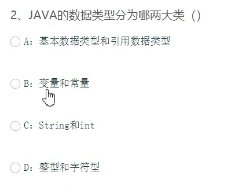
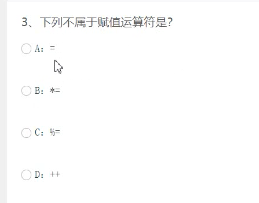


答案：C

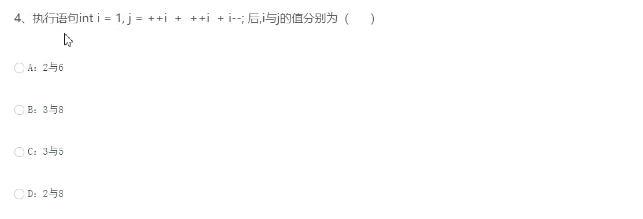


答案：A

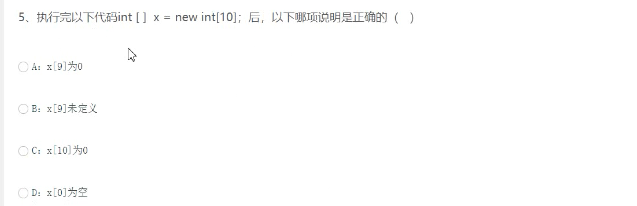


答案：D

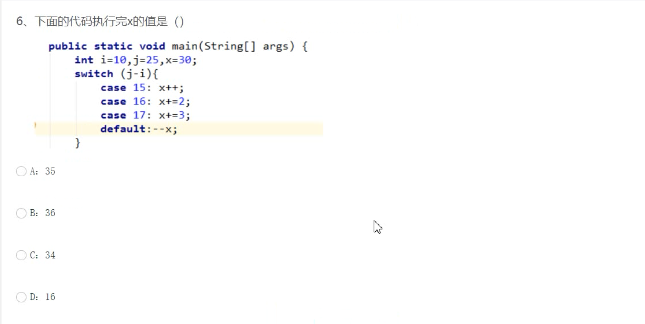
int a = 100; 扩展运算符 += a+=10 a = a + 10; short s = 10; s = (short)(s + 10);//error s += 10;//ok

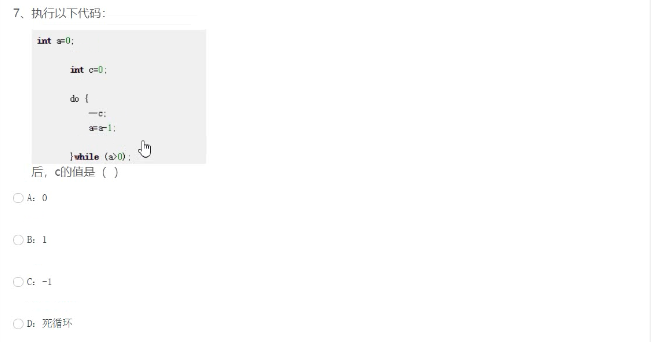


答案：D



答案：A

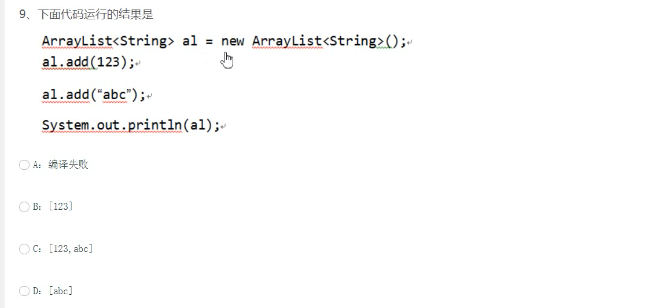
 答案：A

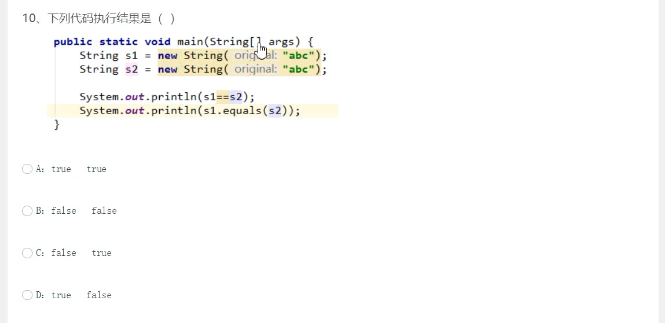


答案：C

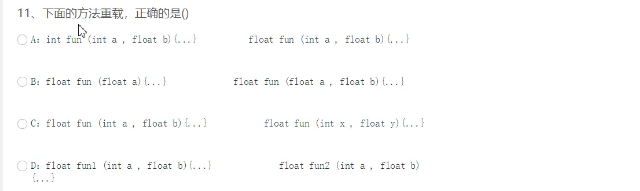


答案：B

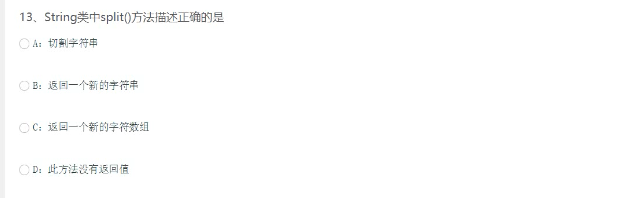
答案：A



答案：C 只要new就会创建一个新的对象就会在堆内存中开辟空间

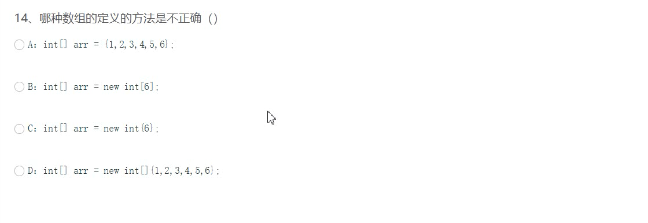


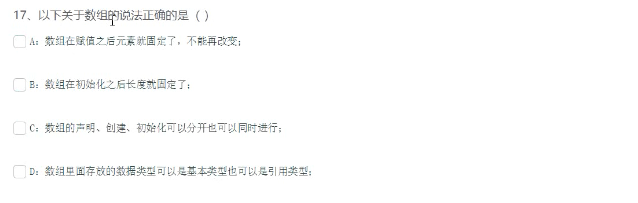
答案：B

 答案：D 答案：A

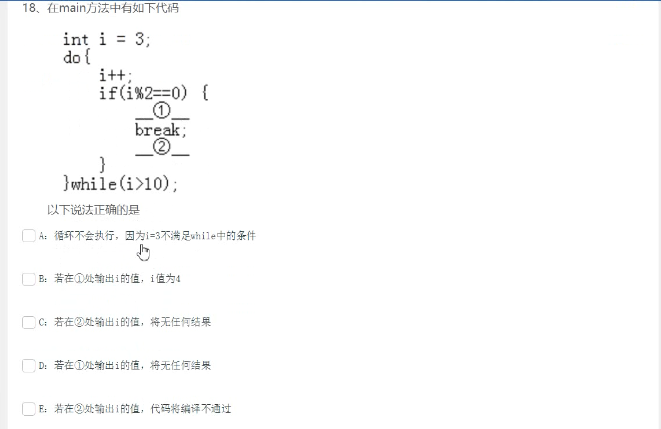
String str = “a,b,c”;

String[] str = str.split(“,”);

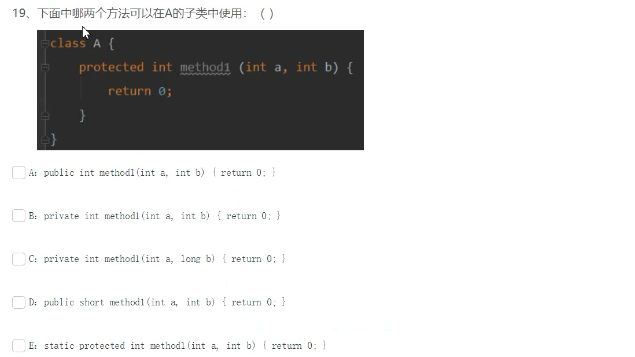
 答案：C 答案：B

 答案：BC 

答案：BCD



答案：BE



答案：AB



答案：AC

构造方法和类名保持一致，包括大小写，构造方法可以重载

有无参构造方法也有有参构造方法，编译器会默认提供一个无参构造方法

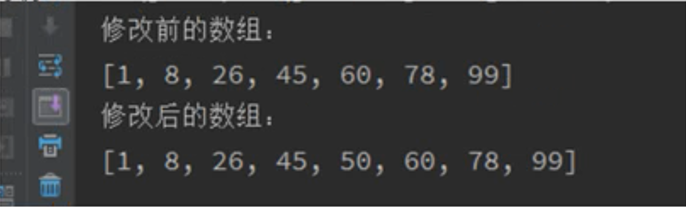
当我们自己定义类构造方法编译器不再默认提供

在创建对象时构造方法被调用执行，编译器会根据参数决定调用哪个构造方法

## 代码题 :

21、有一个已经排好序的数组[1, 8, 26, 45, 60, 78, 99]，现在键盘录入一个0-100之间包含0和100（范围无需代码判断）的数字，将该数字按从小到大顺序插入到数组（如输入数字50, 那么这个数组的结果为：[1, 8, 26, 45, 50, 60, 78, 99]）。并在控制台中打印插入前和插入后的数组。

示例如下所示 :



答案：

**int**[] arr = {1, 8, 26, 45, 60, 78, 99}; //比如插入50 比50小的放在50前边 比50大的放在50后边

//获取键盘录入功能

Scanner scanner = new Scanner(System.in);

System.out.println("请输入一个整数：");

//i就是我们键盘录入的数字

int i = scanner.nextInt(101); //0-100

int[] new\_arr= new int[arr.length+1]; //创建一个新数组 保存排好序的数组 新数组要确定50数据的索引

//遍历旧数组 取出值 （和i进行比较）

int index=0;

for (int i1 = 0; i1 < arr.length; i1++) {

if (arr[i1] <i){

new\_arr[i1]=arr[i1];

index++;//记录插入数据的索引

} else{//这是比50大的元素

new\_arr[i1+1]=arr[i1] ; 四个元素比20小

}

}

new\_arr[index]=i; //50放入新数组中的 4

for (int i1 = 0; i1 < new\_arr.length; i1++) {

System.out.println(new\_arr[i1]);

}

22. 定义测试类Test , 完成以下需求 :

① 定义public static ArrayList<Phone> filter(ArrayList<Phone> list,double price) {...}方法:

要求：遍历list集合，将list中价格大于参数price的元素存入到另一个ArrayList<Phone> 中并返回

② 在main方法内完成以下要求:

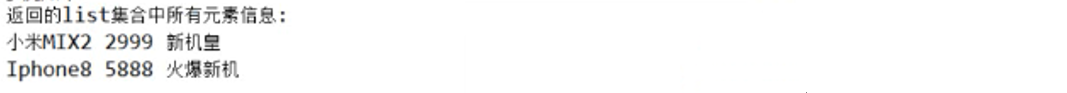
// a.根据以下内容创建并初始化3个Phone对象

// {"小米MIX2",2999,"新机皇"}

// {"Iphone8", 5888,"火爆新机"}

// {"VIVO X9s",1998,"火爆新机"}

// b.创建一个ArrayList<Phone> list\_phone，将上面的3个Phone对象添加到list\_phone中，调用filter方法传入list\_phone和2000，根据返回的list集合输出所有元素信息 , 示例如下所示 :



public class Phone {

private String brand;

private int price;

private String style;

public Phone() {

}

public String getBrand() {

return brand;

}

public void setBrand(String brand) {

this.brand = brand;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

public String getStyle() {

return style;

}

public void setStyle(String style) {

this.style = style;

}

public Phone(String brand, int price, String style) {

this.brand = brand;

this.price = price;

this.style = style;

}

@Override

public String toString() {

return "Phone{" +

"brand='" + brand + '\'' +

", price=" + price +

", style='" + style + '\'' +

'}';

}

}

**public class** Test2 {  
 **public static void** main(String[] args) {  
 Phone phone = **new** Phone(**"小米MIX2"**, 2999, **"新机皇"**);  
 Phone phone1 = **new** Phone(**"Iphone8"**, 5888,**"火爆新机"**);  
 Phone phone2 = **new** Phone(**"VIVO X9s"**,1998,**"火爆新机"**);  
 ArrayList<Phone> list\_phone = **new** ArrayList<>();  
 list\_phone.add(phone);  
 list\_phone.add(phone1);  
 list\_phone.add(phone2);  
 ArrayList<Phone> list = *filter*(list\_phone, 2000);  
 System.***out***.println(list);  
 }  
 **public static** ArrayList<Phone> filter(ArrayList<Phone> list, **double** price) {  
 ArrayList<Phone> newList = **new** ArrayList<>();  
 **for** (**int** i = 0; i < list.size(); i++) {  
 Phone phone = list.get(i);  
 **int** price1 = phone.getPrice();  
 **if** (price1>price) {  
 newList.add(phone);  
 }  
 }  
 return newList;  
 }  
}