

课程回顾

1 嵌套if结构如何return

扩展：String的功能判断两个字符是否相等

- 1 `equals()`:区分大小写判断两个字符串相等
- 2 `equalsIgnoreCase()`:区分大小写判断两个字符串相等

2 循环结构

循环概念：重复的做同一件事情

while循环：

流程执行特点：基于先循环条件判断，条件成立，才会执行循环操作。如果条件一次都不成立，循环操作也一次都不会执行。

循环语法：

```
1  循环变量；
2  while(循环条件){
3      循环操作；
4
5      修改循环变量的值；
6  }
```

3 debug方式

课后第六题参考代码

```
1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/11
6   * @Description: 课后作业第六题：录入5门课成绩并求平均分
7   * @Version: 1.0
8   */
9  public class T6 {
10     public static void main(String[] args) {
11         Scanner input=new Scanner(System.in);
12         //1.输入学生姓名，不是循环操作
13         System.out.println("请输入学生姓名");
14         String studentName = input.nextLine();
15
16         //2.循环接收用户输入5门课成绩，并求和
17         int count=0;
18         double sum=0;//保存总分
19         String choose="yes";//保存是否继续输入的信号
```

```

20         while(count<3&&sum<=300&&choose.equals("yes")){
21             System.out.println("请输入第"+(count+1)+"门课程的成绩: ");
22             int score = input.nextInt();//保存学生录入的成绩
23
24             //求和
25             sum+=score;//sum=sum+score;
26             //修改循环变量的
27             count++;
28             //给出停止程序的选择
29             System.out.println("输入yes表示继续录入"+studentName+"同学的成绩，输入其他停止录入: ");
30             choose = input.next();//选择是否继续录入成绩,choose作用域不够
31         }
32         // 输出平均分，不是循环操作
33         System.out.println(studentName+"的五门课成绩平均分是: "+sum/count);
34     }
35 }

```

课程目标

1 do-while循环 ==== 掌握

2 for循环 ==== 掌握

3 嵌套循环 ===== 难点

4 break和continue === 掌握

课程实施

1 do-while循环

1-1 语法

```

1  do{
2      //循环操作
3  }while(循环条件);
4
5  与while语句最大的区别，执行流程发生变化：
6  先做循环操作，再判断循环条件。如果条件成立，继续执行循环操作。
7  无论循环条件是否满足，循环操作至少会执行一次。
8
9  do-while:先做 后判断
10 while: 先判断，再执行

```

1-2 课堂案例

while实现看书上机操作题目

```

1  import java.util.Scanner;
2
3  /**
4   * @Author: lc

```

```

5  * @Date: 2022/3/11
6  * @Description: while案例
7  * 三要素:
8  * 循环操作: 上午阅读教材, 学习理论部分。下午上机编程, 掌握代码部分
9  * 修改循环变量值: 事情是否合格
10 *
11 * 循环条件: 不合格, 一直重复读书+编码
12 *
13 * 循环变量: y合格    n不合格    String
14 * @Version: 1.0
15 */
16 public class Demo1 {
17     public static void main(String[] args) {
18         Scanner input = new Scanner(System.in);
19         //思考: 冗余代码    流程有没有优化空间?
20         System.out.println("合格了吗? (y/n)");
21         //循环变量
22         String answer=input.next();
23         while(answer.equalsIgnoreCase("n")){
24
25             System.out.println("上午阅读教材! ");
26             System.out.println("下午上机编程! ");
27
28             System.out.println("合格了吗? (y/n)");
29             answer = input.next();
30         }
31         System.out.println("完成学习任务! ");
32     }
33 }
34

```

do-while实现

```

1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/11
6   * @Description: while案例
7   * 三要素:
8   * 循环操作: 上午阅读教材, 学习理论部分。下午上机编程, 掌握代码部分
9   * 修改循环变量值: 事情是否合格
10  *
11  * 循环条件: 不合格, 一直重复读书+编码
12  *
13  * 循环变量: y合格    n不合格    String
14  * @Version: 1.0
15  */
16 public class Demo1 {
17     public static void main(String[] args) {
18         Scanner input = new Scanner(System.in);
19         String answer;//未知答案
20         do{
21             System.out.println("上午阅读教材! ");
22             System.out.println("下午上机编程! ");
23
24             System.out.println("合格了吗? (y/n)");

```

```

25         answer = input.next();
26     }while(answer.equalsIgnoreCase("n"));
27     System.out.println("完成学习任务! ");
28 }
29 }
30

```

1-3 学生练习

需求：取款

- 1.取款金额必须整百
- 2.取款金额必须小于等于卡内余额
- 3.循环保证取款操作时，金额是合法的

```

1  do{
2      sout("输入取款金额: ")
3      int money=input.nextInt();
4
5  }while(取款金额%100!=0 || 取款金额>卡内余额);
6
7  卡内余额-=取款金额;
8  sout()

```

参考代码

```

1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/11
6   * @Description: 控制取款金额的合法性
7   * @Version: 1.0
8   */
9  public class Demo2 {
10      public static void main(String[] args) {
11          Scanner input = new Scanner(System.in);
12          //定义变量，表达卡内余额
13          double balance=994.88;
14          //布尔类型，true false
15          boolean isOk;//isOk表示用户输入的取款金额是合法的
16          int money;//取款金额
17          do {
18              isOk=true;//重置记号
19              System.out.print("请输入取款金额: ");
20              money = input.nextInt();
21              //不是整百
22              if(money % 100 != 0) {
23                  System.out.println("取款金额必须是整百");
24                  isOk = false;
25              }
26              //取款金额超标
27              if(balance <= money) {

```

```

28         System.out.println("取款金额必须小于或等于卡内余额! 卡内余额不足");
29         isok = false;
30     }
31     }while(!isok);
32     balance-=money;//取款
33     System.out.println("取款成功, 卡内余额是: "+balance);
34     //转账
35 }
36 }

```

2 for循环

引入案例: 1-10所有能被5整除的整数

方案一: while循环

方案二: do-while循环

课堂演示案例

while实现

```

1  /**
2   * @Author: lc
3   * @Date: 2022/3/11
4   * @Description: PACKAGE_NAME
5   * @Version: 1.0
6   */
7  public class Demo3 {
8      public static void main(String[] args) {
9          /*
10           while存在的问题:
11           1.循环三要素, 书写过于分散, 程序员很容易遗漏: i++(死循环)
12           2.循环变量写在什么位置合适? while语法缺点
13
14           解决方案: for语法, 执行流程与while一模一样。for语法紧凑
15           for适用场景: 循环次数明确的循环
16           举例: 打印一万次我爱你  获取5门课程的成绩  数组  集合
17           */
18           int i=1;
19           while(i<=10){
20               if (i%5==0) {
21                   System.out.println(i);
22               }
23               i++;
24           }
25       }
26   }
27

```

for实现

```

1  /**
2   * @Author: lc
3   * @Date: 2022/3/11
4   * @Description: PACKAGE_NAME

```

```

5  * @Version: 1.0
6  */
7  public class Demo3 {
8      public static void main(String[] args) {
9          //while案例转换for应用场景
10         /*for(循环变量定义;循环条件;循环变量++/--){
11             //循环操作
12         }*/
13         for(int i=1;i<=10;i++){
14             //条理化很清楚
15             if (i%5==0) {
16                 System.out.println(i);
17             }
18         }
19     }
20 }
21

```

for语法

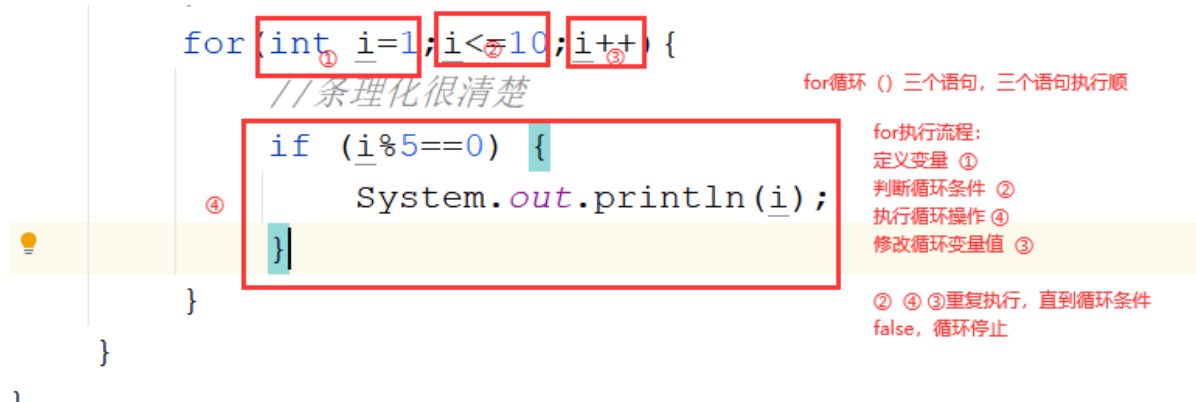
```

1  for(定义循环变量;循环条件;修改循环变量表达式){
2      //循环操作
3  }

```

for循环执行流程

for循环其实就是while循环，执行流程与while循环一模一样。如下图所示



学生练习

需求：找出所有水仙花数

水仙花：三位数。个位 十位 百位各个数字立方和=数字本身

```

1  int num=234;
2  if(2*2*2+3*3*3+4*4*4=234){
3
4  }
5
6  for(100,i<1000;i++){
7      gewei=.
8      if(2*2*2+3*3*3+4*4*4=234){
9          sout(i)
10     }
11 }

```

参考代码

```

1  /**
2   * @Author: lc
3   * @Date: 2022/3/11
4   * @Description: PACKAGE_NAME
5   * @Version: 1.0
6   */
7  public class Demo4 {
8      public static void main(String[] args) {
9          for(int num=100;num<1000;num++){
10             //个位 十位 百位
11             //123/100=1...23
12             int baiwei=num/100;
13             //23/10==2...3
14             int shiwei=num%100/10;
15             int gewei=num%100%10;
16             if (Math.pow(baiwei,3)+Math.pow(shiwei, 3)+Math.pow(gewei,
17 3)==num) { //Math.pow(a,b)表示a的b次幂方
18                 System.out.println(num); //输出100-1000之间所有的整数
19             }
20         }
21     }
22 }

```

3 流程控制关键字

3-1 break

break作用：用在循环中，用来终止循环，break后面的代码将不再继续执行

3-2 continue

continue作用：继续。终止本次循环，使循环进入下一次

3-4 break continue和return的区别

break:终止循环

```
if(1==1) {  
    System.out.println("aa");  
    break;  
}  
System.out.println("连接数据库，  
System.out.println("程序执行结束
```

break只能用在switch和循环的里面

Break outside switch or loop

continue: 结束一次循环，循环还得进入下一次

```
if(1==1) {  
    System.out.println("aa");  
    //break;  
    continue;  
}  
System.out.println("连接数据库，保存数据");
```

continue 只能用在循环中

Continue outside of loop

return: 停止程序，让程序不再继续执行。理解: jvm停止

return可以使用在if或循环或switch，但是，return一旦出现，程序就停止啦。请谨慎。

学生练习

需求:

- 1.Scanner提示用户输入5个学生成绩。当用户输入的成绩>120分，提示重新输入本次成绩
- 2.计算平均分

需求分析:

```
1  请输入第一门课成绩:  
2  999  
3  成绩只能是0-120之间，输入数据不合法  
4  请输入第一门课成绩:  
5  456  
6  成绩只能是0-120之间，输入数据不合法  
7  请输入第一门课成绩:  
8  90  
9  第二门  
10 第三门  
11  
12 伪代码:  
13 for(int i=1;i<=5;i++){  
14     请输入第一门课成绩:  
15     成绩=用户输入的值;
```



```

16
17     if(成绩>120){
18         sout(成绩只能是0-120之间, 输入数据不合法);
19         continue;
20     }
21     sum+=成绩
22 }
23 while()
24

```

参考代码

```

1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/11
6   * @Description: continue使用场景
7   * @Version: 1.0
8   */
9  public class Demo7 {
10     public static void main(String[] args) {
11         Scanner input = new Scanner(System.in);
12         double sum=0;
13         for(int i=1;i<=3;i++){
14             System.out.println("请输入第"+i+"门课程的成绩: ");
15             int score = input.nextInt();
16             //是否求和, 判断分输入是否对, 合法
17             /*if (score<=120 && score>=0) {
18                 sum+=score;
19
20             }else{
21                 System.out.println("输入的成绩不合法, 请输入0-120之间数据。");
22                 i--;
23             }*/
24             //排除不合法, 剩下都是合法
25             if(score>120||score<0){
26                 System.out.println("输入的成绩不合法, 请输入0-120之间数据。");
27                 i--;
28                 //break;//遇到不合法, 停止输入
29                 //continue;//循环进入下一次, 后面代码跳过不执行
30             }
31             sum+=score;
32         }
33         System.out.println("五门课程平均分: "+sum/3); //sum/0 除数不能为零
34     }
35 }

```

4 嵌套循环

4-1 嵌套概念

```
1  while(){
2      for(){
3
4      }
5  }
6
7  for(){
8      for(){
9
10     }
11 }
12
13 do{
14     while(){
15
16     }
17     for(){
18
19     }
20 }while();
21 嵌套层级不要超过3层
```

4-2 嵌套循环执行流程

流程图绘制，理解执行流程

外层循环执行一次，内层循环执行一轮（即：外层循环变量变化一次，内层循环变量从最小值到最大值变化一轮）

参考如下案例分析：

```
// System.out.println("* * * * *"); 简化只输出一颗*
for (int i=0; i<5; i++) { //修改变量名时，所有引用的位置同时修改：shift+(F6+
    /*
    当i=0,j会出现哪些值的赋值?
    0*0 0*1 0*2 0*3 0*4
    当i=1,j会出现哪些值的赋值?
    1*0 1*1 1*2 1*3 1*4
    当i=4,会出现哪些值的赋值?
    4*0 4*1 4*2 4*3 4*4
    */
    for (int j=0; j<5; j++) {
        System.out.print(i+"*"+j+", "); //
    }
    //换行
    System.out.println(); //换行
    //System.out.print("\n");
}
/*System.out.println("* * * * *");
System.out.println("* * * * *");
```

4-3 案例

矩形图案

```
1 * * * * *
2 * * * * *
3 * * * * *
4 * * * * *
5 * * * * *
6 * * * * *
7 * * * * *
```

分析:

参考代码 方案一：顺序结构

```
1 /**
2  * @Author: lc
3  * @Date: 2022/3/11
4  * @Description: 打印矩形图案: 打印5行, 每行5颗*
5  * @Version: 1.0
6  */
7 public class Demo8 {
8     public static void main(String[] args) {
9         System.out.println("* * * * *");简化只输出一颗*
10        System.out.println("* * * * *");
11        System.out.println("* * * * *");
12        System.out.println("* * * * *");
13        System.out.println("* * * * *");
14    }
15 }
```

参考代码 方案二：循环结构while do-while for

```
1 /**
2  * @Author: lc
3  * @Date: 2022/3/11
4  * @Description: 打印矩形图案: 打印5行, 每行5颗*
5  * @Version: 1.0
6  */
7 public class Demo8 {
8     public static void main(String[] args) {
9         // System.out.println("* * * * *");简化只输出一颗*
10        for (int i=0;i<5;i++) { //修改变量名时, 所有引用的位置同时修改: shift+
(F6+FN)
11            /*
12             当i=0,j会出现哪些值的赋值?
13             0*0 0*1 0*2 0*3 0*4
14             当i=1,j会出现哪些值的赋值?
15             1*0 1*1 1*2 1*3 1*4
16             当i=4,会出现哪些值的赋值?
17             4*0 4*1 4*2 4*3 4*4
18            */
19            for (int j=0;j<5;j++) {
20                System.out.print(i+"*"+j+",");//
21            }
```

```

22         //换行
23         System.out.println();//换行
24         //System.out.print("\n");
25     }
26     /*System.out.println("* * * * *");
27     System.out.println("* * * * *");
28     System.out.println("* * * * *");
29     System.out.println("* * * * *");*/
30 }
31 }

```

九九乘法表

```

1  1*1=1
2  2*1=2 2*2=4
3  3*1=3 3*2=6 3*3=9
4  . . . . .
5  9*1=9 9*2=18. . . . . 9*9=81

```

分析

```

1  for(int i=1;i<10;i++){
2      /* 当i=1,输出第一行的数据 j循环几次? 1 j=1 j<=1
3      * 当i=2 输出第二行, j循环几次? 2次 j=1 j<=2
4      * 当i=3 输出第三行, j循环三次? 3次 j=1 j<=3
5      . . . j<=i
6      */
7      for(int j=1;j<=i;j++){
8          sout(i+"*"+j+"="+i*j);
9      }
10     sout();//换行
11 }

```

参考代码

```

1  public class Exercise04 {
2      public static void main(String[] args) {
3          for (int i = 1; i < 10; i++) {
4              for (int j = 1; j <= i; j++) {
5                  System.out.print(i+"*"+j+"="+i*j+"\t");
6              }
7              System.out.println();
8          }
9      }
10 }

```

等腰三角形

当i=1时，打印第一行图案

```
for(int k=1;k<=4;k++){ //控制空格
    cout<<"\t";
}
for(int j=1;j<=1;j++){ //控制*
    cout<<"*";
}
```

换行

当i=2

一个循环控制输出空格数量

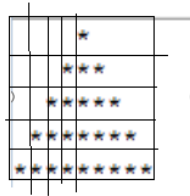
```
for(int k=1;k<=3;k++){
    cout<<"\t";
}
for(int j=1;j<=3;j++){
    cout<<"*";
}
```

换行

当i=3

一个循环控制输出空格数量

```
for(int k=1;k<=2;k++){
    cout<<"\t";
}
for(int j=1;j<=5;j++){
```



输出五行图形

```
for(int i=1;i<=5;i++){
    for(int k=1;k<=5-i;k++){
        cout<<"\t";
    }
    for(int j=1;j<=2*i-1;j++){
        cout<<"*";
    }
    cout<<"\n";
}
```



分析i和k之间的算法关系

i=1	k=1	2	3	4
i=2	k=1	2	3	
i=3	k=1	2		
i=4	k=1			
i=5	k=0			

i+k=5
k=5-i

分析i和j之间的算法关系

i=1	j=1
i=2	j=1 2 3
i=3	j=1 2 3 4 5
i=4	j=1 2 3 4 5 6 7
i=5	j=1 2 3 4 5 6 7 8 9

i j==最大值算法

2*i-1=j

参考代码

```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/11
4   * @Description: PACKAGE_NAME
5   * @Version: 1.0
6   */
7  public class Demo9 {
8      public static void main(String[] args) {
9          System.out.println("请输入打印图形的总行数: ");
10         for(int i=1;i<=5;i++){//控制打印图形总行数
11             for(int k=1;k<=5-i;k++){ //控制空格
12                 System.out.print("\t");
13             }
14             for(int j=1;j<=2*i-1;j++){ //控制*
15                 System.out.print("*\t");
16             }
17             System.out.println();
18         }
19     }
20 }
```

课程总结

1 do-while

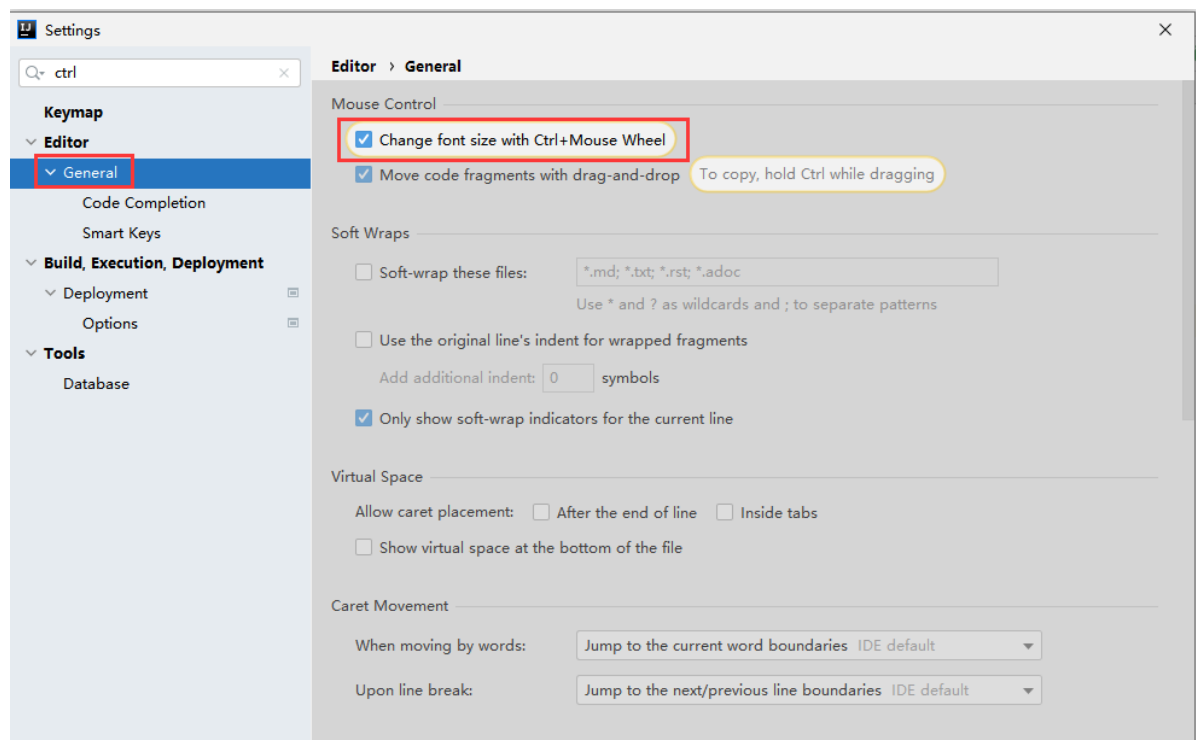
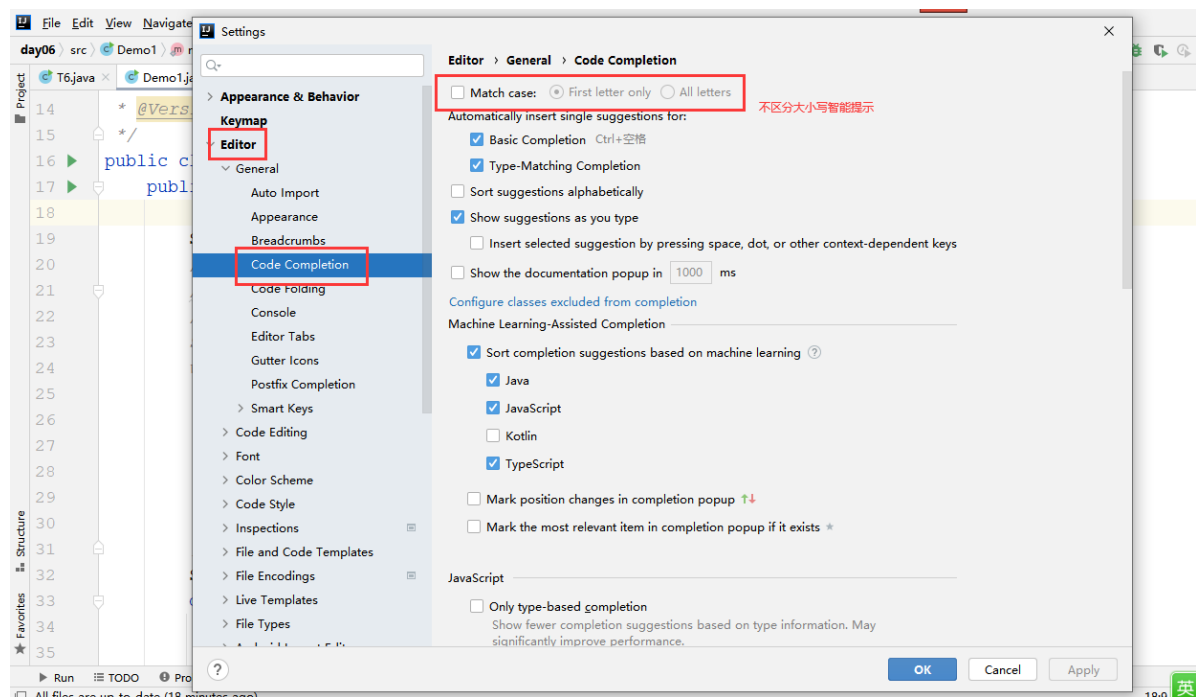
2 for适用场景和语法执行

3 break continue return

4 for(){for()}嵌套（嵌套循环执行流程：）

补充

1 idea常用配置



预习安排

数组：定义 循环数组 下标控制数据存取