1. 单项选择题：
2. A

①--/++放变量前面先做整句表达式后改变变量值，放后面先改变变量值后做整句表达式

②整数除法3/2=1

\*歧义：因为本题中并未说明声明x,y,z为double还是int，故B.3.5也是有可能的

但一般按照整数来理解

1. C
2. C

“+”从左到右运算，只有遇到不是同类型相加时才转义为字符串

1. B

在java中可有两种方式实现多线程，一种是继承Thread类，一种是实现Runnable接口

本题为使用Runnable接口的方法（程序开发中主流方法）：

①Runnable作为一个interface只有一个需要实现的函数public void run()

②Thread是Runnable的一个实现

③Thread的构造方法：public Thread(Runnable targer)

④Thread的成员函数start()：

开一个线程运行run()函数，且同一个实例化**只能调用一次**，否则运行报错；

线程不同于函数可同时运行而非前一个做完再做后一个

\*优势：避免点继承的局限，一个类可以继承多个接口

\*e.g.1：实现两个窗口同时卖10张票

public class test

{

public static void main(String[] args)

{

MyThread mt=new MyThread();

new Thread(mt).start();

new Thread(mt).start();

**//当MyThread为extends Thread时不能直接用两次mt.start()**

}

}

class MyThread extends Thread/implements Runnable

{

private int ticket=10;

public void run()

{

for(int i=0;i<20;i++)

{

if(this.ticket>0)

System.out.println("Ticket "+this.ticket--);

}

}

}

\*e.g.2：两个窗口同时各自独立卖10张票

public class test

{

public static void main(String[] args)

{

MyThread mt1=new MyThread();

MyThread mt2=new MyThread();

mt1.start();/new Thread(mt1).start();

mt2.start();/new Thread(mt2).start();

**//当MyThread为implements Runnable时不能用mt1.run();mt2.run();**

**//变成先1号窗卖完10张票再让2号窗卖10张，因为run()是函数不是线程**

}

}

class MyThread extends Thread/implements Runnable

1. C

①“int anar[]=new int[]{1,2,3};”效果与“int[] anar={1,2,3}”等同

②下标从0开始计数

1. B

\*歧义：goto为JAVA的保留关键字，意思是当前版本这个词不能通过编译

但按照单选题的理解只能选B

\*说明：为了解决不能用goto的问题，JAVA事实上有代替goto的方法

e.g.：int sum=0;

outer:

for(int i=0;i<10;i++)

{

for(int j=0;j<10;j++)

{

sum+=1;

if(sum==2)

break outer;

}

}

System.out.println(sum);

结果为2，一般break只能跳出内层循环，结果为92；continue有同样用法

1. C

JAVA对基本数据型变量初始化为0，boolean型初始化为false，Object型初始化为null

1. C

子类覆盖父类不带参数的amethod函数，继承了带一个String参数的amethod函数

1. A

%与被除数符号保持一致

1. D

count=1+1+2+3+4+5=16

1. 多项选择题：
2. A、D

abstract表抽象函数，private、protected不能出现在interface中，否则编译报错

1. C、D

public static void main(String[] StringName)除了StringName都必须一致

1. A、B、C、D

new时新建；start时就绪；获得处理器资源时运行；

sleep时阻塞（等到通知后会重新就绪）；run()完或强制终止时死亡

1. A、B、C、D

A、B、C都需要>>import java.io.\*; 且必须在main函数后加入throws IOException

A、B：Reader读出的是char[]/String，InputStream读取出来的是byte数组（二进制）

FileInputStream是InputStream的子类，专用于从文件中获得输入字节

>>FileInputStream fis = new FileInputStream(String FileName);

InputStreamReader、BufferReader是Reader的子类，

分别将字节流转换为字符流、从字符流中读取文本缓冲各个字符

>>InputStreamReader isr = new InputStreamReader(fis);

>>BufferedReader br = new BufferedReader(isr);

最后可以输出：

>>System.out.println(br.readLine());

C：

* 构造：RandomAccessFile file = new RandomAccessFile(String Filename, "rw");

其中"rw"表示以读写方式打开，可以以只读方式打开："r"，

但不支持以只写方式打开："w"

* 写：修改而非插入模式

writeInt(4字节)、writeDouble(8字节)、writeUTF、writeBoolean(1字节)、 writeShort(2字节)、writeLong(8字节)、writeFloat(4字节)、writeChar(2字节)

* 读：与write相对应
* file.seek(num)：定位到第num个字节，下标从0开始计数，

！！！如int型占4个字节，定位在中间某字节写再读会出乱码

* file. setLength(num)：预分配num个字节的文件空间
* 应用：复制文件

>>int len=(int)file.length();

>>byte[] b=new byte[len];

>>file.readFully(b);

>>fileCopy.write(b);

！！！一般write后打开会是乱码，因为write的编码格式不对，需要转换编码：

>>String temp="你好";

>>String h=new String(temp.getBytes("unicode"),"unicode");

>>rf.write(h.getBytes());

但是不再支持readInt等函数，只能用readLine

1. A、C

内部类：为了可以随意使用外部类的任何变量而不用生成外部类对象

内部类实例化：Out.In in=new Out().new In();

静态内部类：内部类前加static修饰，这样内部类只能访问外部类static成员变量

但可以直接new出内部类对象而不用生成外部类对象：

Out.In in=new Out.In();

私有内部类：内部类前加private修饰，表明只希望被外部类操作

不可以再用new生成内部类对象

局部内部类：内部类定义在外部类方法中

1. A、B、C

访问权限：

private 类内 不同类不能

default 类内+包内 在不同的包中即使import后也不能

protect 类内+包内+继承内

public 公共

但是default意思是不加剩下三个中任何一个而不是加上这个关键词

1. A、B

C为赋值符号，编译都会报错

D比较的是两个不同类，故需使用超类Object中的equals方法，比较的是引用/地址

1. A、C、D

B、C：注意下标从0开始计数即可

1. B、C

①抽象类不能实例化，故不能new B();

②子类可以赋值转给父类（但信息仍保存可以再强制转化为子类对象，且instanceof判断仍属于子类），但是父类不能赋值给子类

1. A、C

A、B：可以通过>>System.out.println(new int[2] instanceof Object);验证

C：这与int[] number={31,23,33,43,35,63}等价

1. 填空题：
2. 0（歧义：0.6666666666666666）

解析见一/1

1. try、catch

try{程序块}

catch(ExceptionType1 e){对ExceptionType1的处理}

catch(ExceptionType2 e){对ExceptionType2的处理}

.......

finally{总是要执行的语句}

1. final
2. 不能编译通过

①只支持replace(int start,int end,String str)：从下标start位置开始到下标end-1为止，替换为str，不支持replace(int,int,char)

②若改为"q"，则输出qbcdfefgh, true

s2==s比较的是两者地址，此时一样

1. 120

s=0+MyIntArray[1]+ MyIntArray[3]+ MyIntArray[5]=0+20+40+60=120

1. Runnable
2. 10

以0开头被视为8进制，0x开头被视为16进制

1. StringBuffer；str.length()

String类是不可被改变的，所以对String类做+时要创建StringBuffer类对象，

StringBuffer类用append函数实现拼接，但不用创建新变量，运行将快很多

1. start

解析见一/4

1. 65432
2. 简答题：
3. 解析见三/2

另：ExceptionType除了系统提供的以外还可以自己定义extends Exception，覆盖

>>public String getMessage()，然后

>>ReturnType MethodName(Parameter) **throws** MyException，在该方法中某些条件下

>>throw new MyException(InitializeParameter);可以实现某种合法性判断

1. List是有序的Collection，且允许有相同的元素

Set是一个不包含重复元素的Collection

Map 提供key到value的映射，不能包含相同的key

1. ①==对于基本类比较的是内容，对于超类Object对象比较的是引用，但-128~127之间的Byte、Integer、Long； Character比较的是缓存中的内容，属于特例

②equals是超类Object的成员函数，只能比较两个Object对象的引用是否相同，除非重载，而String、Integer、Boolean等primitive type的autoboxing自带重载，比较的是内容

1. ①一个class可以implements多个接口，但只能extends一个抽象类

②实现接口时必须实现里面所有的抽象函数，但对于抽象类不一定，可以保留抽象函数

③接口中访问修饰符只能是默认public，但抽象类可以有public/protected/default

④接口中不能有main函数，但抽象类可以有

⑤接口不能有默认实现方法，但抽象类可以有

⑥接口不能有构造函数，但抽象类可以有

1. 解析见一/4
2. 综合题：
3. //可能可以缩减重复代码animal.run();animal.talk();

class Animal

{

public void run()

{ System.out.println("Animal is running."); }

public void talk()

{ System.out.println("Animal is talking."); }

}

class Dog extends Animal

{

public void talk()

{ System.out.println("Wang!Wang!"); }

}

class Cat extends Animal

{

public void talk()

{ System.out.println("Miao!Miao!"); }

}

class Goat extends Animal

{

public void talk()

{ System.out.println("Mie!Mie!"); }

}

public class RandomAnimal

{

public static void main(String[] args)

{

int tmp=(int)(Math.random()\*3);

if(tmp==0)

{

Dog animal=new Dog();

animal.run();

animal.talk();

}

else if(tmp==1)

{

Cat animal=new Cat();

animal.run();

animal.talk();

}

else

{

Goat animal=new Goat();

animal.run();

animal.talk();

}

}

}

1. 第五个猴子看到：1+5k，但是为了(1+5k)/0.8是整数，k=4m-1，所以20m-4

第四个猴子看到：1+(20m-4)/0.8=25m-4，为了能被0.8整除，m=4n，所以100n-4

第三个猴子看到：1+(100n-4)/0.8=125n-4，为了能被0.8整除，n=4p，所以500p-4

第二个猴子看到：1+(500p-4)/0.8=625p-4，为了能被0.8整除，p=4q，所以2500q-4

第一个猴子看到：1+(2500q-4)/0.8=3125q-4，所以最少3121个桃子