1、构造器的特征

1、它具有与类相同的名称

2、不声明返回值类型。

3、不能被 static 、 final 、 synchronized 、 abstract 、 native 修饰，不能有return语句返回值

2、构造器的作用

1、创建对象

```
Animal animal = new Animal();
```

2、给对象进行初始化

```
class
 Animal {
 private
 int legs;
 //
构造器
 public
 Animal() {
```

```
9   legs
10   = 4;
11   }
12   public
13   void setLegs( int i )
14   legs
15   = i;
16   }
17   public
18   int getLegs() {
19   return
20   legs;
21   }
22  }
23  public class Zoo {
24   public static void main(String[] args) {
25   Animal animal1 = new Animal(); // 初始化
26   System.out.println(animal1.getLegs()); //4
27   }
28  }
29
30
31   Person p = new Person (“Peter”,15) ;
```

3、构造器分类

- 隐式无参构造器（系统 默认 提供）
- 显式定义一个或多个构造器（无参、有参）

4、说明

1、Java 语言中，每个类都至少有一个构造器

2、默认构造器的修饰符与所属类的修饰符一致

3、定义构造器的格式：权限修饰符 类名（形参列表）{ }

4、如果没有显式的定义类的构造器的话，则系统默认提供一个空参的构造器

5、一旦显式定义了 构造器，则系统不再提供默认构造器

6、一个类中定义的多个构造器，彼此构成重载

5、构造器重载

- 构造器一般用来创建对象的同时初始化对象
- 构造器重载使得对象的创建更加灵活，方便创建各种不同的对象。
- 构造器重载，参数列表必须不同

```
public class Person{
 String name;
 int age;
 Data data;

 public Person(String name, int age, Data data) {
 this.name = name;
 this.age = age;
 this.data = data;
  }

  public Person(String name, int age) {
  this.name = name;
  this.age = age;
  }

  public Person(String name, Data data) {
  this.name = name;
  this.data = data;
  }
}
```

6、习题练习

1、(1)定义Student类有4个属性:

String name;

int age;

String school;

String major;

(2)定义Student类的3个构造器:

- 第一 个构造器 Student(String n, int a) 设置类的 name 和 age 属性;

- 第二 个构造器 Student(String n, int a, String s) 设置类的 name, age 和 school 属性;
- 第三 个构造器 Student(String n, int a, String s, String m) 设置类的 name, age ,school 和 major 属性;

(3)在main方法中分别调用不同 的构造器创建的对象，并输出其属性值。

```
class Student {
 private String name ;
 private int age ;
 private String school;
 private String major;

 // 设置类的name和age属性;
 public Student(String name, int age) {
 this.name = name;
  this.age = age;
  }

  // 设置类的name,age和school属性
  public Student(String name, int age, String school) {
  this.name = name;
  this.age = age;
  this.school = school;
  }

  // 设置类的name,age和school以及major属性
  public Student(String name, int age, String school, String major) {
```

```
 this.name = name;
 this.age = age;
 this.school = school;
 this.major = major;
 }

 public String getName() {
 return name;
 }

 public int getAge() {
 return age;
 }

 public String getSchool() {
 return school;
 }

 public String getMajor() {
 return major;
 }
}

public class StudentTest {
 public static void main(String[] args) {
 Student s1 = new Student("灰灰",12);
 System.out.println("name: " + s1.getName() + " ;age: " + s1.getAge());
 System.out.println();
 Student s2= new Student("灰灰",12, "天工大");
 System.out.println("name: " + s2.getName() + " ;age: " + s2.getAge() +
" ;school: " + s2.getSchool());
 System.out.println();
 Student s3 = new Student("花花",18, "天工大","机械设计");
 System.out.println("name: " + s1.getName() + " ;age: " + s1.getAge() +
" ;school: "
 + s1.getSchool() + " ;major: " + s3.getMajor());
 System.out.println();
 }
}
```

2、编写两个类 TriAngle和TriAngleTest 其中 TriAngle 类中声明私有的底
边长base和高height同时声明公共方法访问私有变量 。此外提供类
必要的构造器。另一个类中使用这些公共方法计算三角形的面积。

```
public class TriAngle {
 private double base; // 底边长
 private double height; // 高

 public double getBase() {
 return base;
 }

 public void setBase(double base) {
  this.base = base;
  }

  public double getHeight() {
  return height;
  }

  public void setHeight(double height) {
  this.height = height;
  }

  // 计算面积
  public double area(double b, double h) {
  return (b * h) / 2.0;
  }
 }
 public class TriAngleTest {
  public static void main(String[] args) {
  TriAngle t = new TriAngle();
  System.out.println("base :" + t.getBase() + " ;height: " +
t.getHeight());
  System.out.println();
  t.setBase(1);
  t.setHeight(1.5);
  System.out.println("base :" + t.getBase() + " ;height: " +
t.getHeight());
  System.out.println("area: " + t.area(t.getBase(), t.getHeight()));
```

```
35    }
36  }
```

7、总结

不同位置属性赋值的先后顺序：

- 赋值的位置
  1. 默认初始化
  2. 显式初始化
  3. 构造器中初始化
  4. 通过“对象 属性“或“对象 方法”的方式赋值

- 赋值的先后顺序：

1->2->3->4