语法

面向对象程序设计

第 3 讲 Java 语法语法 -2

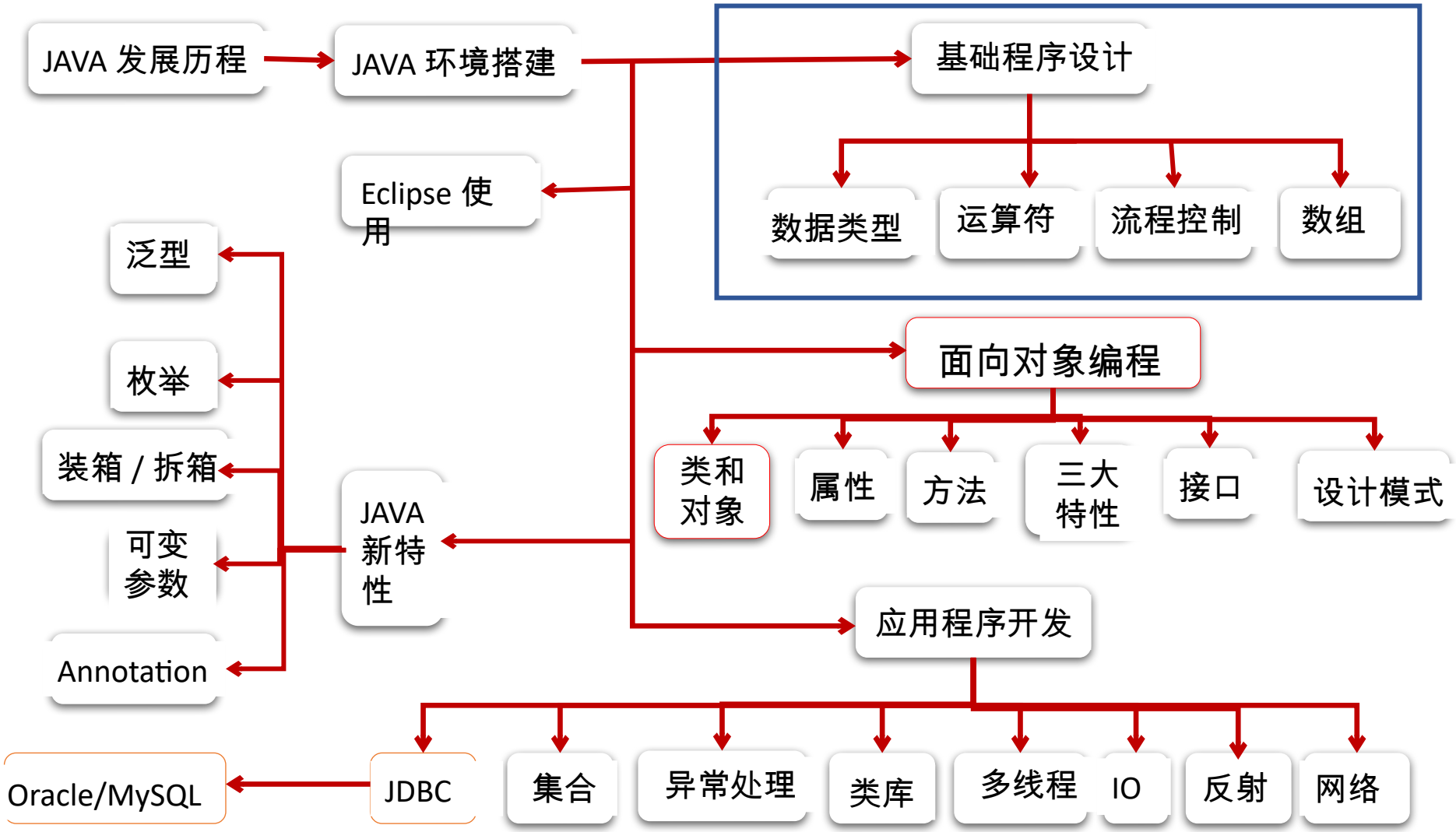
刘进

2230652597@qq.com

OOP 教辅 2025 秋季 QQ 群：

Java 基础知识图解

主要知识点



第 3 讲 Java 语法基础 -2

3.1 流程控制（选择、循环）

控制
程序设计方法学

3.2 数组

3.3 示例学习

控 制 语 句

•控制语句用于改变程序执行的顺序。程序利用控制语句有条件地执行语句、循环地执行语句或者跳转到程序中的其他部分执行语句。

•当编写程序的时候，如果没有使用控制语句，计算机将顺序执行所有的语句。如果要改变程序的流程，可以在程序中使用控制语句来有条件地选择执行语句或重复执行某个语句块。

•Java 的控制语句有：

· if-else 语句
· switch 语句
· while 和 do-while 语句
· for 语句
· 跳转语句
· 异常处理语句

程序流程控制

● 顺序结构

➢ 程序从上到下逐行地执行，中间没有任何判断和跳转。

● 分支结构

➢ 根据条件，选择性地执行某段代码。

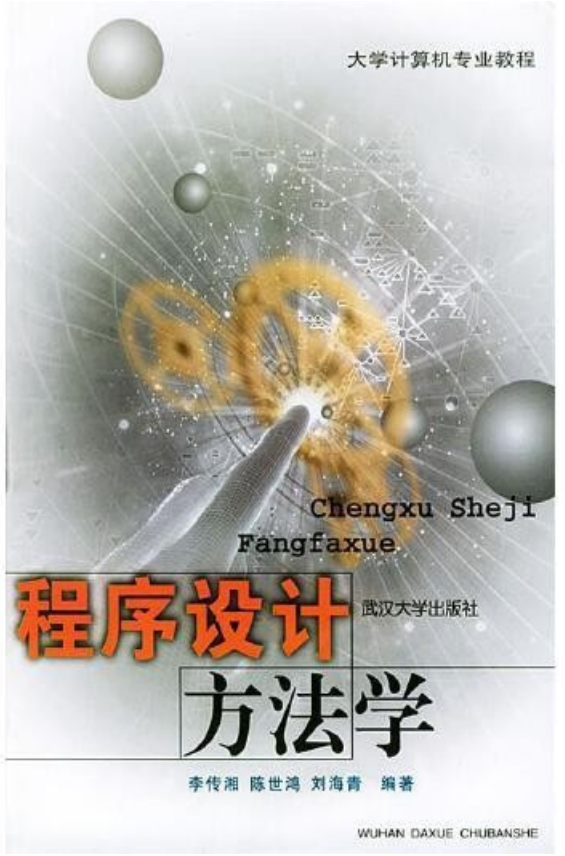
➢ 有 if…else 和 switch 两种分支语句。

● 循环结构

➢ 根据循环条件，重复性的执行某段代码。

➢ 有 while 、 do…while 、 for 三种循环语句。

➢ 注： JDK1.5 提供了 foreach 循环，方便的遍历集合、数组元素。



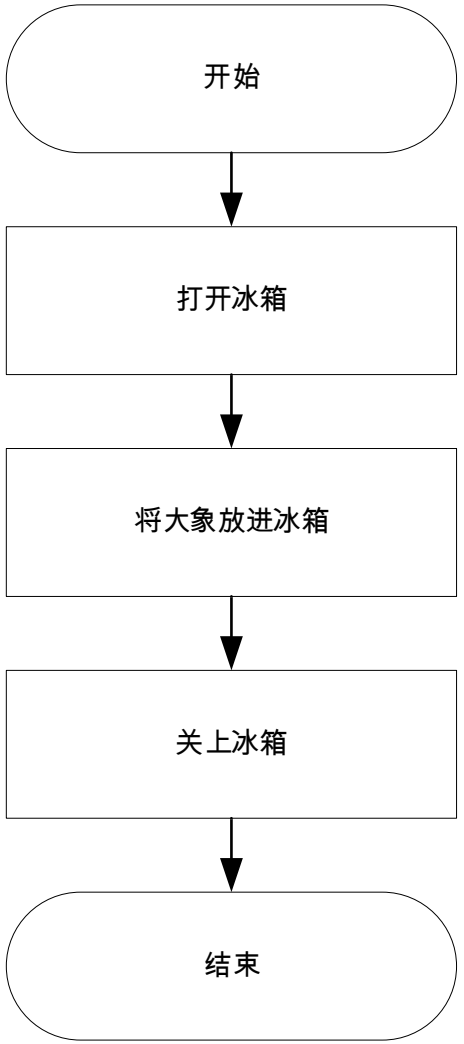
控制
程序设计方法学

程序流程控制

●顺序结构

把大象装冰箱里一共需要三步：
第一步是打开冰箱，
第二步是把大象放进冰箱里面，
第三步是把冰箱门关上。

以上步骤必须依次运行，而且每个步骤只需运行一次。



顺序结构

程序流程控制

●顺序结构

Java 中定义成员变量时采用合法的前向引用。如：

先定义再使用

```
public class Test{
        int num1 = 12;
        int num2 = num1 + 2;
}
```

错误形式：

```
public class Test{
       int num2 = num1 + 2  ;
       int num1 = 12;
}
```

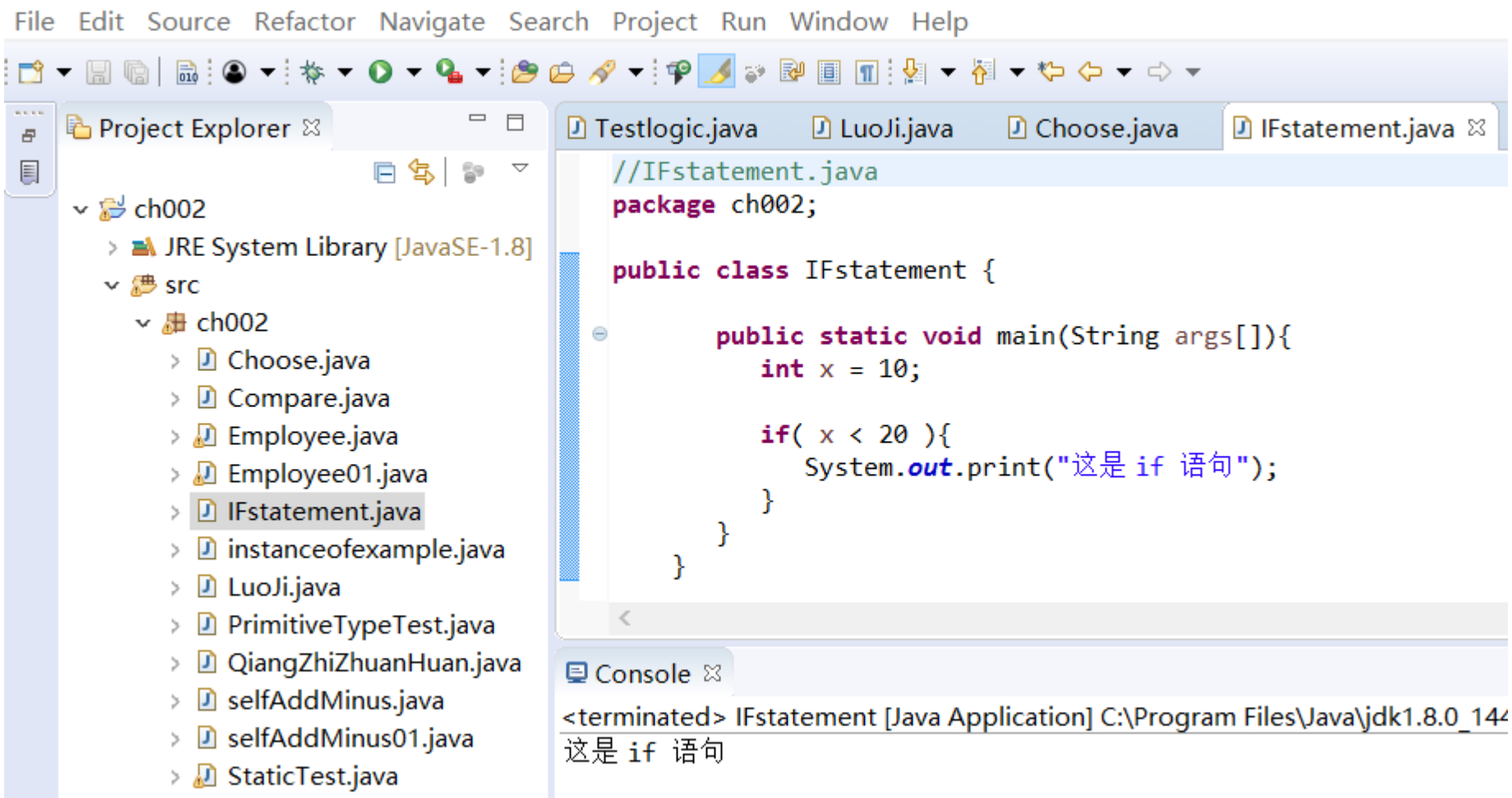
分支语句 1 ： if-else 语句

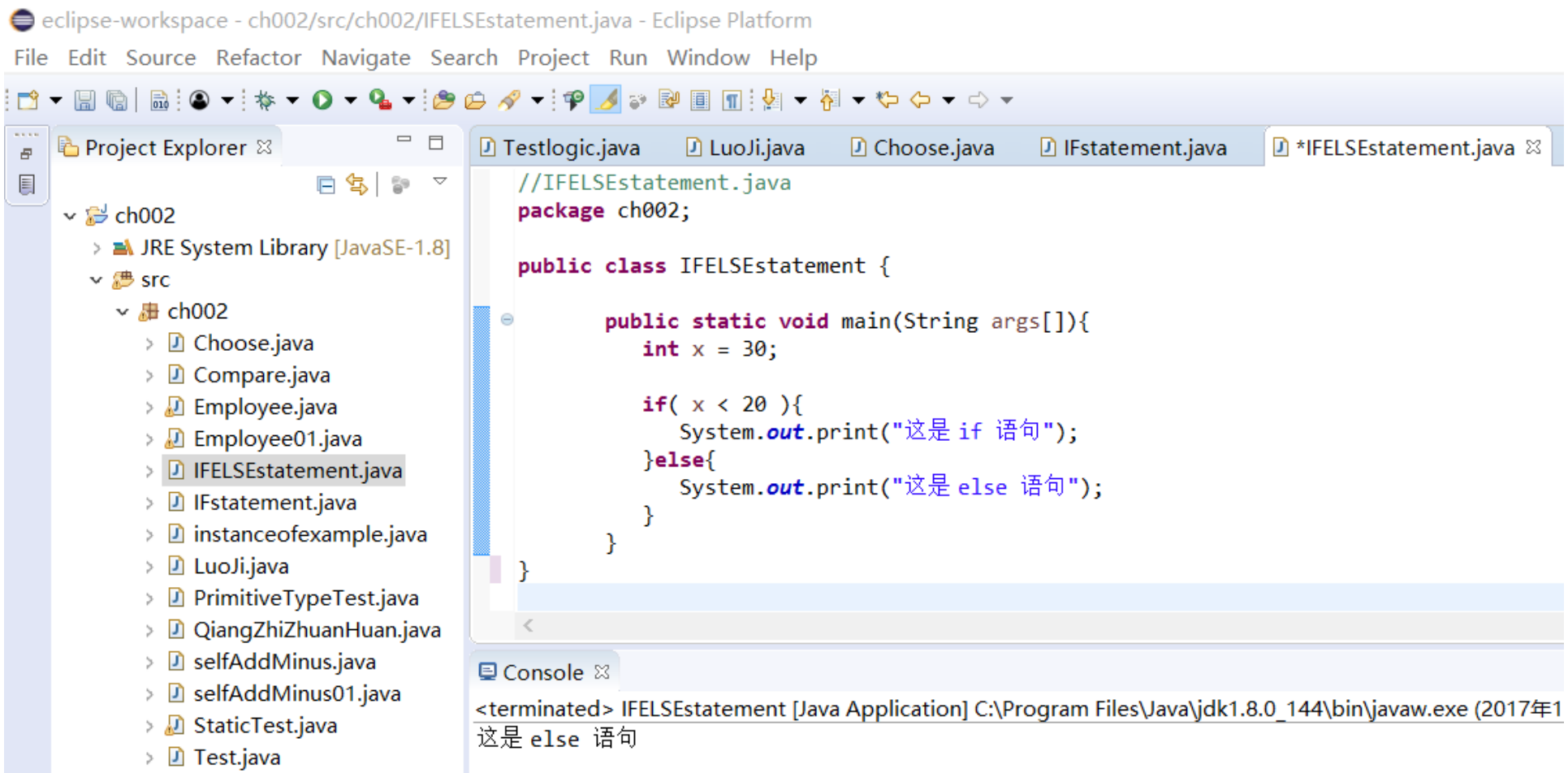
if 语句三种格式：

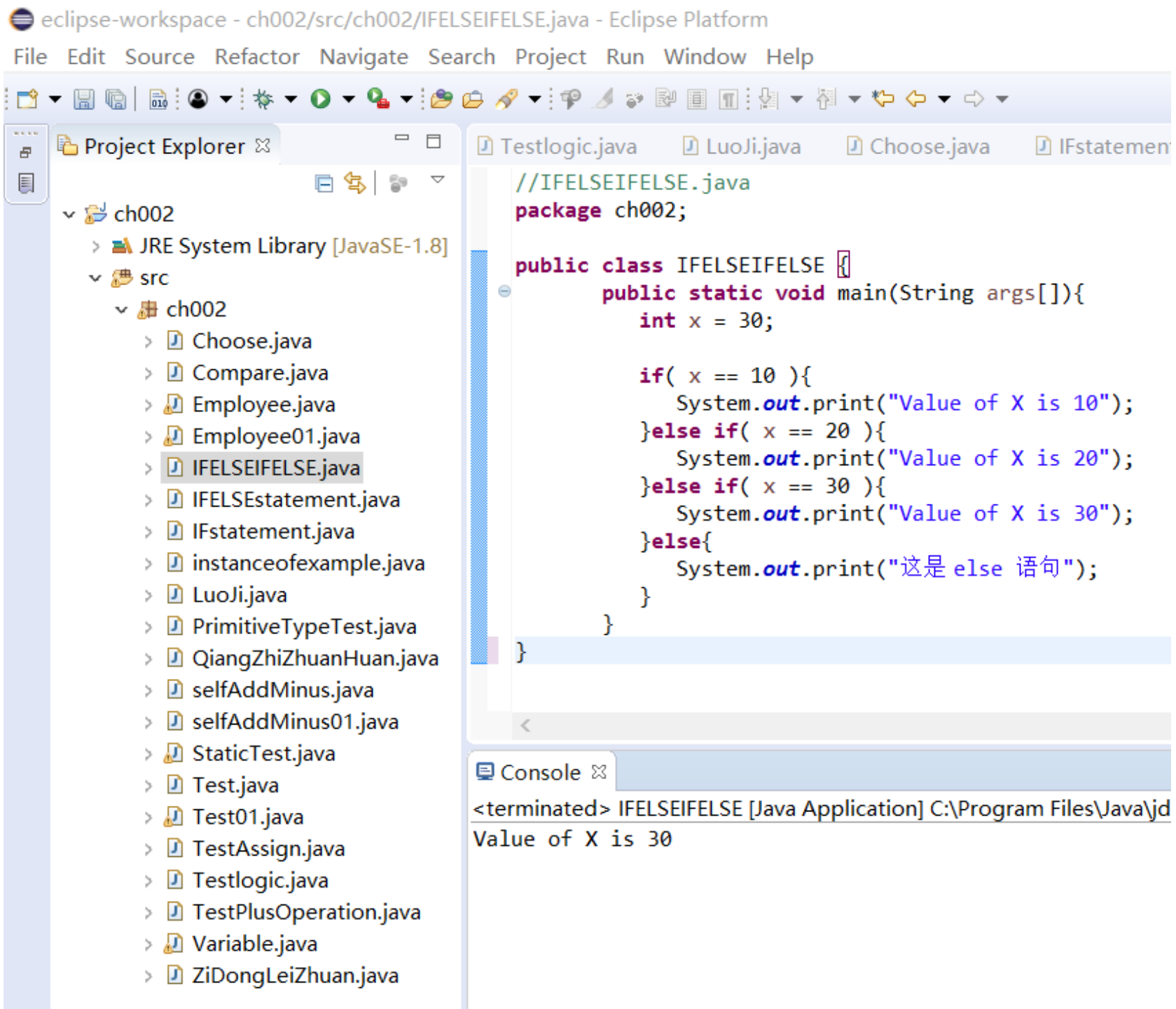
1. if(条件表达式){
        执行代码块 A ；
    }

2. if(条件表达式){
        执行代码块 A ；
    }
    else{
        执行代码块 B ；
    }

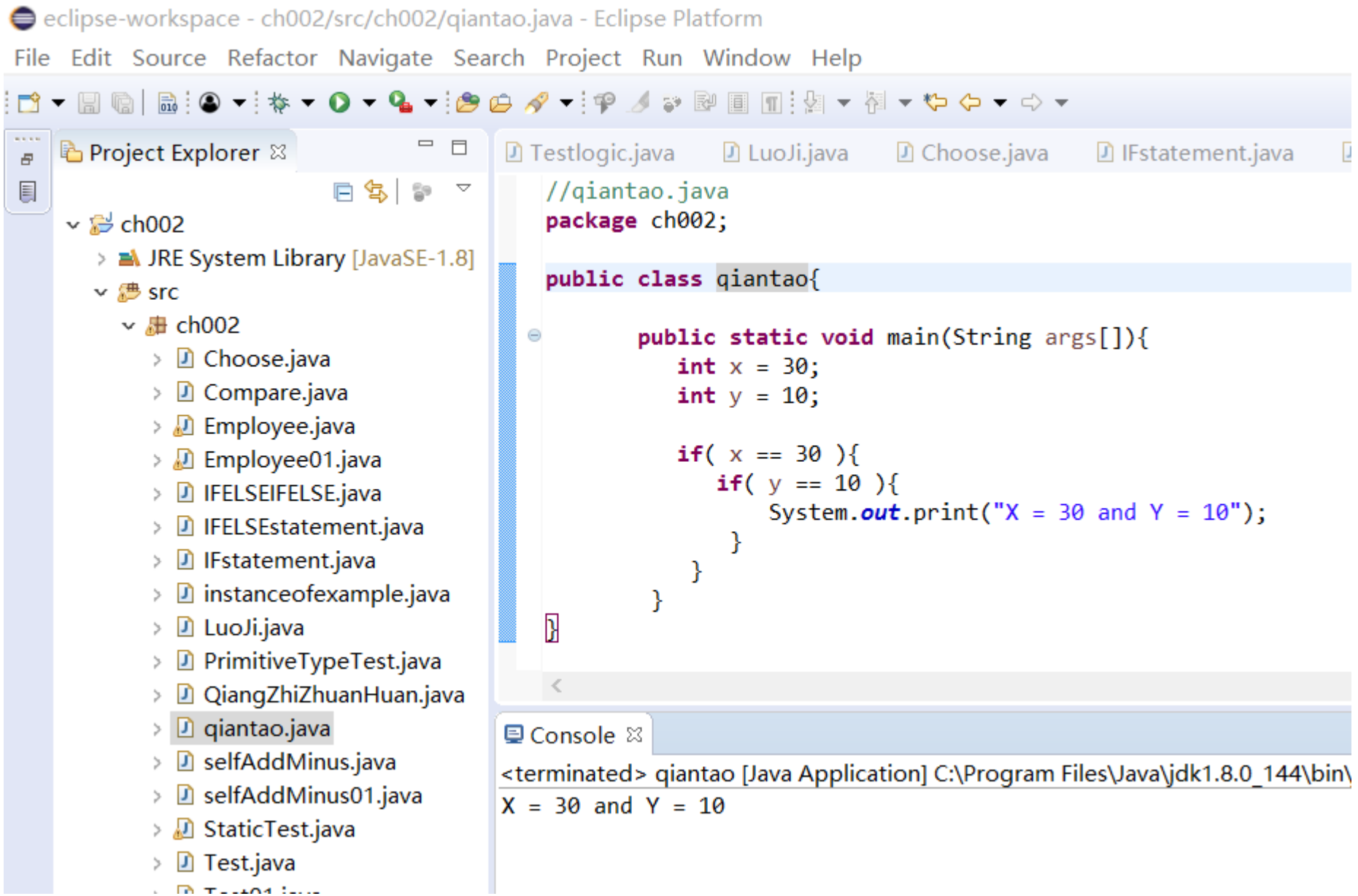
3. if(条件表达式 1){
        执行代码块 A ；
    }
    else if (条件表达式 2)
{
        执行代码块 B ；
    }
……
    else if (条件表达式 n)
{
        执行代码块 N ；
    }
    else{
        执行代码块 N+1 ；
    }


File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ch002
JRE System Library [JavaSE-1.8]
src
ch002
Choose.java
Compare.java
Employee.java
Employee01.java
IFstatement.java
instanceofexample.java
LuoJi.java
PrimitiveTypeTest.java
QiangZhiZhuanHuan.java
selfAddMinus.java
selfAddMinus01.java
StaticTest.java
Testlogic.java
LuoJi.java
Choose.java
IFstatement.java
//IFstatement.java
package ch002;
public class IFstatement {
public static void main(String args[]){
int x = 10;
if(x < 20){
System.out.print("这是 if 语句");
}
}
}
Console
<terminated> IFstatement [Java Application] C:\Program Files\Java\jdk1.8.0_144
这是 if 语句


eclipse-workspace - ch002/src/ch002/IFELSEstatement.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ch002
JRE System Library [JavaSE-1.8]
src
ch002
Choose.java
Compare.java
Employee.java
Employee01.java
IFELSEstatement.java
IFstatement.java
instanceofexample.java
LuoJi.java
PrimitiveTypeTest.java
QiangZhiZhuanHuan.java
selfAddMinus.java
selfAddMinus01.java
StaticTest.java
Test.java
Testlogic.java
LuoJi.java
Choose.java
IFstatement.java
*IFELSEstatement.java
//IFELSEstatement.java
package ch002;
public class IFELSEstatement {
public static void main(String args[]){
int x = 30;
if(x < 20){
System.out.print("这是 if 语句");
}else{
System.out.print("这是 else 语句");
}
}
}
Console
<terminated> IFELSEstatement [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (2017年1
这是 else 语句


eclipse-workspace - ch002/src/ch002/IFELSEIFELSE.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ch002
JRE System Library [JavaSE-1.8]
src
ch002
Choose.java
Compare.java
Employee.java
Employee01.java
IFELSEIFELSE.java
IFELSEstatement.java
IFstatement.java
instanceofexample.java
LuoJi.java
PrimitiveTypeTest.java
QiangZhiZhuanHuan.java
selfAddMinus.java
selfAddMinus01.java
StaticTest.java
Test.java
Test01.java
TestAssign.java
Testlogic.java
TestPlusOperation.java
Variable.java
ZiDongLeiZhuan.java
Testlogic.java
LuoJi.java
Choose.java
IFstatemen
//IFELSEIFELSE.java
package ch002;
public class IFELSEIFELSE {
public static void main(String args[]){
int x = 30;
if(x == 10){
System.out.print("Value of X is 10");
}else if(x == 20){
System.out.print("Value of X is 20");
}else if(x == 30){
System.out.print("Value of X is 30");
}else{
System.out.print("这是 else 语句");
}
}
}
Console
<terminated> IFELSEIFELSE [Java Application] C:\Program Files\Java\jd
Value of X is 30

if else if

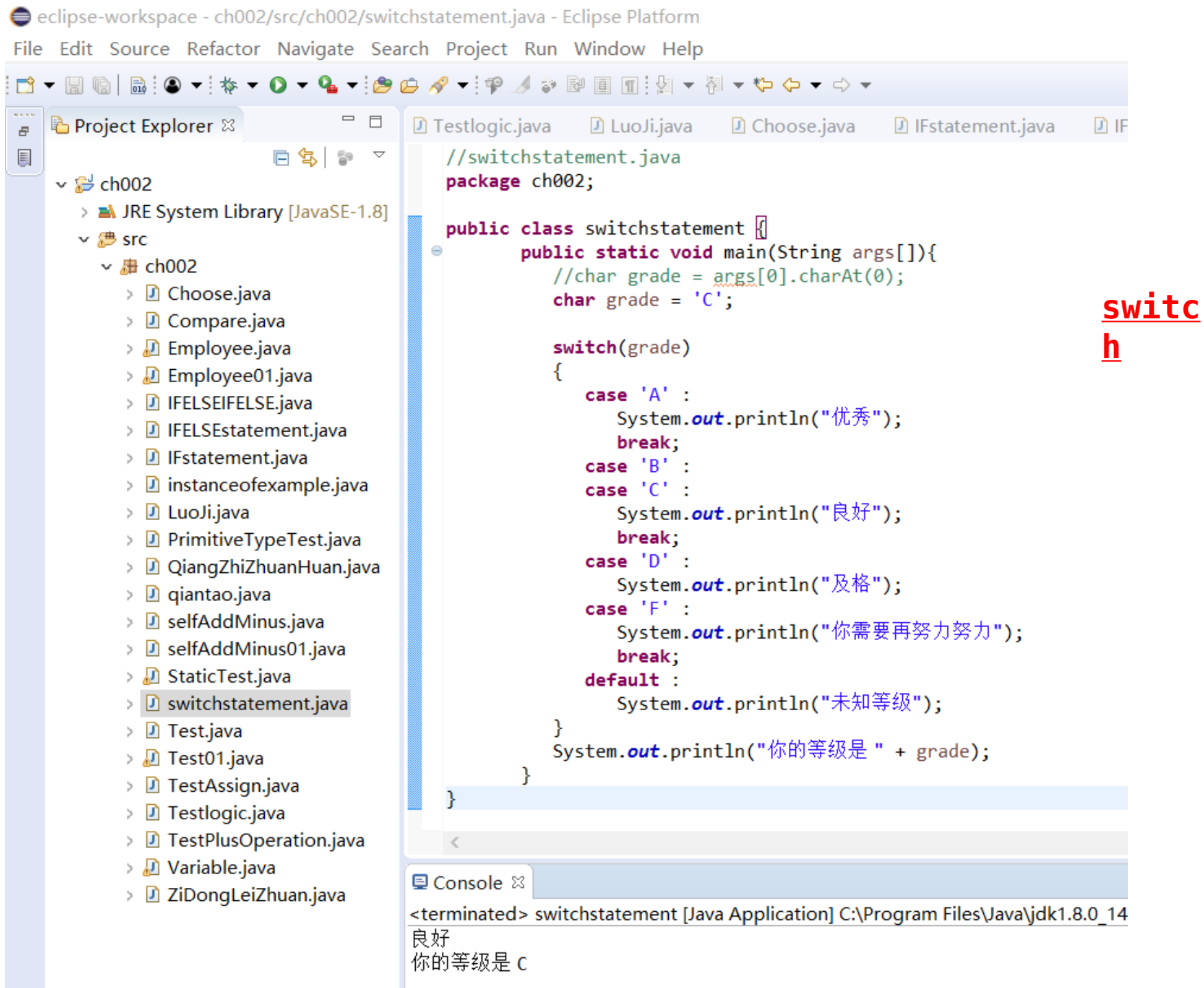

eclipse-workspace - ch002/src/ch002/qiantao.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ch002
JRE System Library [JavaSE-1.8]
src
ch002
Choose.java
Compare.java
Employee.java
Employee01.java
IFELSEIFELSE.java
IFELSEstatement.java
IFstatement.java
instanceofexample.java
LuoJi.java
PrimitiveTypeTest.java
QiangZhiZhuanHuan.java
qiantao.java
selfAddMinus.java
selfAddMinus01.java
StaticTest.java
Test.java
Testlogic.java
LuoJi.java
Choose.java
IFstatement.java
//qiantao.java
package ch002;
public class qiantao{
public static void main(String args[]){
int x = 30;
int y = 10;
if(x == 30){
if(y == 10){
System.out.print("X = 30 and Y = 10");
}
}
}
}
Console
<terminated> qiantao [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\
X = 30 and Y = 10

分支结构 2 ： switch 语句

```
switch( 变量 ){
  case  常量 1:
        语句 1;
        break;
  case  常量 2:
        语句 2;
        break;
  … …
  case  常量 N:
        语句 N;
        break;
  default:
        语句 ;

 }
```

switc
h


eclipse-workspace - ch002/src/ch002/switchstatement.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ch002
JRE System Library [JavaSE-1.8]
src
ch002
Choose.java
Compare.java
Employee.java
Employee01.java
IFELSEIFELSE.java
IFELSEstatement.java
IFstatement.java
instanceofexample.java
LuoJi.java
PrimitiveTypeTest.java
QiangZhiZhuanHuan.java
qiantao.java
selfAddMinus.java
selfAddMinus01.java
StaticTest.java
switchstatement.java
Test.java
Test01.java
TestAssign.java
Testlogic.java
TestPlusOperation.java
Variable.java
ZiDongLeiZhuan.java
Testlogic.java
LuoJi.java
Choose.java
IFstatement.java
//switchstatement.java
package ch002;
public class switchstatement {
public static void main(String args[]){
//char grade = args[0].charAt(0);
char grade = 'C';
switch(grade)
{
case 'A' :
System.out.println("优秀");
break;
case 'B' :
case 'C' :
System.out.println("良好");
break;
case 'D' :
System.out.println("及格");
case 'F' :
System.out.println("你需要再努力努力");
break;
default :
System.out.println("未知等级");
}
System.out.println("你的等级是 " + grade);
}
}
Console
<terminated> switchstatement [Java Application] C:\Program Files\Java\jdk1.8.0_14
良好
你的等级是 C

switch

switch 语句应用举例

```
public class Test{
    public static void main(String args[]){
                String season = “summer”;
                switch (season) {
                case “spring”:
                        System.out.println( “春暖花开 ");
                        break;
                case “summer”:
                        System.out.println( “夏日炎炎 ");
                        break;
                case “autumn”:
                        System.out.println( “秋高气爽 ");
                        break;
                case “winter”:
                        System.out.println( “冬雪皑皑 ");
                        break;

                default:
                        System.out.println( “季节输入有误 ");
                        break;
                }}}
```

switch

switch 语句有关规则

- switch(表达式) 中表达式的**返回值**必须是下述几种类型之一： **byte ， short ， char ， int ，枚举， String** ；
- case 子句中的值必须是**常量**，且所有 case 子句中的值应是不同的；
- default 子句是**可任选的**，当没有匹配的 case 时，执行 default
- break 语句用来在执行完一个 case 分支后使程序跳出 switch 语句块；如果没有 break ，程序会顺序执行到 switch 结尾

switch

switch 语句练习 1

- 使用 switch 语句改写下列 if 语句：

```
int a = 3;
int x = 100;
if(a==1)
        x+=5;
else if(a==2)
        x+=10;
else if(a==3)
        x+=16;
else
        x+=34;
```

switch 和 if 语句的对比

if 和 switch 语句很像，具体什么场景下，应用哪个语句呢？

- 如果判断的具体数值不多，而且符合 byte 、 short 、 int 、 char 这四种类型。虽然两个语句都可以使用，建议使用 swtich 语句。因为效率稍高。

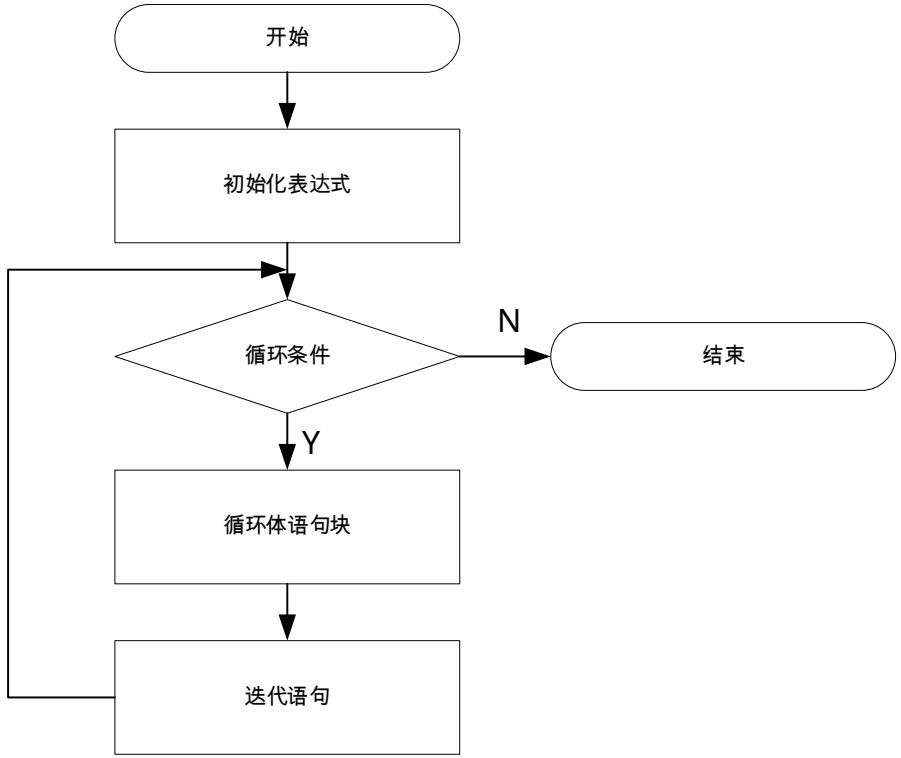
离散量，连续量

- 其他情况：对区间判断，对结果为 boolean 类型判断，使用 if ， if 的使用范围更广。

循环结构

循环

- **循环语句功能**
  - 在某些条件满足的情况下，反复执行特定代码的功能
- **循环语句的四个组成部分**
  - 初始化部分（ init_statement ）
  - 循环条件部分（ test_exp ）
  - 循环体部分（ body_statement ）
  - 迭代部分（ alter_statement ）
- **循环语句分类**
  - for 循环
  - while 循环
  - do/while 循环



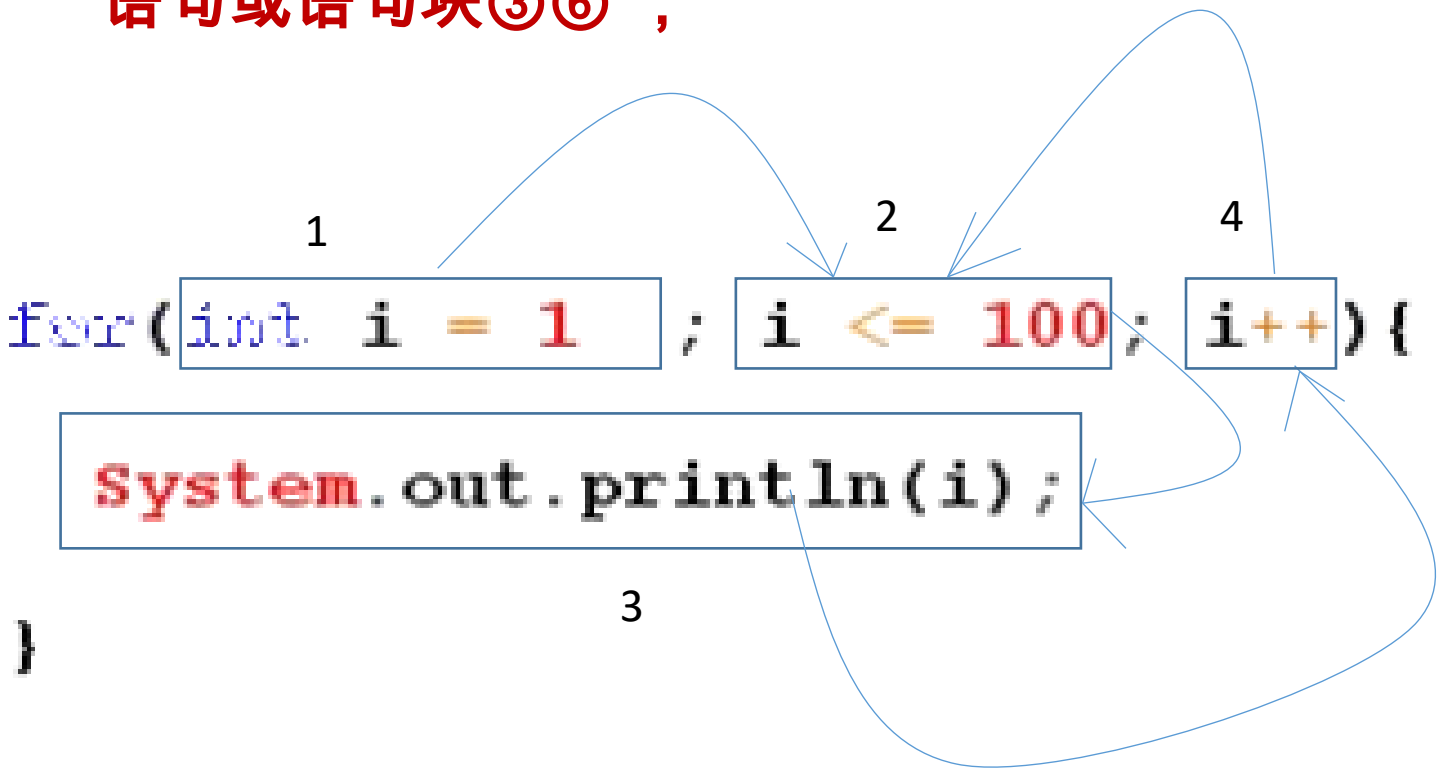
for 循环语句

●语法格式

for (初始化表达式①；布尔值测试表达式②⑤⑦；更改表达式) {

语句或语句块③⑥ ；

}



for 循环

●举例

```
public class ForLoop {
        public static void main(String args[]){
                int result = 0;
                for(int i=1; i<=100; i++) {
                   result += i;
                }
            System.out.println("result=" + result);
             }
}
```

for 循环

while 循环语句

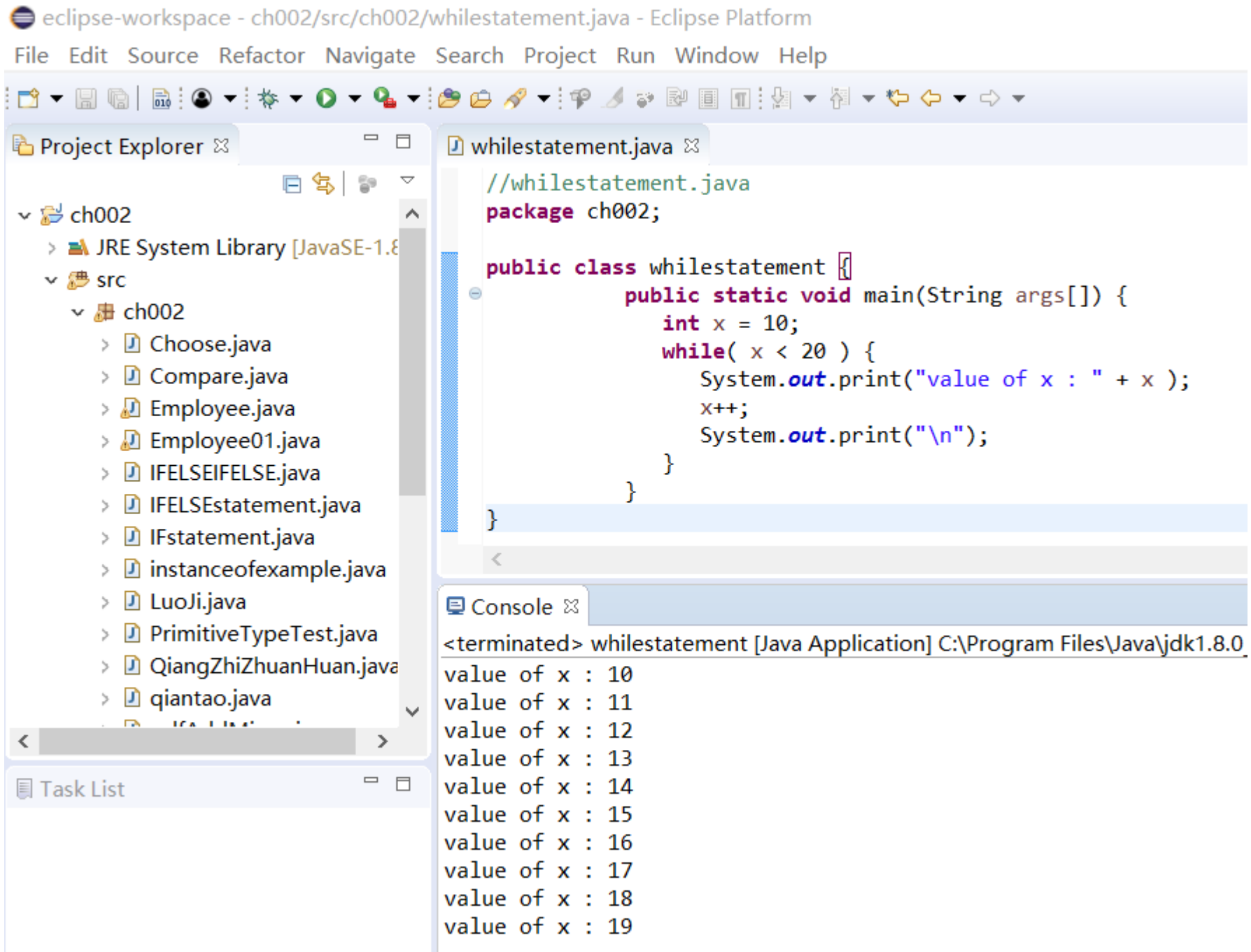
- 语法格式

```
[ 初始化语句 ]
while( 布尔值测试表达式 ) {
        语句或语句块 ;
        [ 更改语句 ;]
}
```

- 应用举例

```
public class WhileLoop {
    public static void main(String args[]){
        int result = 0;
        int i=1;
        while(i<=100) {
            result += i;
            i++;
        }
            System.out.println("result=" + result);
    }
}
```

while 循环


eclipse-workspace - ch002/src/ch002/whilestatement.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ch002
JRE System Library [JavaSE-1.8
src
ch002
Choose.java
Compare.java
Employee.java
Employee01.java
IFELSEIFELSE.java
IFELSEstatement.java
IFstatement.java
instanceofexample.java
LuoJi.java
PrimitiveTypeTest.java
QiangZhiZhuanHuan.java
qiantao.java
Task List
whilestatement.java
//whilestatement.java
package ch002;
public class whilestatement {
public static void main(String args[]) {
int x = 10;
while(x < 20) {
System.out.print("value of x : " + x);
x++;
System.out.print("\n");
}
}
}
Console
<terminated> whilestatement [Java Application] C:\Program Files\Java\jdk1.8.0
value of x : 10
value of x : 11
value of x : 12
value of x : 13
value of x : 14
value of x : 15
value of x : 16
value of x : 17
value of x : 18
value of x : 19

while 循环

do-while 循环语句

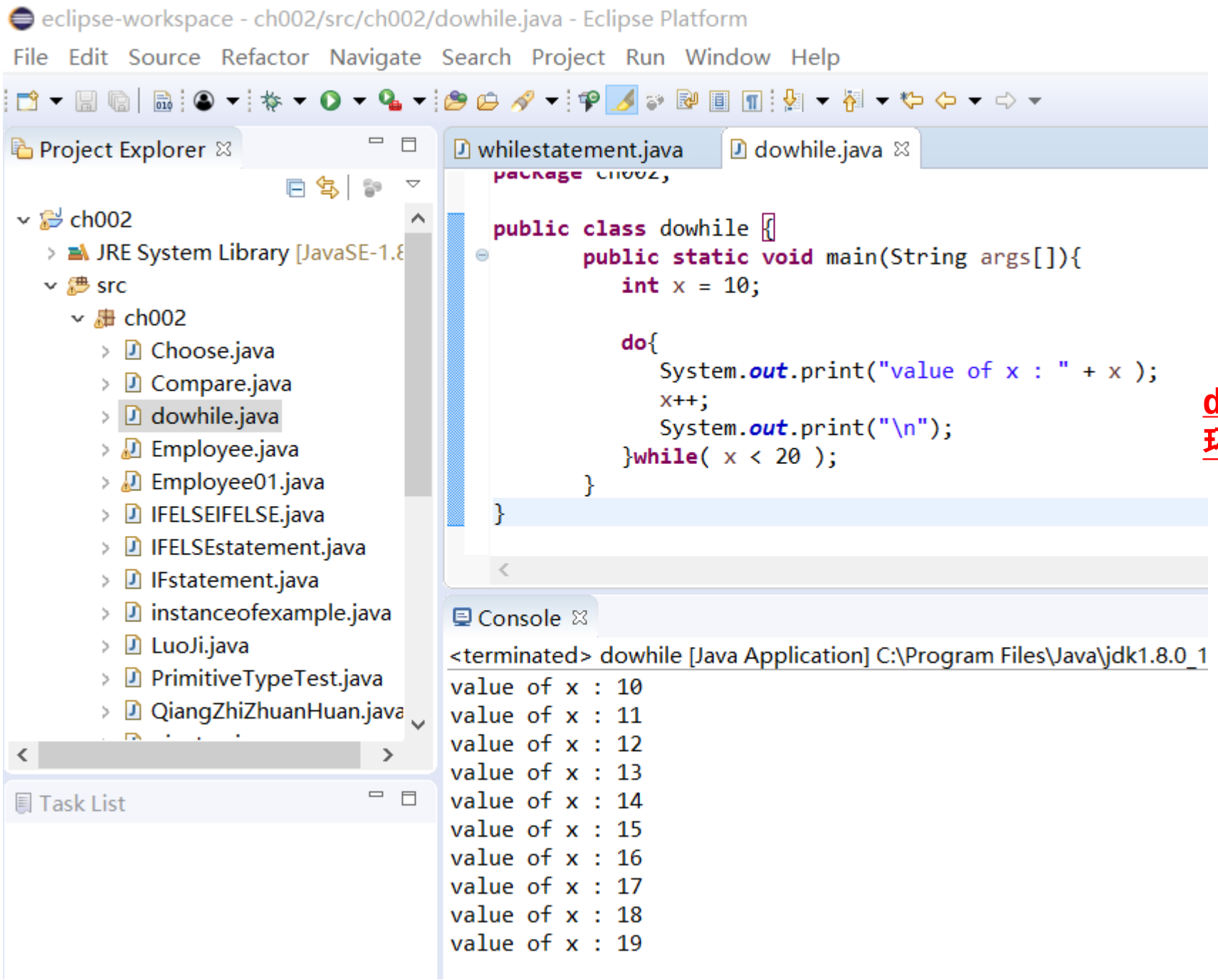
- 语法格式

```
[ 初始化语句 ]
do {
语句或语句块 ;
    [ 更改语句 ;]
} while( 布尔值测试表达式 );
```

- 应用举例

```
public class WhileLoop {
    public static void main(String args[]){
        int result = 0,  i=1;
            do{
                result += i;
                i++;
            }while(i<=100);
        System.out.println("result=" + result);
    }
}
```

do-while 循环


eclipse-workspace - ch002/src/ch002/dowhile.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ch002
JRE System Library [JavaSE-1.8
src
ch002
Choose.java
Compare.java
dowhile.java
Employee.java
Employee01.java
IFELSEIFELSE.java
IFELSEstatement.java
IFstatement.java
instanceofexample.java
LuoJi.java
PrimitiveTypeTest.java
QiangZhiZhuanHuan.java
Task List
whilestatement.java
dowhile.java
public class dowhile {
public static void main(String args[]){
int x = 10;
do{
System.out.print("value of x : " + x);
x++;
System.out.print("\n");
}while(x < 20);
}
}
Console
<terminated> dowhile [Java Application] C:\Program Files\Java\jdk1.8.0_1
value of x : 10
value of x : 11
value of x : 12
value of x : 13
value of x : 14
value of x : 15
value of x : 16
value of x : 17
value of x : 18
value of x : 19

do-while 循环

Java 增强 for 循环

Java5 引入了一种主要用于数组的增强型 for 循环。

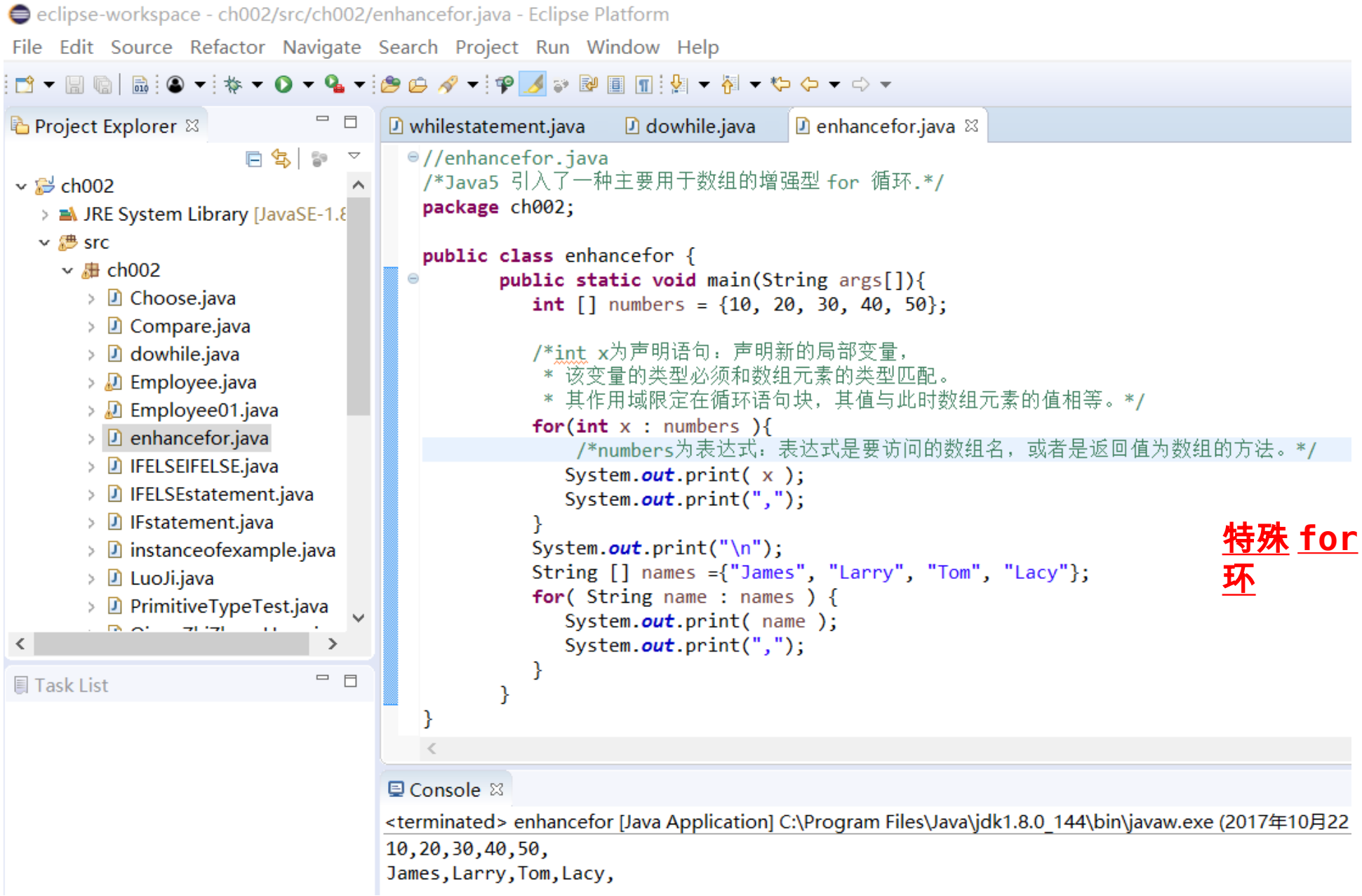
Java 增强 for 循环语法格式如下:

```
for(声明语句 : 表达式)
{
   //代码句子
}
```

声明语句：声明新的局部变量，该变量的类型必须和数组元素的类型匹配。其作用域限定在循环语句块，其值与此时数组元素的值相等。

表达式：表达式是要访问的数组名，或者是返回值为数组的方法。

特殊 for 循环


eclipse-workspace - ch002/src/ch002/enhancefor.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ch002
JRE System Library [JavaSE-1.8
src
ch002
Choose.java
Compare.java
dowhile.java
Employee.java
Employee01.java
enhancefor.java
IFELSEIFELSE.java
IFELSEstatement.java
IFstatement.java
instanceofexample.java
LuoJi.java
PrimitiveTypeTest.java
Task List
whilestatement.java
dowhile.java
enhancefor.java
//enhancefor.java
/*Java5 引入了一种主要用于数组的增强型 for 循环.*/
package ch002;
public class enhancefor {
public static void main(String args[]){
int [] numbers = {10, 20, 30, 40, 50};
/*int x为声明语句：声明新的局部变量，
* 该变量的类型必须和数组元素的类型匹配。
* 其作用域限定在循环语句块，其值与此时数组元素的值相等。*/
for(int x : numbers){
/*numbers为表达式：表达式是要访问的数组名，或者是返回值为数组的方法。*/
System.out.print(x);
System.out.print(",");
}
System.out.print("\n");
String [] names ={"James", "Larry", "Tom", "Lacy"};
for(String name : names) {
System.out.print(name);
System.out.print(",");
}
}
}
特殊 for 循环
Console
<terminated> enhancefor [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (2017年10月22
10,20,30,40,50,
James,Larry,Tom,Lacy,

嵌套循环

- 将一个循环放在另一个循环体内，就形成了嵌套循环。其中， for ,while ,do…while 均可以作为外层循环和内层循环。

- 实质上，嵌套循环就是把内层循环当成外层循环的循环体。当只有内层循环的循环条件为 false 时，才会完全跳出内层循环，才可结束外层的当次循环，开始下一次的循环。

- 设外层循环次数为 m 次，内层为 n 次，则内层循环体实际上需要执行 m*n=mn 次。

例题：1）九九乘法表
2）1—100 之间的所有质数

for 循环和 for 循环嵌套

特殊流程控制语句 1

break 主要用在循环语句或者 switch 语句中，用来跳出整个语句块。

break 跳出最里层的循环，并且继续执行该循环下面的语句。

break 完全跳出

特殊流程控制语句 1

- break 语句用法举例

```
public class TestBreak{
        public static void main(String args[]){
      for(int i = 0; i<10; i++){
            if(i==3)
                break;
            System.out.println(" i =" + i);
      }
      System.out.println("Game Over!");
        }
}
```

break 完全跳出

特殊流程控制语句 2

- **continue 语句**
  - continue 语句用于跳过某个循环语句块的**一次**执行
  - continue 语句出现在多层嵌套的循环语句体中时，可以通过标签指明要跳过的是哪一层循环

- continue 语句用法举例

```
public class ContinueTest {
        public static void main(String args[]){
    for (int i = 0; i < 100; i++) {
            if (i%10==0)
                    continue;
                System.out.println(i);
            }
        }
}
```

continue 跳出当前这次循环

特殊流程控制语句 3

- return ：并非专门用于结束循环的，它的功能是结束一个方法。当一个方法执行到一个 return 语句时，这个方法将被结束。

- 与 break 和 continue 不同的是，**return 直接结束整个方法**，不管这个 return 处于多少层循环之内

特殊流程控制语句说明

- break 只能用于 **switch 语句**和**循环语句**中。
- continue 只能用于**循环语句**中。
- 二者功能类似，但 continue 是终止**本次**循环， break 是终止**本层**循环。
- break 、 continue 之后不能有其他的语句，因为程序永远不会执行其后的语句。
- 标号语句必须紧接在循环的头部。标号语句不能用在非循环语句的前面。

第 3 讲 Java 语法基础 -2

3.1 流程控制 (选择、循环)

3.2 数组

3.3 示例学习

数 组

在 Java 语言中，数组是一种最简单的复合数据类型（引用数据类型）。

- 数组是有序数据的集合，数组中的每个元素具有相同的数据类型，
- 可以声明任何类型的数组——原始类型或类类型，
- 可以象创建对象一样，使用关键字 new 创建一个数组。
- 可以用一个统一的数组名和下标来唯一地确定数组中的元素，所有数组的下标都从 0 开始。

数组

创建数组

Java语言使用new操作符来创建数组，语法如下：

```
arrayRefVar = new dataType[arraySize];
```

上面的语法语句做了两件事：

- 一、使用 dataType[arraySize] 创建了一个数组。
- 二、把新创建的数组的引用赋值给变量 arrayRefVar。

数组变量的声明，和创建数组可以用一条语句完成，如下所示：

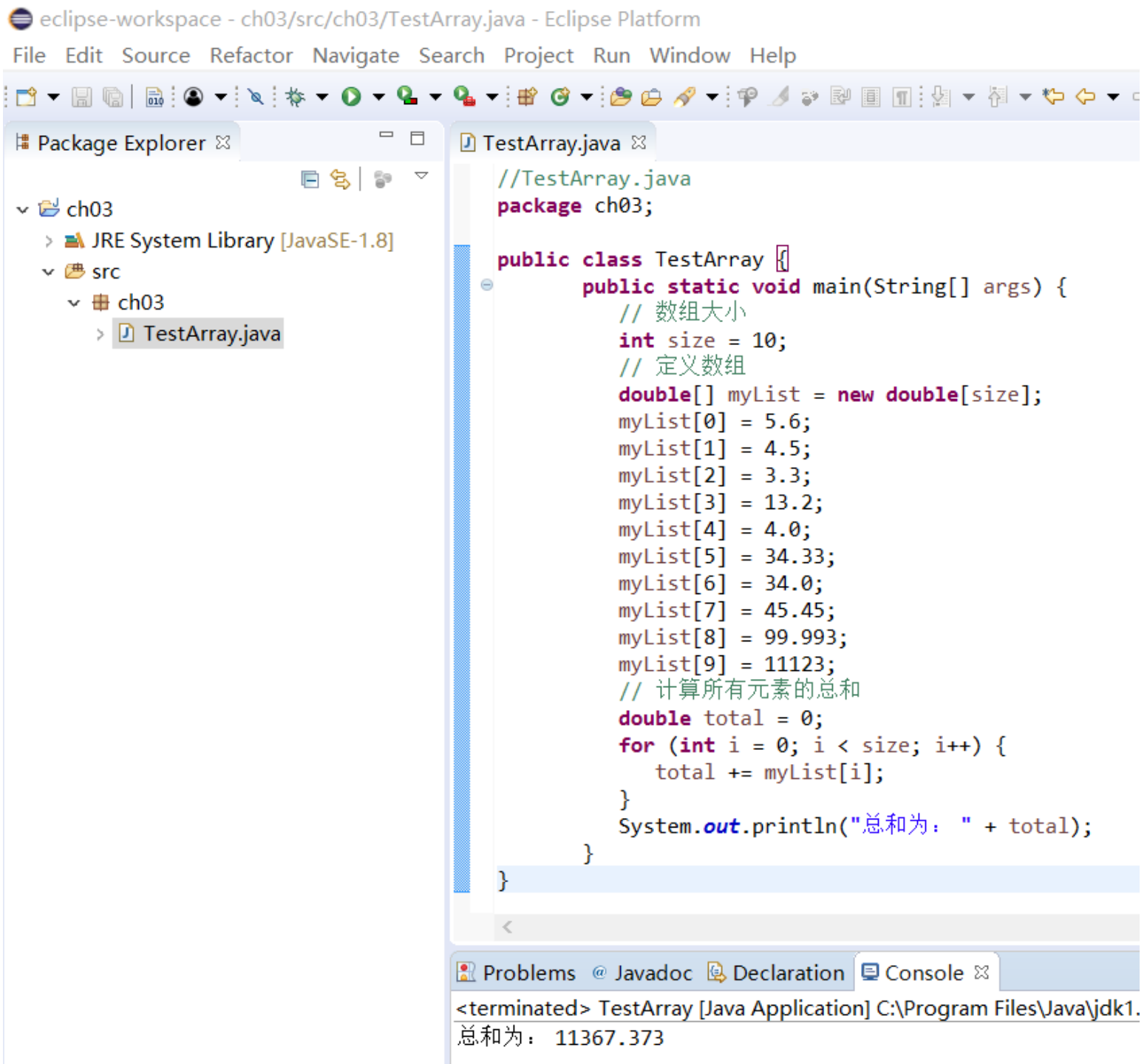
```
dataType[] arrayRefVar = new dataType[arraySize];
```

另外，你还可以使用如下的方式创建数组。

```
dataType[] arrayRefVar = {value0, value1, ..., valuek};
```

数组的元素是通过索引访问的。数组索引从 0 开始，所以索引值从 0 到 arrayRefVar.length-1。

创建数组
1）先定义，再赋值
2）枚举方式



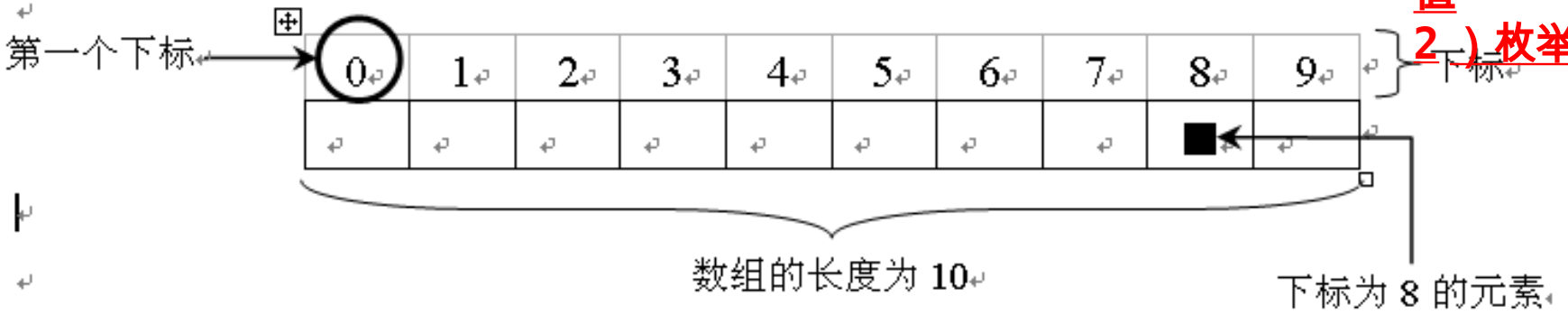
创建数组

1）先定义，再赋值

数 组

•数组的长度在数组创建的时候就已经确定。一旦创建以后，数组就有了固定长度。如图所示，数组的长度为 10 ，第一个下标为 0 ，下标为 8 的元素为第 9 个元素。

创建数组
1 ）先定义，再赋值
2 ）枚举方式



•数组元素就是数组中的一个成员，可以通过数组中的位置来访问它。如果需要在一个结构中存储不同类型的数据，或者需要一个长度可以动态改变的结构，可以考虑使用 Vector 类型而不使用数组。

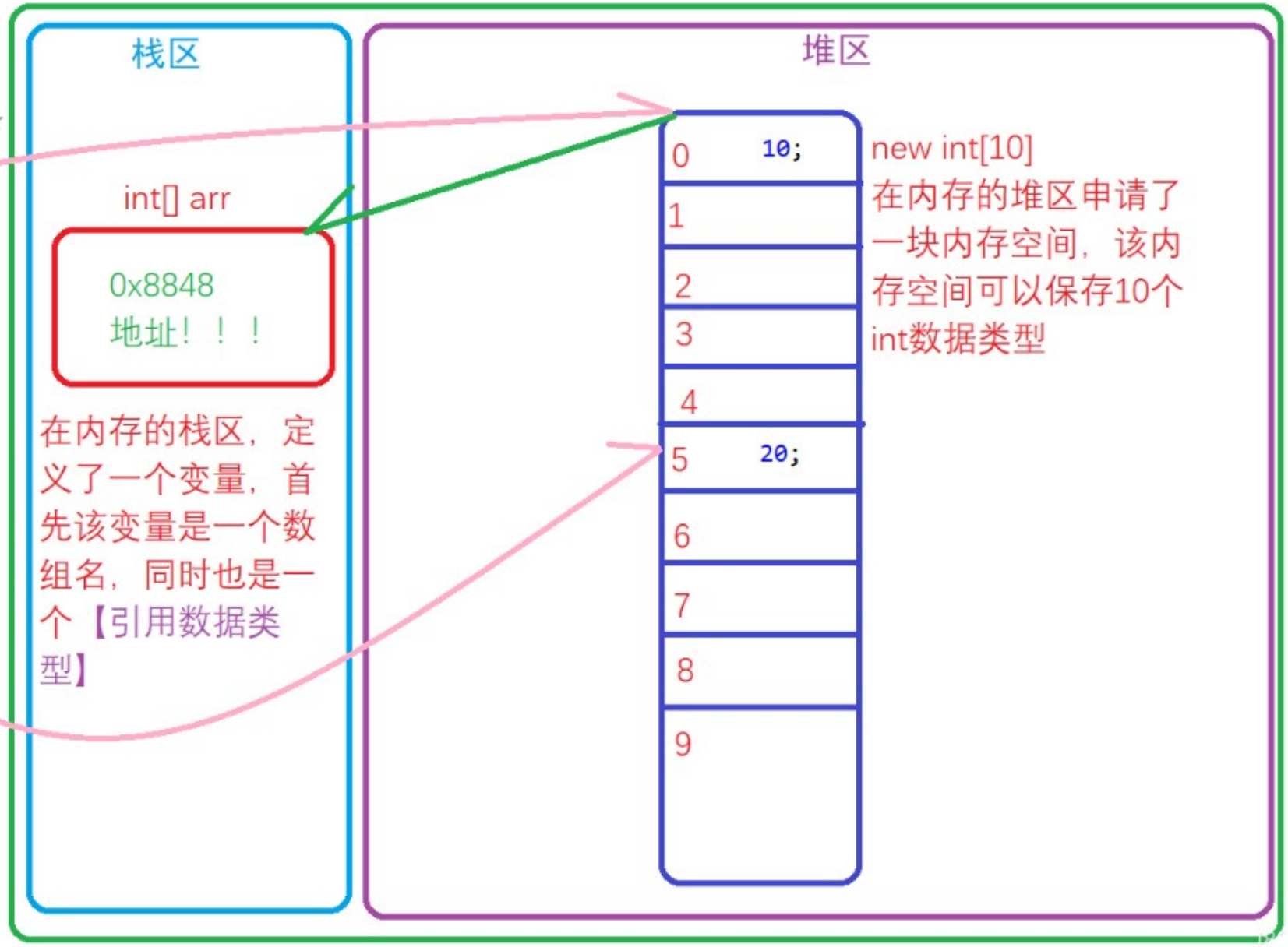
```java
/*
 定义一个int类型数组
 数据类型[] 数组名 = new 数据类型[容量];
 这里创建一个可以保存10个int类型数据的数组，数组名是 arr
*/
int[] arr = new int[10];

// 给予数组中下标为0的元素赋值
arr[0] = 10;

// 给予数组中下标为5的元素赋值
arr[5] = 20;

// 展示数组中下标为0 和下标为 5数据保存的值
System.out.println(arr[0]);
System.out.println(arr[5]);
```

创建数组



创建和使用数组

1. 声明一个数组

   int[] anArray;

2. 创建一个数组

   可使用 Java 的 new 运算符来创建一个数组

   anArray = new int[10];

3. 数组初始化程序

   boolean[] answers = { true, false, true, true, false };

4. 访问数组元素

   anArray[i] = i;

5. 确定数组的大小

   arrayname.length

创建数组

对象数组

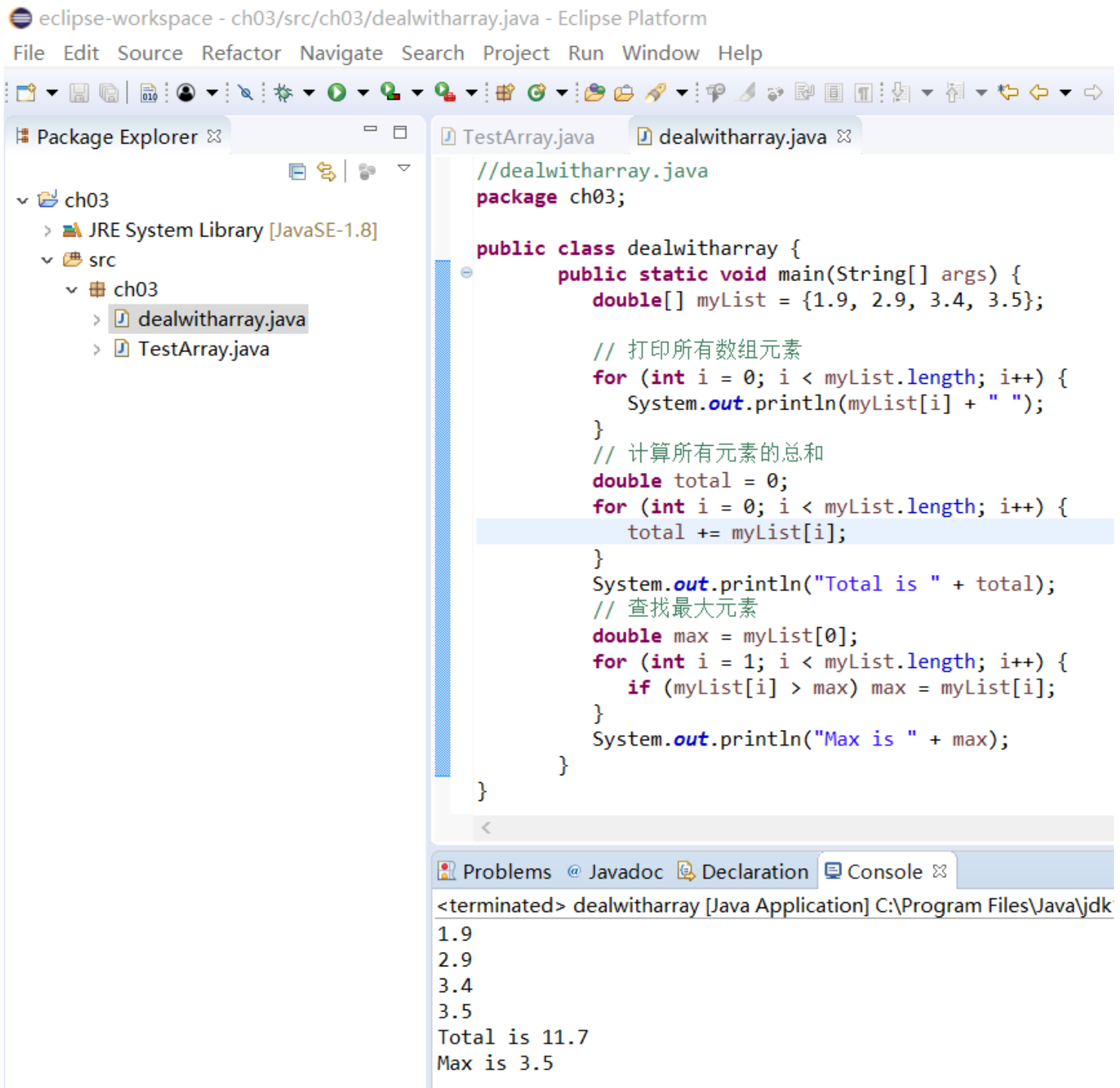
- 由同类型的对象为数组元素组成的数组称为对象数组。

数组可用于保存引用类型的多个对象。

对象数组

- 例 编写一个程序创建一个包含 3 个 String 对象的数组，

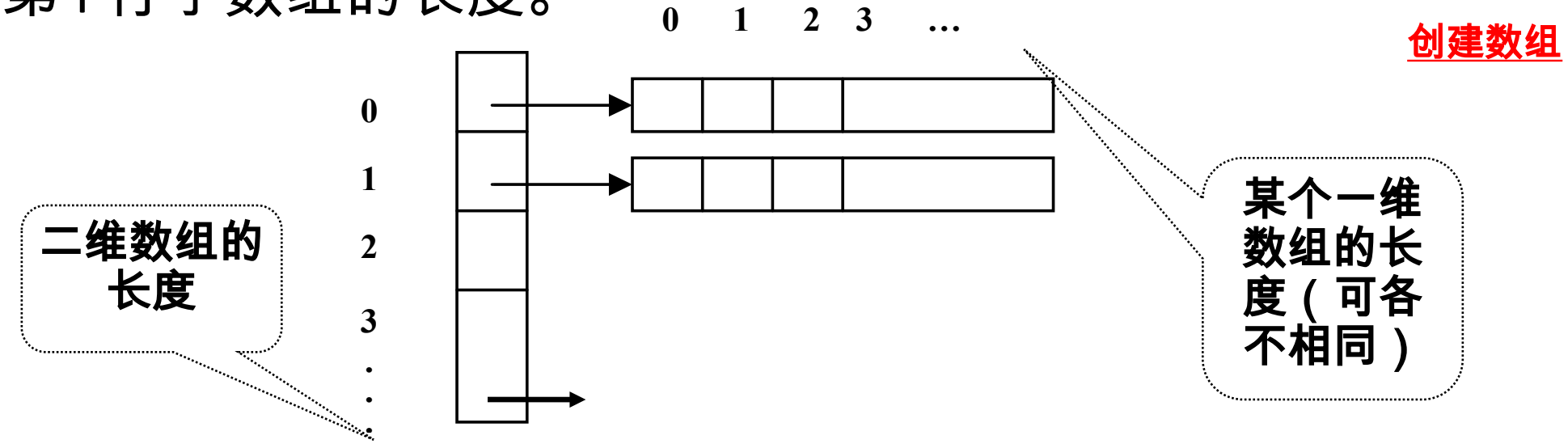
并且将这 3 个字符串以小写字母的形式打印出来。


eclipse-workspace - ch03/src/ch03/dealwitharray.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ch03
JRE System Library [JavaSE-1.8]
src
ch03
dealwitharray.java
TestArray.java
TestArray.java
dealwitharray.java
//dealwitharray.java
package ch03;
public class dealwitharray {
public static void main(String[] args) {
double[] myList = {1.9, 2.9, 3.4, 3.5};
// 打印所有数组元素
for (int i = 0; i < myList.length; i++) {
System.out.println(myList[i] + " ");
}
// 计算所有元素的总和
double total = 0;
for (int i = 0; i < myList.length; i++) {
total += myList[i];
}
System.out.println("Total is " + total);
// 查找最大元素
double max = myList[0];
for (int i = 1; i < myList.length; i++) {
if (myList[i] > max) max = myList[i];
}
System.out.println("Max is " + max);
}
}
Problems Javadoc Declaration Console
<terminated> dealwitharray [Java Application] C:\Program Files\Java\jdk
1.9
2.9
3.4
3.5
Total is 11.7
Max is 3.5

数组

多维数组

- Java 的二维数组实质上是一维数组的数组，如图所示。这个二维数组可用 arrayOfInts.length 代表其长度，该长度为一维数组的个数。 arrayOfInts[i].length 表示第 i 行子数组的长度。



- 数组是多个相同类型数据的组合，实现对这些数据的统一管理
- 数组中的元素可以是任何数据类型，包括基本数据类型和引用数据类型
- 数组属引用类型，数组型数据是对象（object），数组中的每个元素相当于该对象的成员变量

foreach 循环

JDK 1.5 引进了一种新的循环类型，被称为 foreach 循环或者加强型循环，它能在不使用下标的情况下遍历数组。

示例

该实例用来显示数组myList中的所有元素：

foreach 循环

TestArray.java 文件代码：

```
public class TestArray {
   public static void main(String[] args) {
      double[] myList = {1.9, 2.9, 3.4, 3.5};

      // 打印所有数组元素
      for (double element: myList) {
         System.out.println(element);
      }
   }
}
```

以上实例编译运行结果如下：

```
1.9
2.9
3.4
3.5
```

创建对象数组(1)

- 创建元素为引用类型（对象）的数组

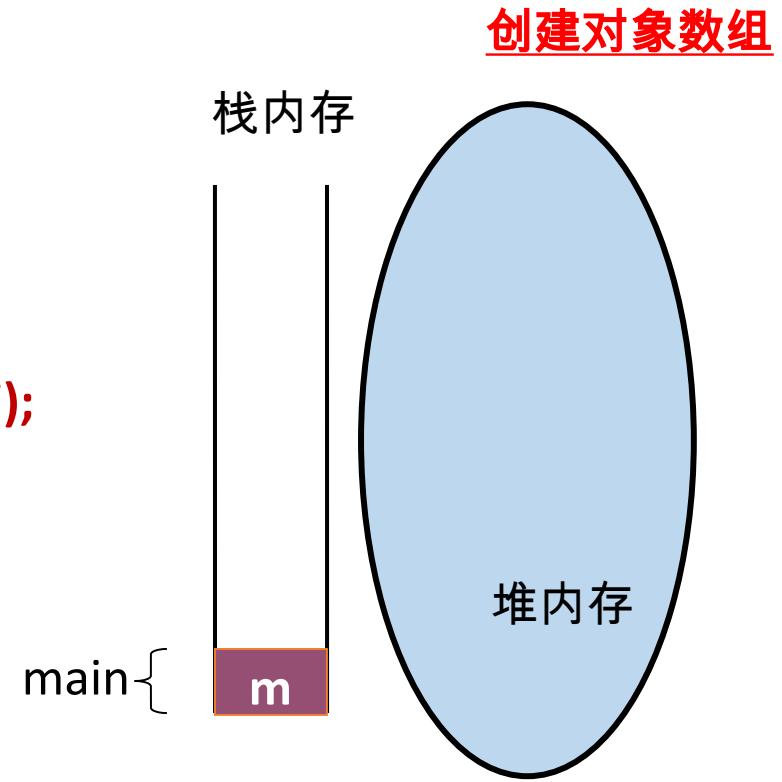
```
class MyDate{
    private int day;
    private int month;
    private int year;
    public MyDate(int d, int m, int y){
            day = d;    month = m;    year = y;
    }
    public void display(){
            System.out.println(year + "-" + month + "-" + day);
    }
}
```

创建对象数组（2）

- 创建元素为引用类型（对象）的数组演示

```
public class Test{
    public static void main(String args[]){
                MyDate[] m;
                m = new MyDate[10];
                for ( int i=0; i<10; i++ ) {
                    m[i] =new MyDate(i+1, i+1,1990+i);
                    m[i].display();
            }
    }
}
```

处内存状态



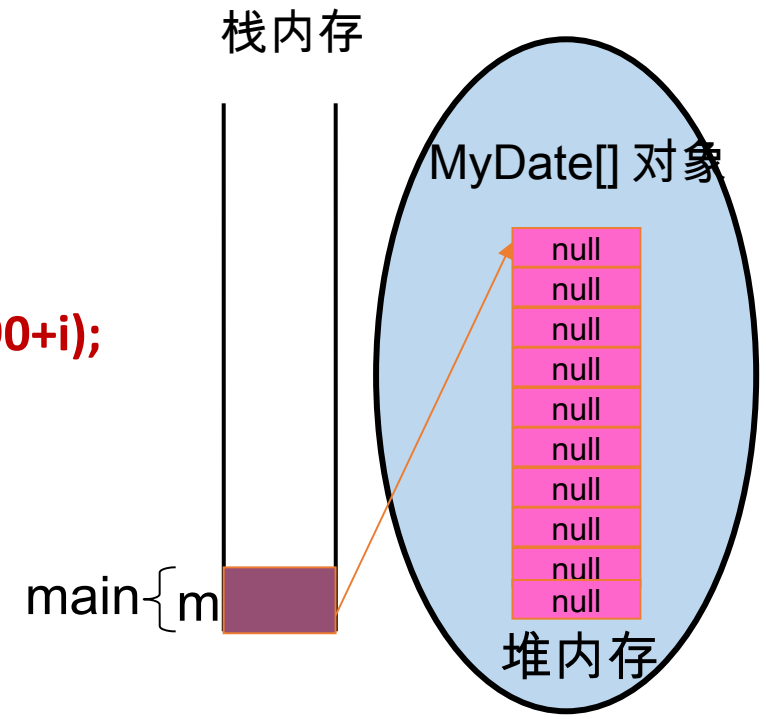
创建对象数组（3)

- 创建元素为引用类型（对象）的数组演示

```
public class Test{
    public static void main(String args[]){
                MyDate[] m;
                m = new MyDate[10];
                for ( int i=0; i<10; i++ ) {
                  m[i] =new MyDate(i+1, i+1,1990+i);
                  m[i].display();
            }
    }
}
```

处内存状态

对象数组存储方式

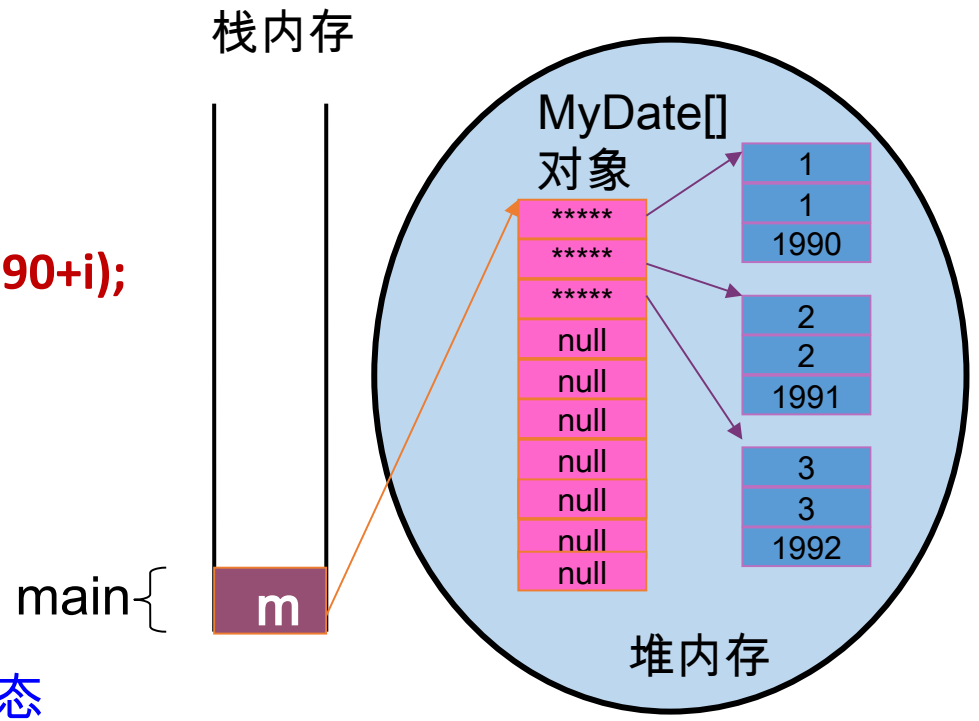


创建对象数组（4)

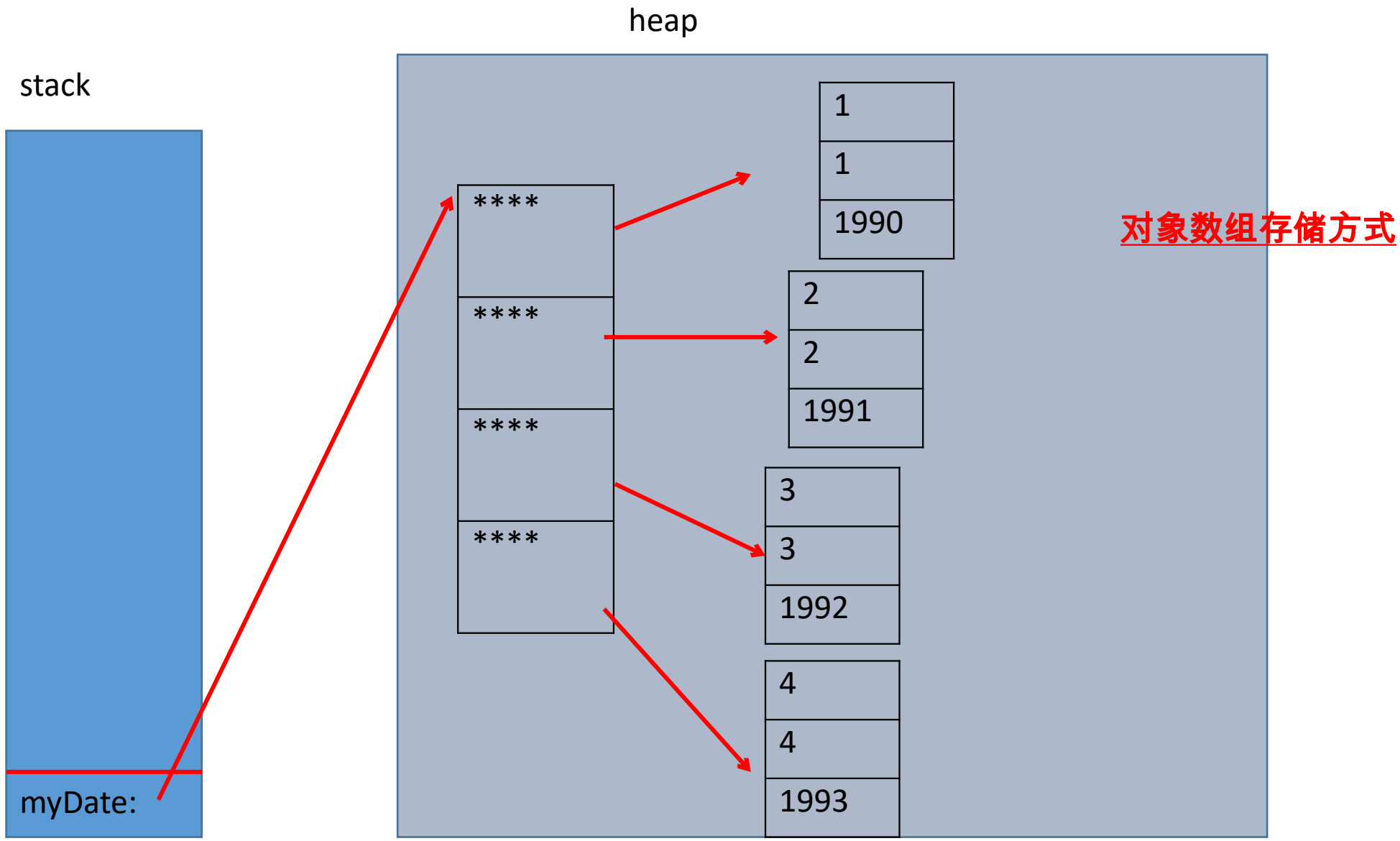
- Java 中使用关键字 new 创建数组对象

```
public class Test{
    public static void main(String args[]){
        MyDate[] m;
        m = new MyDate[10];
        for ( int i=0; i<10; i++ ) {
            m[i] =new MyDate(i+1, i+1,1990+i);
            m[i].display();
        }
    }
}
```

对象数组存储方式



✦ for 循环执行三次后内存状态


heap
stack
myDate:
****
****
****
****
1
1
1990
2
2
1991
3
3
1992
4
4
1993
对象数组存储方式

数组元素的默认初始化

- 数组是引用类型，它的元素相当于类的成员变量，因此数组一经分配空间，其中的每个元素也被按照成员变量同样的方式被隐式初始化。例如：

数组初始化

```
public class Test {
    public static void main(String argv[]){
        int a[]= new int[5];
        System.out.println(a[3]);          //a[3] 的默认值为 0
    }
}
```

  - **对于基本数据类型而言，默认初始化值各有不同**
  - **对于引用数据类型而言，默认初始化值为 null(注意与 0 不同！)**

数组元素的引用

- 定义并用运算符 **new** 为之分配空间后，才可以引用数组中的每个元素；

数组下标

- 数组元素的引用方式：数组名 [数组元素下标]
  - 数组元素下标可以是整型常量或整型表达式。如 a[3] , b[i] , c[6*i];
  - 数组元素下标从 0 开始；长度为 n 的数组合法下标取值范围：0 —>n-1 ；如 int a[]=new int[3]; 可引用的数组元素为 a[0] 、 a[1] 、 a[2]
- 每个数组都有一个属性 **length** 指明它的长度，例如： **a.length** 指明数组 **a** 的长度 (元素个数)
  - 数组一旦初始化，其长度是不可变的

多维数组

二维数组 [][] ：数组中的数组

格式 1（动态初始化）： int[][] arr = new int[3][2];

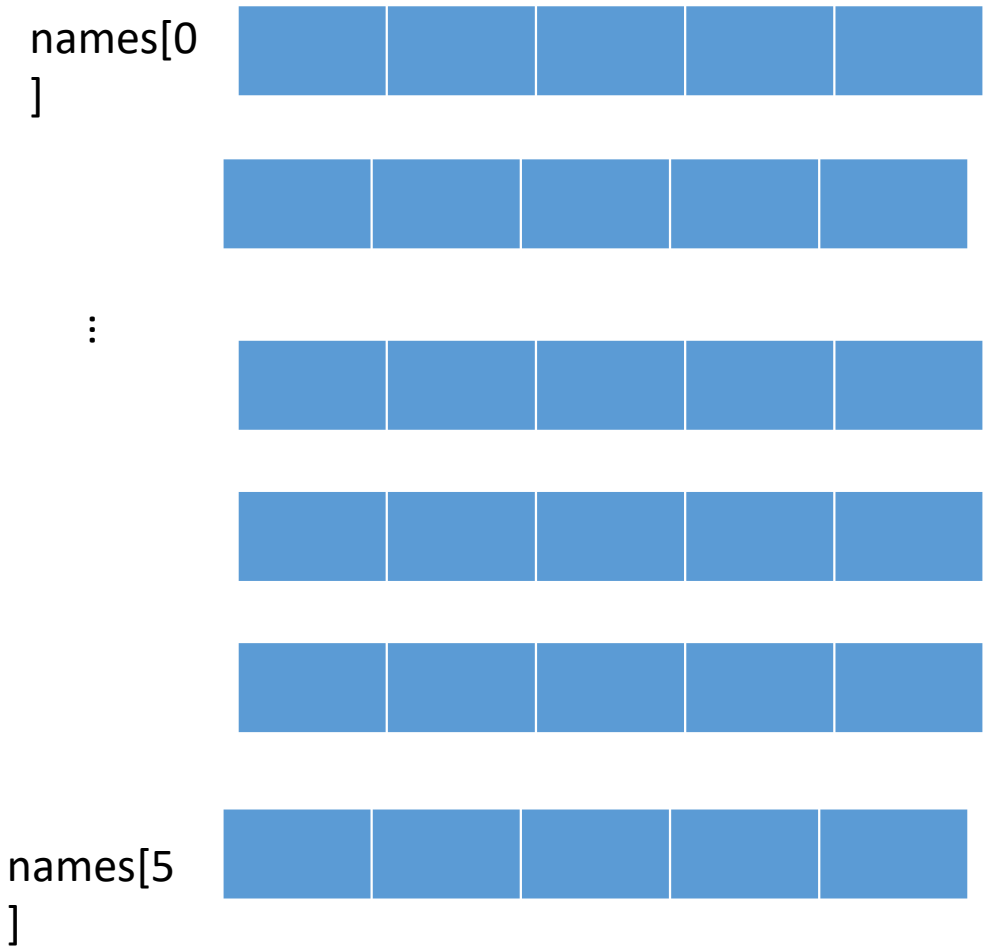
定义了名称为 arr 的二维数组
二维数组中有 3 个一维数组
每一个一维数组中有 2 个元素
一维数组的名称分别为 arr[0], arr[1], arr[2]
给第一个一维数组 1 脚标位赋值为 78 写法是： arr[0][1] = 78;

格式 2（动态初始化）： int[][] arr = new int[3][];

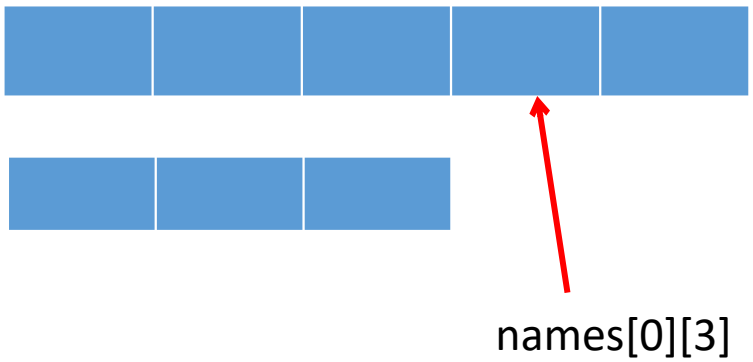
二维数组中有 3 个一维数组。
每个一维数组都是默认初始化值 null（注意：区别于格式 1 ）
可以对这个三个一维数组分别进行初始化
arr[0] = new int[3]; arr[1] = new int[1]; arr[2] = new int[2];
注：
int[][]arr = new int[][3]; // 非法

多维数组

String[][] names;
names = **new String[6][5];**



String[][] names;
names = **new String[2][];**
names[0] = new String[5];
names[1] = new String[3];



多维数组

格式 3（静态初始化）： `int[][] arr = new int[][]{{3,8,2},{2,7},{9,0,1,6}};`

定义一个名称为 arr 的二维数组，二维数组中有三个一维数组
每一个一维数组中具体元素也都已初始化
第一个一维数组 arr[0] = {3,8,2};
第二个一维数组 arr[1] = {2,7};
第三个一维数组 arr[2] = {9,0,1,6};
第三个一维数组的长度表示方式： arr[2].length;

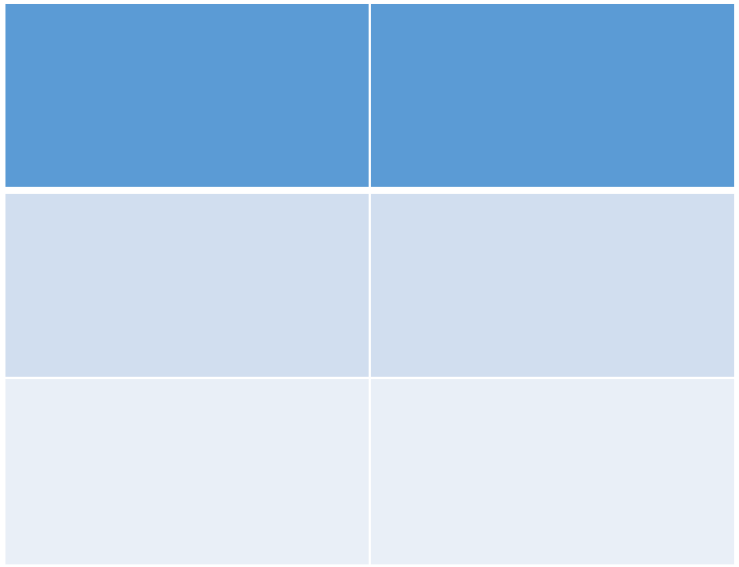
多维数组初始化

- 注意特殊写法情况： int[] x,y[]; x 是一维数组， y 是二维数组。
- Java 中多维数组不必都是规则矩阵形式

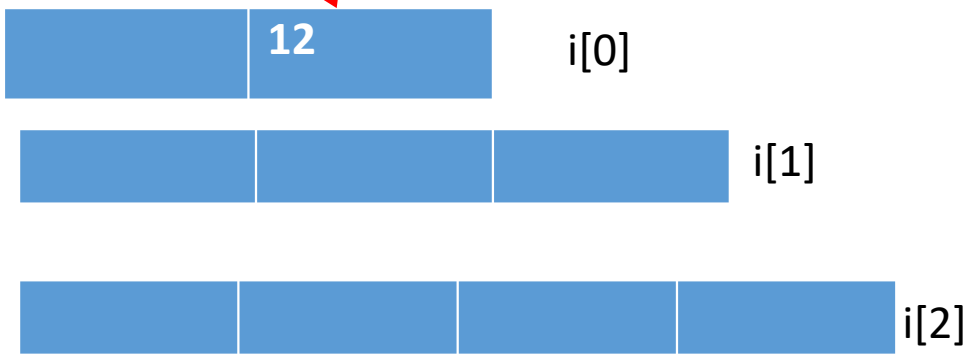
练习：获取 arr 数组中所有元素的和。使用 for 的嵌套循环即可。

j / i	j = 0	j = 1	j = 2	j = 3
i = 0	3	8	2	
i = 1	2	7		
i = 2	9	0	1	6

int[][] i = new int[3][2];



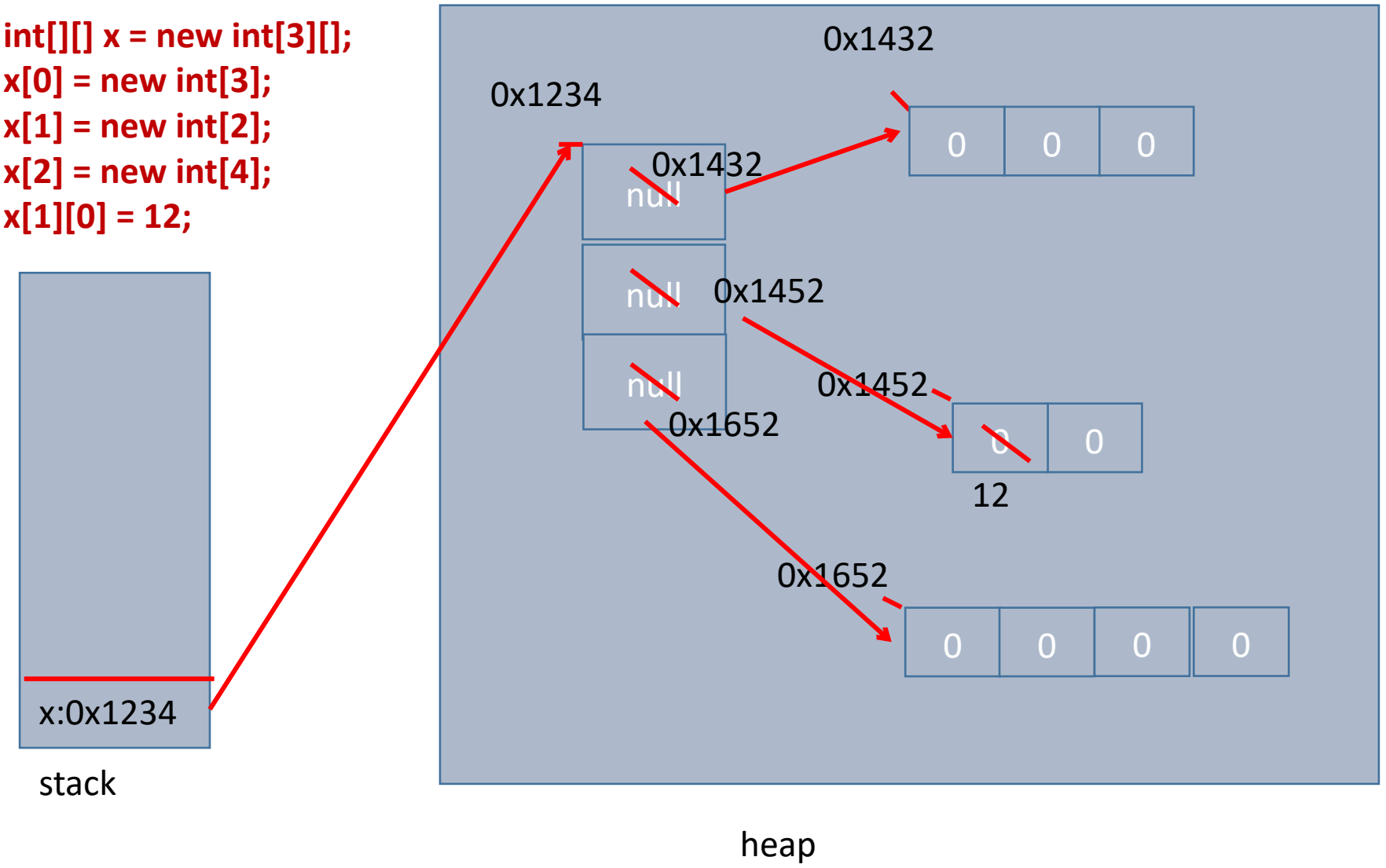
i[0][1] = 12;




```
int[][] i = new int[3][];
i[0] = new int[2];
i[1] = new int[3];
i[2] = new int[4];
```

多维数组初始化

```
int[][] x = new int[3][];
x[0] = new int[3];
x[1] = new int[2];
x[2] = new int[4];
x[1][0] = 12;
```



数组中涉及的常见算法

数组

1. 求数组元素的最大值、最小值、平均数、总和等

2. 数组的复制、反转

3. 数组元素的排序

数组排序

排序

- 插入排序
  - 直接插入排序、折半插入排序、 Shell 排序
- 交换排序
  - 冒泡排序、快速排序（或分区交换排序）
- 选择排序
  - 简单选择排序、堆排序
- 归并排序
- 基数排序

排序方法的选择

排序选择

(1) 若 n 较小 (如 n≤50) ，可采用**直接插入**或**直接选择排序**。

当记录规模较小时，直接插入排序较好；否则因为直接选择移动的记录数少于直接插入，应选直接选择排序为宜。

(2) 若文件初始状态基本有序 (指正序) ，则应选用**直接插入、冒泡**或随机的**快速排序**为宜；

(3) 若 n 较大，则应采用时间复杂度为 O(nlgn) 的排序方法：**快速排序、堆排序**或**归并排序**。

冒泡排序

排序选择

排序思想：

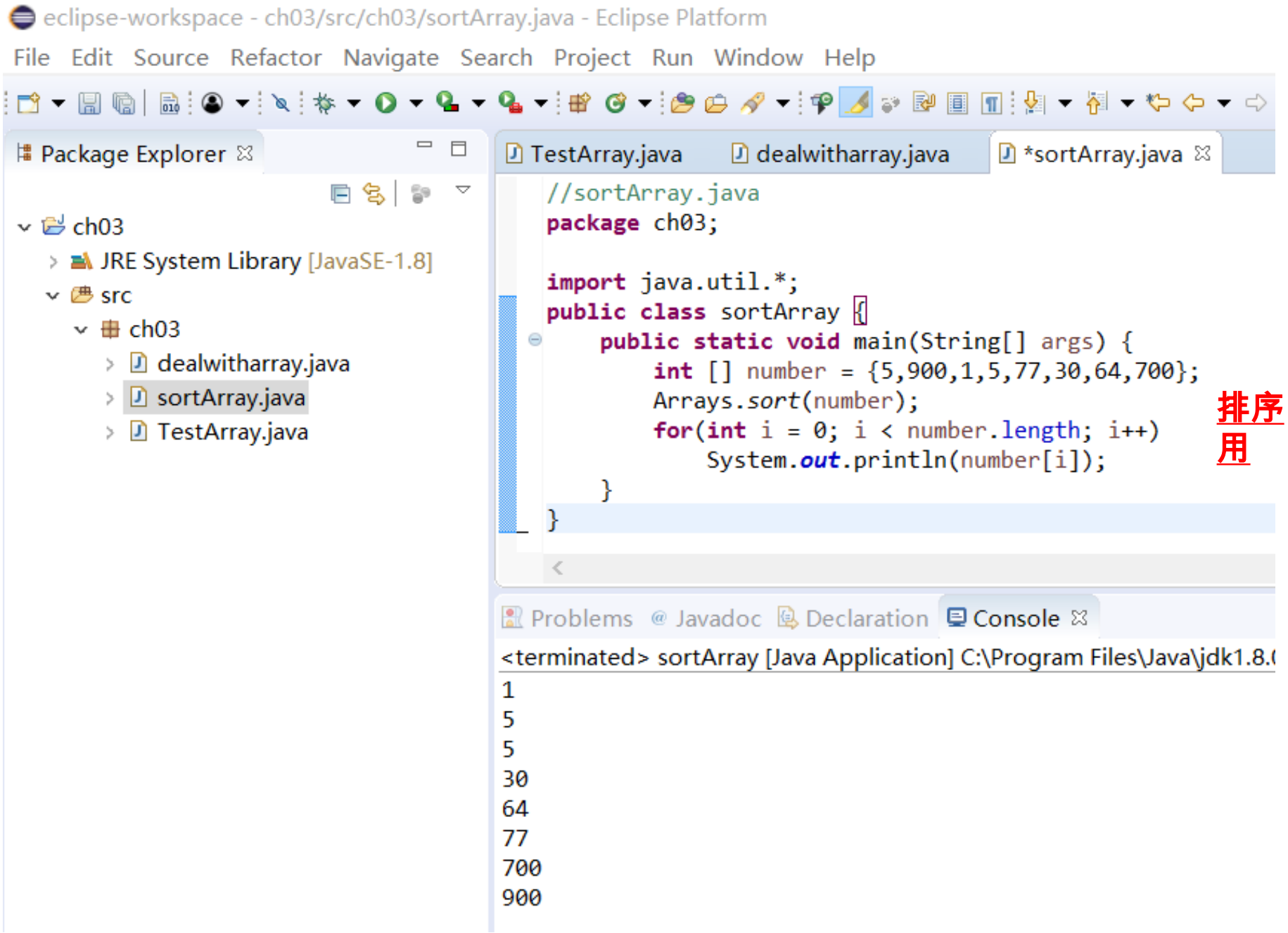
相邻两元素进行比较，如有需要则进行交换，每完成一次循环就将最大元素排在最后（如从小到大排序），下一次循环是将其它的数进行类似操作。

数组排序

排序 API 调用

- java.util.Arrays 类的 sort() 方法提供了数组元素排序功能：

```
import java.util.*;
public class Sort {
        public static void main(String[] args) {
                int [] number = {5,900,1,5,77,30,64,700};
                Arrays.sort(number);
                for(int i = 0; i < number.length; i++)
                        System.out.println(number[i]);
        }
}
```


eclipse-workspace - ch03/src/ch03/sortArray.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
ch03
JRE System Library [JavaSE-1.8]
src
ch03
dealwitharray.java
sortArray.java
TestArray.java
TestArray.java
dealwitharray.java
*sortArray.java
//sortArray.java
package ch03;
import java.util.*;
public class sortArray {
public static void main(String[] args) {
int [] number = {5,900,1,5,77,30,64,700};
Arrays.sort(number);
for(int i = 0; i < number.length; i++)
System.out.println(number[i]);
}
}
排序 API 调用
Problems Javadoc Declaration Console
<terminated> sortArray [Java Application] C:\Program Files\Java\jdk1.8.
1
5
5
30
64
77
700
900

操作数组的工具类： Arrays

- java.util.Arrays 类包含了用来操作数组（比如排序和搜索）的各种方法。 Arrays 拥有一组 static 方法。
  - **equals()** ：比较两个 array 是否相等。 array 拥有相同元素个数，且所有对应元素两两相等。
  - **fill()** ：将值填入 array 中。
  - **sort()** ：用来对 array 进行排序。
  - **binarySearch()** ：在排好序的 array 中寻找元素。

    另： **System.arraycopy()** ： array 的复制。

数组操作常见问题

编译时，不报错！！

数组脚标越界异常 (ArrayIndexOutOfBoundsException)

```
int[] arr = new int[2];
System.out.println(arr[2]);
```

访问到了数组中的不存在的脚标时发生。

空指针异常 (NullPointerException)

```
int[] arr = null;
System.out.println(arr[0]);
```

arr 引用没有指向实体，却在操作实体中的元素时。

第 3 讲 Java 语法基础 -2

3.1 流程控制 (选择、循环)

3.2 数组

3.3 数组的变种 / 示例

Java 中几种常见的数据结构

1 ． ArrayList: 一个可以动态修改的数组，元素单个，效率高，多用于查询（**变长数组**）

数组变种

2 ． Vector: 元素单个，线程安全，多用于查询（可以实现自动增长的**对象数组**，**变长数组**）

3 ． LinkedList: 元素单个，多用于插入和删除（**链表**）

4 ． HashMap: 元素成对，元素可为空（**散列表**，它存储的内容是键值对 (key-value) 映射；实现了 Map 接口）

5 ． HashTable: 元素成对，线程安全，元素不可为空（**散列表**，它存储的内容是键值对 (key-value) 映射）

6 ． TreeMap: 元素成对，元素可为空（实现 SortMap 接口，是一个有序的 key-value 集合，数据结构是**红黑树**）

问题：编写一系统实现添加和查找员工信息功能

分析：

1. 员工信息可用一个类 Staff 专门记录

可变长

2. 员工信息的添加和查找操作可用 ArrayList 类实现

职工类记录职工基本信息

```java
//职员类
class Staff
{
    private String number;
    private String name;
    private float sal;

    public Staff(String number,String name,float sal)
    {
        this.setNumber(number);
        this.setName(name);
        this.setSal(sal);
    }
}
```

Managestaff: 处理职工信息的相关操作

*ArrayListSample.java | VectorSample.java

ArrayList

```java
class Managestaff
{
    private ArrayList al=null;
    public Managestaff()
    {
        al=new ArrayList();
    }
    //加入员工
    public void Addstaff(Staff staff)
    {
        al.add(staff);
        System.out.println("员工添加成功");
    }
    //根据员工号显示员工信息
    public void Showstaff(String number_cha)
    {
        //遍历所有员工
        for(int i=0;i<al.size();i++)
        {
            Staff staff=(Staff)al.get(i);
            if(staff.getNumber().equals(number_cha))
            {
                System.out.println("所找员工的工号为: "+staff.getNumber()+", 姓名为: "
                        +staff.getName()+", 薪水为: "+staff.getSal());
            }
        }
    }
}
```

源码实例

*ArrayListSample.java | VectorSample.java

Array List

```
76 public class ArrayListSample {
77     public static void main(String[] args) throws Exception{
78         Managestaff managestaff= new Managestaff();
79         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
80         while(true)
81         {
82             System.out.println("功能选择: ");
83             System.out.println("1、添加员工");
84             System.out.println("2、查找员工");
85             String operaMenu=br.readLine();
86             switch(operaMenu)
87             {
88                 case "1":
89                 {
90                     System.out.println("请输入新增员工编号: ");
91                     String number=br.readLine();
92                     System.out.println("请输入新增员工姓名: ");
93                     String name=br.readLine();
94                     System.out.println("请输入新增员工薪水: ");
95                     float sal=Float.parseFloat(br.readLine());
96                     Staff staff=new Staff(number,name,sal);
97                     managestaff.Addstaff(staff);
98                 }
99                 break;
100                case "2":
101                {
102                    System.out.println("请输入要查找员工编号: ");
103                    String number=br.readLine();
104                    managestaff.Showstaff(number);
```

运行效果

Array List

问题：随机输入一字符串并读入一文件最后再读取出来

Vector

分析：

1. 文件的读取读入操作使用 DataInputStream 和 DataOutputStream 类

2. 输入字符串信息可用 Vector 类存储

文件的写入操作：

```
DataOutputStream out = new DataOutputStream(new
FileOutputStream("vectorText.txt"));
```

文件的读取操作：

```
DataInputStream in = new DataInputStream(new
                                    FileInputStream("vectorText.txt"));
```

源码实例

ArrayListSample.java | VectorSample.java

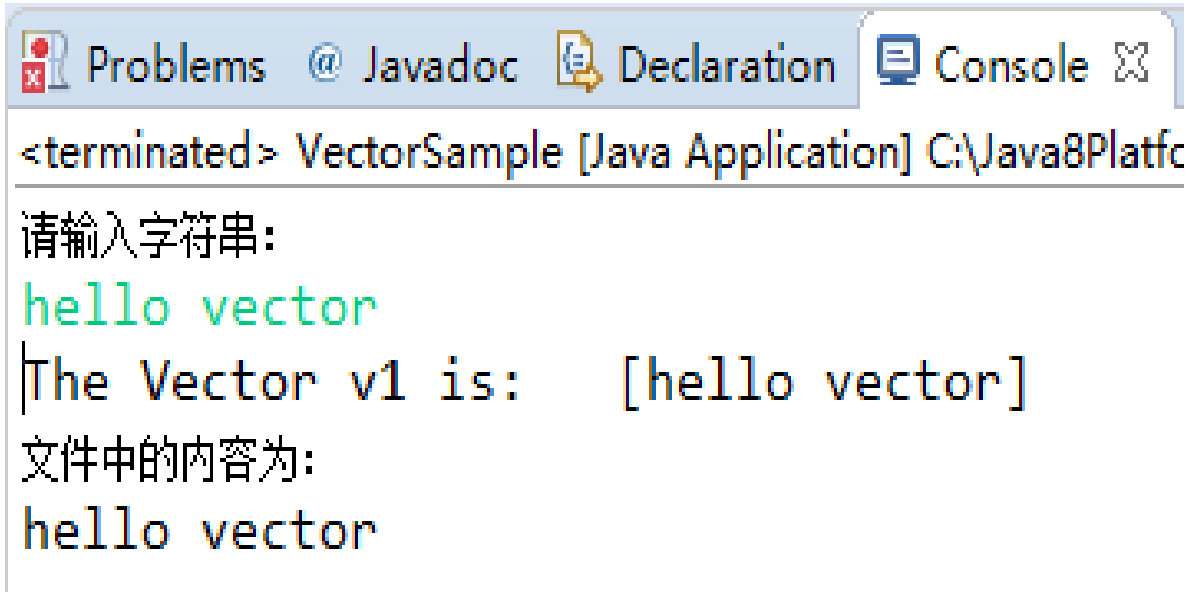
```java
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;
import java.util.Vector;

public class VectorSample {
    public static void main(String[] args) throws FileNotFoundException {
        Vector<String> v1 = new Vector<String>();
        Scanner scn = new Scanner(System.in);
        v1.addElement(scn.nextLine());
        //转为字符串并打印
         System.out.println("The Vector v1 is:\n\t"+v1);
         try {
            DataOutputStream out = new DataOutputStream(new FileOutputStream("vectorText.txt"));
            for (int i = 0; i < v1.size(); i++) {
                  String str = (String) v1.elementAt(i);
                  out.writeBytes(str + "");
                }
            out.close();
            DataInputStream in = new DataInputStream(new FileInputStream("vectorText.txt"));
            System.out.println(in.readLine());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Vector

运行效果

Vecto
r


Problems
Javadoc
Declaration
Console
<terminated> VectorSample [Java Application] C:\Java8Platfc
请输入字符串:
hello vector
The Vector v1 is: [hello vector]
文件中的内容为:
hello vector

问题：实现一个类型栈堆或队列的功能容器

LinkedList

分析：

1. 封装一个类似类型栈堆或队列的功能容器 linkedlist_test1

2. 使用 LinkedList 类对该容器进行处理

功能容器职工类记录职工基本信息

```
class linkedlist_test1 {
    /*实现类似栈堆或队列功能的容器*/
    private LinkedList linkedlist;
    //构造函数，一开始就生成一个链表队列linkedlist
    linkedlist_test1(){
        linkedlist = new LinkedList();
    }
    //定义向链表队列添加元素的功能
    public void addElement(Object obj){
        linkedlist.add(obj);
    }
    //定义一个类似栈堆功能(先进后出)
    public Object ZhanDui(){
        return linkedlist.removeFirst();
    }
    //定义一个类似队列功能(先进先出)
    public Object DuiLie(){
        return linkedlist.removeLast();
    }
    //定义一个判断linkedlsit是否为空的功能
    public boolean isNull(){
        return  linkedlist.isEmpty();
    }
}
```

LinkedList

源码实例

ArrayListSample.java | VectorSample.java | LinkedListSample.java

```
28 }
29
30 public class LinkedListSample {
31     public static void main(String[] args){
32         linkedlist_test1 test = new linkedlist_test1();
33         test.addElement(1);
34         test.addElement("you");
35         test.addElement("and");
36         //栈堆功能
37 //        while (!test.isNull()){
38 //            System.out.println(test.DuiLie());
39 //        }
40         //队列功能
41         System.out.println("队列为:");
42         for(;!test.isNull();){
43             System.out.println(test.ZhanDui());
44         }
45     }
46 }
```

LinkedList

运行效果：

容器实现队列功能

LinkedList

问题：控制台输入一句英语，简单统计各个单词出现的次数

HashMap

分析：

1. 使用 HashMap 类存储英语字符串

2. 使用 Iterator 迭代器循环输出相应结果

HashMap 相关函数操作

HashMap

1.containsKey(Object key)方法，返回值为boolean，用于判断当前hashmap中是否包含key对应的key-value

2.containsValue(Object value)方法，返回值为boolean，用于判断当前hashmap中是否包含value对应的key-value

3.遍历hashmap的两种方式：

（1）利用haspmap.entrySet().iterator()：利用迭代器，从Entry中取出键、取出值，推荐使用这种方式进行遍历，效率较高

（2）利用hashmap.keySet().iterator()：利用键的迭代器，每次取出一个键，再根据键，从hashmap中取出值，这种方式的效率不高，不推荐使用

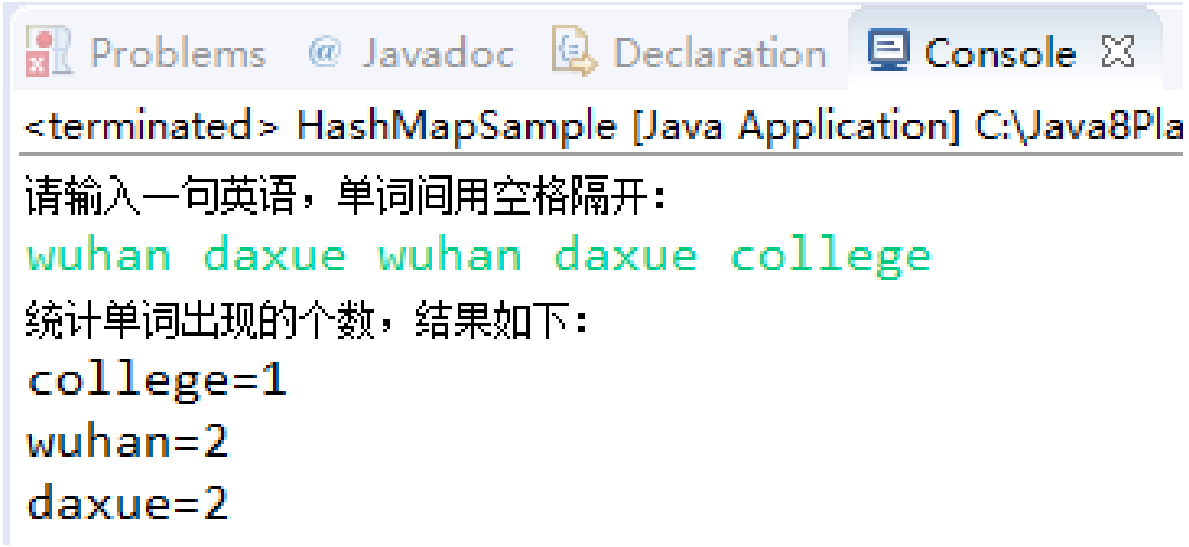
源码实例

ArrayListSample.java | VectorSample.java | LinkedListSample.java | HashMapSample.java

```java
public class HashMapSample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入一句英语，单词间用空格隔开：");
        String sentence = sc.nextLine();
        String[] arr = sentence.split(" ");
        // 键代表着单词，值代表着次数 map.put
        Map<String, Integer> map = new HashMap<String, Integer>();
        for (int i = 0; i < arr.length; i++) {
            if (!map.containsKey(arr[i])) {
                map.put(arr[i], 1);
            } else {
                // 说明map中，存在该元素
                int num = map.get(arr[i]);
                map.put(arr[i], ++num);
            }
        }
        System.out.println("统计单词出现的个数，结果如下：");
        Set<String> set = map.keySet();
        for (Iterator<String> iterator = set.iterator(); iterator.hasNext();) {
            String key = iterator.next();
            Integer value = map.get(key);
            System.out.println(key + "=" + value);
        }
    }
}
```

HashMap

运行效果：


Problems
Javadoc
Declaration
Console
<terminated> HashMapSample [Java Application] C:\Java8Pla
请输入一句英语，单词间用空格隔开：
wuhan daxue wuhan daxue college
统计单词出现的个数，结果如下：
college=1
wuhan=2
daxue=2

HashTable

问题：编写一系统实现对储户信息的相关操作

分析：

1. 封装一个类 Saverperson 存储储户的相关信息

2. 使用 Hashtable 类实现对储户信息的操作

类 Saverperson 存储储户的相关信息

```java
class Saverperson{
    String name;
    double money;
    double rate=0.03;

    public double getRate() {
        return rate;
    }
    public void setRate(double rate) {
        this.rate = rate;
    }
    Scanner sc=new Scanner(System.in);
    public Saverperson(String name,double money){
        this.name=name;
        this.money=money;
    }
    public double MonthRate(){
        return rate/12;
    }
    public double account(){
        return money+(money*MonthRate());
    }
    public void display(){
        System.out.println("name:"+this.name+", money:"+this.money);
    }
}
```

HashTable

源码实例

HashTableSample.java

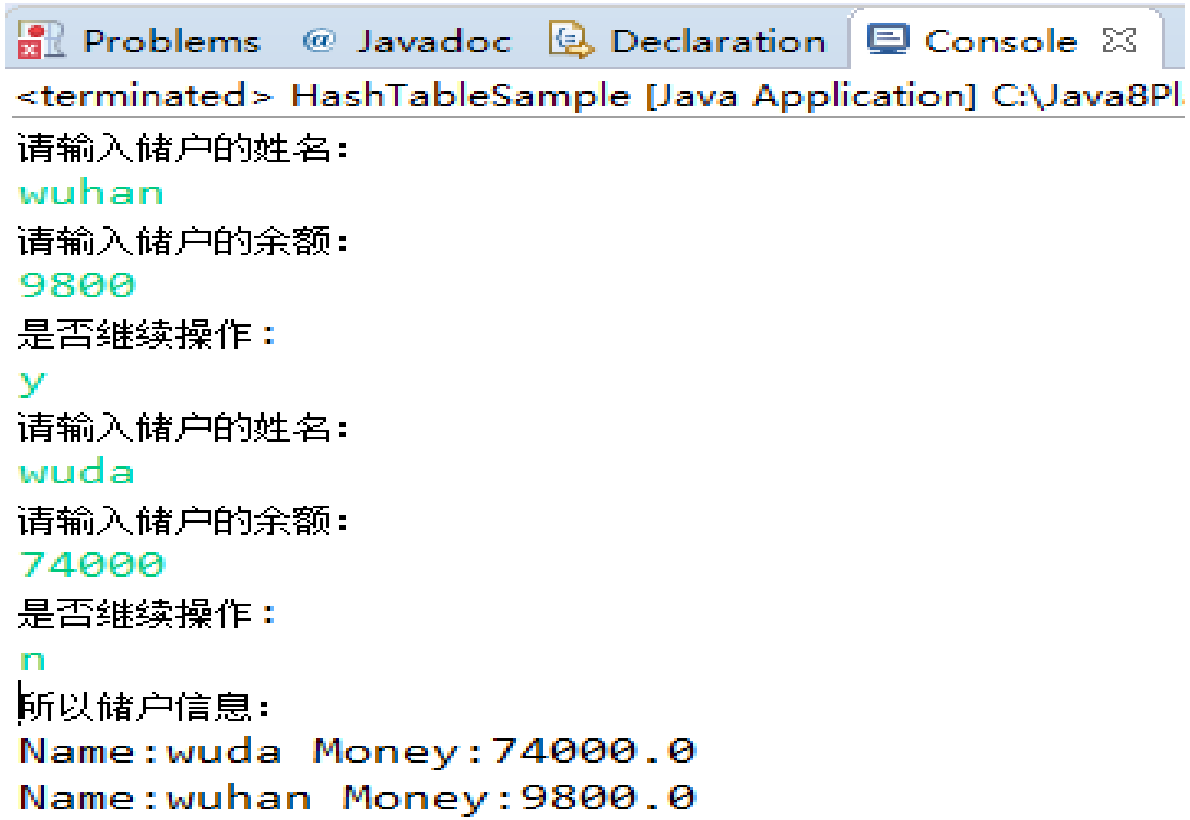
```java
public class HashTableSample {
    public static void main(String[] args) {
        Hashtable<String, Double> Saver=new Hashtable<String, Double>();
        Scanner sc=new Scanner(System.in);
        String temp="";
        do{
            System.out.println("请输入储户的姓名: ");
            String name=sc.next();
            System.out.println("请输入储户的余额: ");
            double money=sc.nextDouble();
            Saverperson saverperson=new Saverperson(name,money);
            Saver.put(saverperson.name, saverperson.money);
            System.out.println("是否继续操作:");
            temp=sc.next();
        }while(temp.equals("Y") || temp.equals("y"));

        System.out.println("所以储户信息: ");
        Enumeration<String> em=Saver.keys();
        while(em.hasMoreElements()){
            Object key=em.nextElement();
            Object value=Saver.get(key);
            System.out.println("Name:"+key+" Money:"+value);
        }
    }
}
```

HashTable

HashTable

运行效果：



TreeMap

问题：获取输入字符串中每一个字母出现的次数

分析：

1. 使用 TreeMap 类实现对字符串的相关操作

2. 使用 StringBuilder 字符串缓冲区变量输出最终结果

TreeMap 相关函数操作

ceilingEntry(K key)
返回与大于或等于给定键的最小键关联的键 - 值映射，或者null如果没有这样的键。

ceilingKey(K key)
返回大于或等于给定键的最小键，或者null如果没有这样的键。

clear()
从此映射中删除所有映射。

clone()
返回此TreeMap实例的浅表副本。

comparator()
返回用于对此映射中的键进行排序的比较器，或者 null此映射是否使用其键的**自然顺序**。

containsKey(Object key)
true如果此映射包含指定键的映射，则返回。

containsValue(Object value)
返回true如果此映射将一个或多个键映射到指定值。

descendingKeySet()
返回**NavigableSet**此映射中包含的键的逆序视图。

descendingMap()
返回此映射中包含的映射的逆序视图。

entrySet()
返回**Set**此映射中包含的映射的视图。

firstEntry()
返回与此映射中的最小键关联的键 - 值映射，或者null如果映射为空。

firstKey()
返回此映射中当前的第一个（最低）键。

TreeMap

源码实例

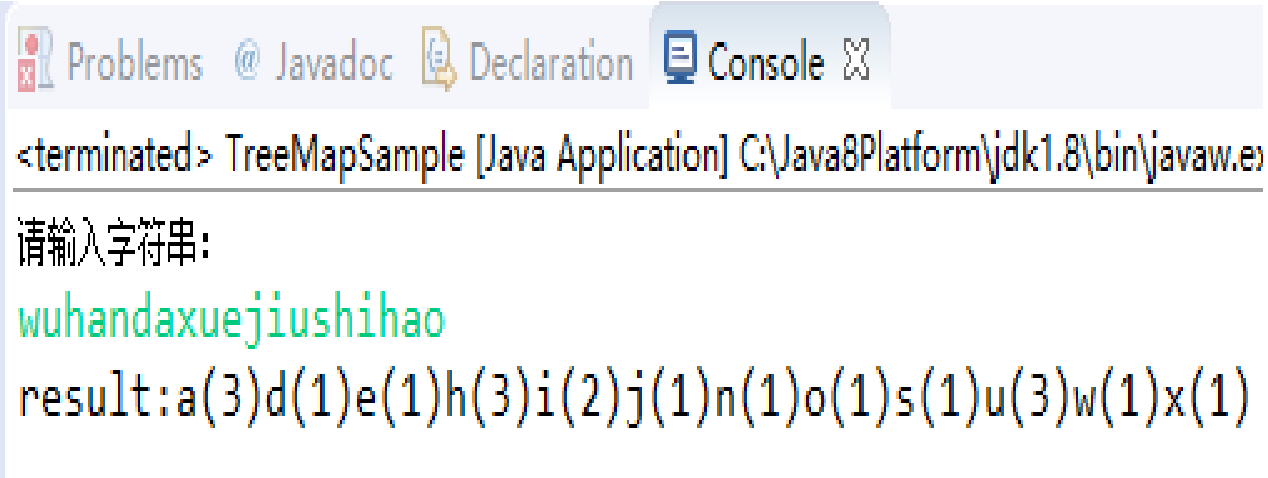
TreeMapSample.java

```
26      public static void main(String[] args) {
27          Scanner sc = new Scanner(System.in);
28          System.out.println("请输入字符串: ");
29          String line=sc.nextLine();
30          TreeMap<Character, Integer> tm = new TreeMap<Character,Integer>();
31          char [] chs=line.toCharArray();
32          for(char key:chs){
33              Integer i=tm.get(key);
34              if(i==null){
35                  tm.put(key, 1);
36              }else{
37                  i++;
38                  tm.put(key, i);
39              }
40          }
41          //定义字符串缓冲区变量
42          StringBuilder sb = new StringBuilder();
43          //遍历集合，得到键和值，进行按照要求拼接
44          Set<Character> set = tm.keySet();
45          for(Character key:set){
46              Integer value=tm.get(key);
47              sb.append(key).append("(").append(value).append(")");
48          }
49          //把字符串缓冲区转换为字符串输出
50          String result=sb.toString();
51          System.out.println("result:"+result);
52      }
53  }
54
```

TreeMap

运行效果：

TreeMap



第三周课堂作业

1 、 Java 的变量名称如何组成？

2 、如何提高程序可读性？

4、请写出下面程序的运行结果

```
1  public class Draw {
2      public static void main(String[] args) {
3          for (int m = 1; m <= 4; m++) {
4              for (int n = 0; n <= m; n++) {
5                  System.out.print(" ");
6              }

7              for (int x = 1; x <= 7 -2 * (m - 1); x++) {
8                  System.out.print("*");
9              }
10             System.out.println();
11         }
12     }
13 }
```

3、请写出下面程序的运行结果

作业

```
1  public class Test1 {

2      private static int a = 2;

3      private static int b = 3;
4
5      public static int MultiplicationOne(int m)

6      {
7          b = a * m;

8          return b;
9      }
10
11     public static int AddTwo(int a, int b)

12     {
13         b += MultiplicationOne(a);
14         return b;

15     }
16
17     public static void main(String[] args)
18     {
19         b = AddTwo(4, 7);
20         System.out.print(b);
21     }
22 }
```

参考

第 2 周作业讲解

1. 面向过程和面向对象有什么区别？各有什么优缺点？

2. 输入一行字符（大于 30 个），分别统计出其中中英文字母、空格、数字和其他字符的个数。

3. 输入一个正整数，对其分解质因数。输出结果间用空格隔开

4. 第 2 讲 ppt 上的所有程序代码编写（与调试）

1、 面向过程和面向对象有什么区别？各有什么优缺点？

参考

参考答案：面向过程与面向对象编程的区别：面向过程就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用就可以了；面向对象是把构成问题事务分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描叙某个事物在整个解决问题的步骤中的行为。

面向过程与面向对象的优缺点：

面向过程：

优点：性能比面向对象高，因为类调用时需要实例化，开销比较大，比较消耗资源，比如单片机、嵌入式开发、Linux/Unix 等一般采用面向过程开发，性能是最重要的因素。

缺点：没有面向对象易维护、易复用、易扩展

面向对象：

优点：易维护、易复用、易扩展，由于面向对象有封装、继承、多态性的特性，可以设计出低耦合的系统，使系统更加灵活、更加易于维护

缺点：性能比面向过程低

输入一行字符（大于 30 个），分别统计出其中中英文字母、空格、数字和其他字符的个数。

参考答案：

参考

```
import java.util.Scanner;
public class Caculate {
public static void main(String args[]){
 int y[]=new int[4];
 char x;
 Scanner input=new Scanner(System.in);
 String s=input.nextLine();
 int l=s.length()-1;
 while(l>=0){
  x= s.charAt(l);
  if ((x-'a'>=0&&x-'a'<26) || (x-'A'>=0&&x-'A'<26)){
     y[0]++;
  }
  else
  if (x-'0'<10&&x-'0'>0){
     y[1]++;
  }
  else
  if (x==' '){
     y[2]++;
  }
  else
    y[3]++;
  l--;
 }
 for (int i=0;i<4;i++){
    System.out.println(y[i]);
 }
 }
}
```

从 System.in 中输入一个正整数，对其分解质因数。输出结果间用空格隔开。

示例输入一： 13　　示例输出一： 13

示例输入二： 12971322　　示例输出二： 2 3 3 7 13 7919

参考

参考答案：

```
import java.util.Scanner;
public class Factor {
    public static void main(String args[]) {
        int n = new Scanner(System.in).nextInt();
        for (int i = 2; i * i <= n; i++) {
            if (n % i == 0) {
                System.out.print(i + " ");
                n /= i--;
            }
        }
        System.out.print(n);
    }
}
```

5. 从《孙子算经》中记载了这样一道题目：今有雉兔同笼，上有三十五头，下有九十四足，问鸡兔各几只？

参考

参考答案：

JiTu.java

```
1  //假设鸡数量i，兔数量j
2  //因为头有35，所以鸡数量0<=i<=35,兔数量j=35-i
3  //因为共有足94，鸡有2条腿,兔有4条腿,满足2*i+4*j=94
4  public class JiTu {
5  public static void main(String[] args) {
6   int j = 0;
7   for (int i = 0; i <= 35; i++) {
8           j = 35 - i;
9           if (2 * i + 4 * j == 94){
10             System.out.println(i + " " + j);
11          }
12  }
13 }
14 }
15
```

Problems @ Javadoc Declaration Console

```
<terminated> JiTu [Java Application] C:\Program Files\Java\jre1.8.0_144\
23 12
```

6. 百钱百鸡，鸡翁一，值钱五，鸡母一，值钱三，鸡雏三，值钱一，百钱买白鸡，问翁、母、雏各几何

参考答案：

参考

BaiqianBaiji.java

```java
//公鸡x，母鸡y，小鸡z，总价100
public class BaiqianBaiji {
    public static void main (String [] args)
    {
        for (int x = 0; x <= 19; x++)
        {
            for (int y = 0; y <= 33; y++)
            {
                int z = 100 - x - y;
                if((x * 5 + y * 3 + z / 3 == 100 ) && z % 3 == 0)
                {
                    System.out.print("可买鸡翁只数:"+x+"\t");
                    System.out.print("鸡母只数:" + y+"\t");
                    System.out.println("鸡雏只数:" + z);
                }
            }
        }
    }
}
```

Problems @ Javadoc Declaration Console

<terminated> BaiqianBaiji [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (2020年9月22日

```
可买鸡翁只数:0    鸡母只数:25    鸡雏只数:75
可买鸡翁只数:4    鸡母只数:18    鸡雏只数:78
可买鸡翁只数:8    鸡母只数:11    鸡雏只数:81
可买鸡翁只数:12   鸡母只数:4     鸡雏只数:84
```

第三周课后作业 -1

作业

1. 使用 switch 把小写类型的 char 型转为大写。只转换 a, b, c, d, e. 其它的输出“ other ”。

2. 对学生成绩大于 60 分的，输出“合格”。低于 60 分的，输出“不合格”。

3. 根据用于指定月份，打印该月份所属的季节。

3,4,5 春季 6,7,8 夏季 9,10,11 秋季 12, 1, 2 冬季

4. 编写程序：从键盘上输入 2014 年的“ month ”和“ day ”，要求通过程序输出输入的日期为 2014 年的第几天。

第三周课后作业 -2

5. 打印 1~100 之间所有奇数的和

6. 打印 1~100 之间所有是 7 的倍数的整数的个数及

   总和（体会设置计数器的思想）

7. 输出所有的水仙花数，所谓水仙花数是指一个 3 位数，其各个位上数字立方和等于其本身。

   例如： 153 = 1*1*1 + 3*3*3 + 5*5*5

for 语句

第三周课后作业 -3

作业

8. Java 有哪些基本数据类型？描述其分类情况，编写一个简单的程序，各声明一个变量，初始化并输出其值。

9. 比较 break 与 continue 语句的区别。

10. 编写一个程序，把变量 n 的初始值设置为 1678 ，然后利用除法运算和取余运算把变量的每位数字都提出来并打印，输出结果为： n = 1 ，6 ，7 ，8

11. 编写 Java 程序，接受用户输入的 1~12 之间的整数，若不符合条件则重输入，利用 switch 语句输出对应月份的天数。

12. 编写 Java 程序计算一个整数之内的素数并输出。

13. Java 数组的特点是什么？如何创建和使用对象数组？

14. 编写 Java 程序实现：输入一组整数存放在数组中，比较并输出其中的最大值和最小值；再将数组元素从小到大排序并输出。

第三周课后作业 -4

作业

15. 定义一个 int 型的一维数组，包含 10 个元素，分别赋一些随机整数，然后求出所有元素的最大值，最小值，平均值，和值，并输出出来。

16. 有一对兔子，从 3 个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？（斐波那契数列）

第三周课后作业 -5

第 3 讲 ppt 上的所有程序代码编写（与调试）

作业发邮件到：
2230652597@qq.com