

综合复习和编程思维提升 - Day07

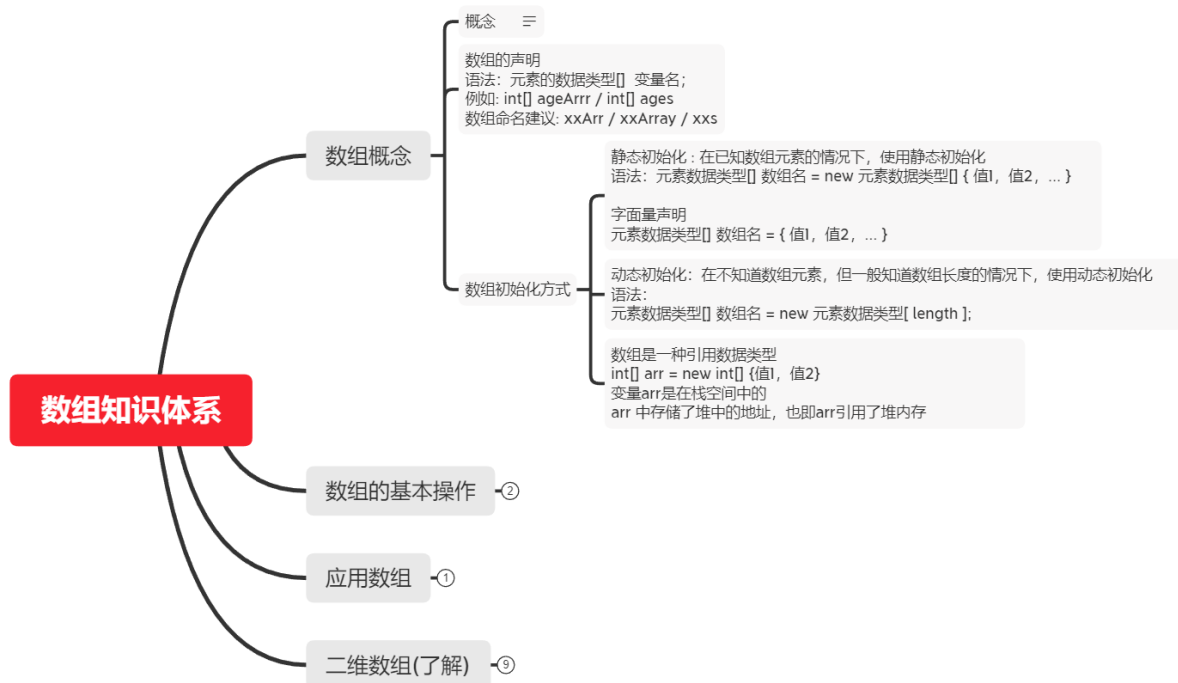
今日学习内容:

- XMind
- 编程思维提升
- 使用包管理和组织类

今日学习目标:

- 通过 XMind 能完整复述之前学习的知识，构建知识体系
- 能独立完成课堂实际案例分析，培养编程思维
- 理解 package 和 import关键字

1、XMind思维导图工具



2、编程思维提升

需求: 定义一个方法(getLeap),传入年份 (year), 判断传入的年份是否闰年, 返回true是闰年, false不是闰年。

写一个测试类, 测试下2016年是否闰年。

提示:

能被4整除且不能被100整除的为闰年 或者 能被400整除的是闰年。

```
1 public class LeapDemo {
2     /*
3     public static boolean getLeap(int year) {
4         boolean isLeap;
5         if(( year % 4 == 0 && year % 100 != 0) || ( year % 400 == 0 )) {
6             isLeap = true;
```

```

7         } else {
8             isLeap = false;
9         }
10        return isLeap;
11    }*/
12
13    /*
14    public static boolean getLeap(int year) {
15        boolean isLeap = false;
16        if( ( year % 4 == 0 && year % 100 != 0 ) || ( year % 400 == 0 ) ) {
17            isLeap = true;
18        }
19        return isLeap;
20    }*/
21
22    public static boolean getLeap(int year) {
23        boolean isLeap;
24        isLeap = (( year % 4 == 0 && year % 100 != 0 ) || ( year % 400 == 0 ))
? true : false;
25        return isLeap;
26    }
27
28    public static void main(String[] args) {
29
30    }
31 }

```

给定一个time = 4567s, 请问time中有?时?分?秒

```

1 public class TimeDemo {
2     public static void main(String[] args) {
3         int time = 4567;
4         int hour = time / 3600;
5         int minute = (time % 3600) / 60;
6         int second = time % 60;
7     }
8 }

```

使用switch编写程序, 给定一个学生成绩, 打印相应等级:

- (1) 90~100 优秀 [90,100] 90 91 9299 100
- (2) 80~89 良好 [80,89] 80 81 82 8389
- (3) 70~79 中等 [70,79]
- (4) 60~69 及格 [60,69]
- (5) 0~59 不及格 [0,59]

```

1 public class ScoreDemo {
2     public static void main(String[] args) {
3         int score = 78;
4         int level = score / 10;

```

```

5         switch( level ) {
6             case 10:
7                 case 9: System.out.println("优秀");break;
8                 case 8: System.out.println("良好");break;
9                 case 7: System.out.println("中等");break;
10                case 6: System.out.println("及格");break;
11                default: System.out.println("不及格");break;
12            }
13        }
14    }

```

给定某年某月某日，判断这一天是这一年的第几天？

```

1  public class DaysDemo {
2      public static boolean getLeap(int year) {
3          boolean isLeap;
4          isLeap = (( year % 4 == 0 && year % 100 != 0 ) || ( year % 400 == 0 ))
? true : false;
5          return isLeap;
6      }
7      public static int getDays(int year,int month) {
8          int days = 0;
9          switch(month) {
10             case 1:
11             case 3:
12             case 5:
13             case 7:
14             case 8:
15             case 10:
16             case 12: days = 31;break;
17             case 2: {
18                 boolean r = DaysDemo.getLeap(year);
19                 days = r ? 29 : 28;
20                 break;
21             }
22             default: days = 30;break;
23         }
24         return days;
25     }
26     public static void main(String[] args) {
27         // 给定某年某月某日，这一天是这一年的第几天？
28         int year = 2022;
29         int month = 5;
30         int day = 30;
31         // step 1: 定义一个变量存储累加的天数 sum
32         int sum = 0;
33         // step 2: 把month表示的月份之前的所有月份的天数累加到sum中
34         /*
35         sum += 1月份的天数
36         sum += 2月份的天数
37         sum += 3月份的天数
38         */
39         for(int m = 1;m < month;m++) {
40             sum += DaysDemo.getDays(year,m);
41         }

```

```

42         // step 3: 把day累加到sum中
43         sum += day;
44
45         // step 4: 打印sum
46         System.out.println(sum);
47
48     }
49 }

```

给定两个数组，写一个方法判断两个数组是否相等。

```

1  // 两个数组相等
2  1> 数组元素个数必须一样
3  2> 对应索引位置的元素必须相等
4  arr1 = {1,2,3}
5  arr2 = {1,2,3}
6
7  public static boolean equal(int[] arr1,int[] arr2) {
8      // step 1: 先判断数组长度是否相等
9      if( arr1.length == arr2.length ) {
10         // step 2: 取出每个数组对应索引位置的元素比较，如果不相等直接返回false,如果元
            素相等，继续取下一个元素，直到元素都取出来比较结束
11         int item1;
12         int item2;
13         for(int i = 0;i < arr1.length;i++) {
14             item1 = arr1[i];
15             item2 = arr2[i];
16             if( item1 != item2 ) {
17                 return false;
18             }
19         }
20         return true;
21     } else {
22         return false;
23     }
24 }

```

定义一个正整数的数组arr，删除arr数组中指定索引的元素，删除元素后面的元素需要往前移动，并且最后的索引位置用-1补上。如下图所示：



```

1 public class Demo {
2     public static void main(String[] args) {
3         int[] arr = {1,2,3,4,5};
4         // step 1: 定义一个index,存储要删除的位置
5         int index = 1;
6         // step 2:从index位置开始的下一个位置后续元素依次向前移动
7         for(int i = index;i < arr.length - 1;i++) {
8             arr[i] = arr[i+1];
9         }
10        // step 3:最后的索引位置用-1补上
11        arr[arr.length - 1] = -1;
12    }
13 }

```

3、包

3.1 为类定义包-package (会用)

在开发中存在几百个Java文件，为了更好的管理多个Java文件，我们可以使用package(包)来组织管理，类似于磁盘的文件夹一样(**包可以被看成目录**)，定义语法为：

```

1 语法格式：
2  package 包名.子包名.子子包；
3
4  例如：
5  package test.demo;

```

注意：

必须把该语句作为Java文件中第一行代码（所有代码之前）

包名使用全**小写字母**组成，命名符合标识符规则

父包和子包之间使用点符号（.）分割

因为Java的安全机制，自定义包名**不允许使用**java单词作为包名。

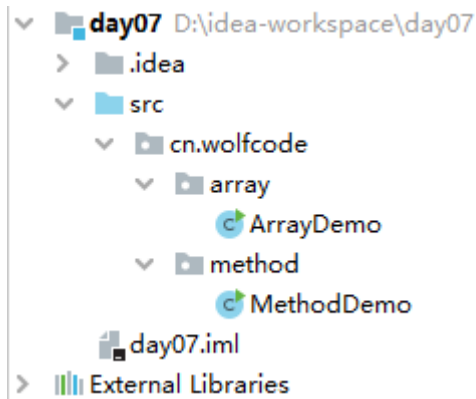
企业中起名规范（后续项目期）：

```

1 package 企业域名倒写.模块名.组件名；
2
3 如 package cn.wolfcode.array;

```

分包效果如下图：



创建出来的类，代码如下：

```
1 package cn.wolfcode.array; //当前类所在的包
2
3 public class ArrayDemo {
4     public static boolean equalArr(int[] arr1, int[] arr2) {
5         if (arr1.length == arr2.length) {
6             for (int i = 0; i < arr1.length; i++) {
7                 int item1 = arr1[i];
8                 int item2 = arr2[i];
9                 if (item1 != item2) {
10                     return false;
11                 }
12             }
13             return true;
14         } else {
15             return false;
16         }
17     }
18 }
```

此时，类的名称：

简单名称：ArrayDemo

全限定名：包名.简单名称，即cn.wolfcode.array.ArrayDemo

3.2 组织导入类-import（掌握）

在MethodDemo代码中使用ArrayDemo类中的equalArr方法比较两个数组是否相等：

```
1 package cn.wolfcode.method;
2
3 public class MethodDemo {
4     public static void main(String[] args) {
5         int[] arr1 = {1,2,3};
6         int[] arr2 = {1,2,3,4};
7         boolean r = cn.wolfcode.array.ArrayDemo.equalArr(arr1, arr2);
8         System.out.println(r);
9     }
10 }
```

注意：

ArrayDemo类在cn.wolfcode.array包中，不在cn.wolfcode.method包中,使用时必须使用全限定名称——cn.wolfcode.array.ArrayDemo 去调用方法。

每次使用equalArr方法，都必须使用ArrayDemo 类的全限定名来调用方法，确实很累，所以此时就该import关键字出场了。

语法:

```
1 | import 类的全限定名; // 表示只导入一个类。  
2 | import 包名.子包名.*; // 表示会引入该包下的所有在当前文件中使用到的类
```

小结：对package定义包和import导入某个类，会用就可以了，不深究。