override the toString method */
41     public String toString() {
42         return '\n' + name.getFullName() + '\n' +
43             address.getFullAddress() + '\n';
44     }
45
46     /** Implement compareTo in the Comparable interface */
47     public int compareTo(Object o) {
48         return name.compareTo(((Person) o).name);
49     }
50 }
51
```