1.B a. 包含了各种类和方法，用于开发java程序

c. 垃圾回收器负责管理内存中的对象，并释放不再被引用的对象占用的内存

d. 译器将Java源代码编译成字节码

2.C a. 字节(byte)的范围是 -128 到 127

b. 应该使用小写的 true

d. 少写了 “f“或”F“

3.A b. x[2]已经定义并初始化为0

c. 访问越界

d.java 虚拟机给数组元素赋默认值0

4.B a. 这是一个声明字符数组的语法

c.错误

d.混淆了创建字符串

# 二．

1.

数组a一维长度：3

数组a二维长度：2

数组d中的元素：

0 0 2

1 0 3 0

2 0 4 0 6

解释：创建了一个包含三个元素的数组，每个元素都是一个数组，通过嵌套循环为每个子数组的元素赋值，内层循环控制变量保证每个子数组内元素若下标为偶数， 则值依次增加2，若下标为奇数，则值为0，外层循环控制变量保证每个子数组首元素的值依次为0,1,2。

oneDimArr 是一个临时的变量名，用来表示数组 d 中的每一个一维子数组。d 是一个二维数组，所以在遍历的过程中，oneDimArr 会依次引用数组 d 中的每一个一维子数组。

这个循环结构能够逐个访问数组 d 中的一维子数组，然后在内层循环中再访问每个一维子数组中的元素。这样可以方便地处理二维数组中的元素。

2.

调用change1(s)后的输出：

6 2 3 4 7

调用change2(s)后的输出：

6 2 8 4 7

解释：由于change1传递了数组对象的堆内存地址给形参，导致赋值操作会修改原数组。

Change2在原数组被change1修改的前提下，再次修改，而s = new int[] { 5, 4, 3, 2, 1 }导致形参s指向一个新创建的数组，当方法调用结束后s会被销毁，实参s依旧指向最开始的共计被修改两次数组。

3.

callme a = 8

callme b = 11

Mydemo.a=8

Mydemo.b=11

obj1.a = 8

obj1.b = 11

解释：由于 a 和 b 是静态变量，它们在类的所有对象之间共享，obj.resetAB(4, 5)和 obj1.resetAB(2, 3)作用的都是同一个a和同一个b，不论是 类名**.**静态变量 还是

实例名**.**静态变量 访问的都是 java虚拟机方法区里同一个静态变量。

# 三．

public class Book{

  private String title;

  private int pageNUM;

  public String getTitle(){

   return title;

  }

  public void setTitle(String str){

    title=str;

  }

  public int getPageNum(){

    return pageNUM;

  }

  public void setPagNum(int a){

    if (a>=0)

    pageNUM=a;

  }

  public void infor(){

    System.out.println(" 书籍名："+title+" "+"页数："+pageNUM);

  }

}

class test{

  public static void main(String[] args) {

    Book book1=new Book();

    Book book2=new Book();

    book1.setTitle("金瓶梅");

    book1.setPagNum(100);

    book2.setTitle("java语言程序设计");

    book2.setPagNum(443);

    book1.infor();

    book2.infor();

  }

}

