一、选择题

1.C

2.D

3.C

4.A

5.B

6.C

7.D

8.D

9.B

10.B

11.C

12.B

13.C

14.AD

15.AC

16.C

17.C

18.A

19.D

20.C

21.C

22.C

23.B

24.C

25.B

26.C

27.B

28.C

29.C

30.A

二、简答题

1.答：

线程的生命周期

线程是一个动态执行的过程，它也有一个从产生到死亡的过程。

生命周期的五种状态

新建（new Thread）：当创建Thread类的一个实例（对象）时，此线程进入新建状态（未被启动）。

就绪（runnable）：线程已经被启动，正在等待被分配给CPU时间片，也就是说此时线程正在就绪队列中排队等候得到CPU资源。

运行（running）：线程获得CPU资源正在执行任务（run()方法），此时除非此线程自动放弃CPU资源或者有优先级更高的线程进入，线程将一直运行到结束。

死亡（dead）：当线程执行完毕或被其它线程杀死，线程就进入死亡状态，这时线程不可能再进入就绪状态等待执行。

堵塞（blocked）:由于某种原因导致正在运行的线程让出CPU并暂停自己的执行，即进入堵塞状态。

创建方式：

1）继承Thread类创建线程

2）实现Runnable接口创建线程

3）使用Callable和Future创建线程

sleep()，yeild()，wait()方法之间的区别：

yield():线程暂时交出CPU的控制权，也就是从running状态进入到runnable状态，但仍然有可能被再次调度。

wait():在其他线程调用此对象的 notify() 方法或 notifyAll() 方法时线程才会继续执行。

sleep():线程指定睡眠一段时间。

2.答：

Serializable接口是启用其序列化功能的接口。实现java.io.Serializable 接口的类是可序列化的。没有实现此接口的类将不能使它们的任意状态被序列化或逆序列化。

3.答：

一般情况下是用try来执行一段程序，如果系统会抛出（throw）一个异常对象，可以通过它的类型来捕获（catch）它，或通过总是执行代码块（finally）来处理；

try：用来指定一块预防所有异常的程序；

catch：子句紧跟在try块后面，用来指定你想要捕获的异常的类型；

throw：语句用来明确地抛出一个异常；

throws：用来声明一个方法可能抛出的各种异常；

finally：为确保一段代码不管发生什么异常状况都要被执行；

4.答：

Java垃圾回收机制的工作原理：

引用计数,标记复制"引用计数"是一种简单但速度很慢的垃圾回收技术。所有对象都有一个引用计数器，当有引用连接时计数器加1，当引用离开作用域时或者被置于NULL时，计数器-1,垃圾回收器会在所有包含对象引用的列表上进行遍历,当发现某个对象的引用计数为0时,就释放占用的空间。"标记复制"的运行机制,垃圾回收器遍历包含所有引用的列表，当发现存活的对象引用时做上标记，这样当遍历完所有对象引用并做上标记的时候，执行垃圾回收，将没有标记的对象堆空间释放。

5.答：

“抽象类”和“接口”的区别如下：

1、抽象类和接口都不能直接实例化，如果要实例化，抽象类变量必须指向实现所有抽象方法的子类对象，接口变量必须指向实现所有接口方法的类对象。

2、抽象类要被子类继承，接口要被类实现。

3、接口只能做方法申明，抽象类中可以做方法申明，也可以做方法实现

4、接口里定义的变量只能是公共的静态的常量，抽象类中的变量是普通变量。

5、抽象类里的抽象方法必须全部被子类所实现，如果子类不能全部实现父类抽象方法，那么该子类只能是抽象类。同样，一个实现接口的时候，如不能全部实现接口方法，那么该类也只能为抽象类。

6、抽象方法只能申明，不能实现，接口是设计的结果 ，抽象类是重构的结果

7、抽象类里可以没有抽象方法

8、如果一个类里有抽象方法，那么这个类只能是抽象类

9、抽象方法要被实现，所以不能是静态的，也不能是私有的。

10、接口可继承接口，并可多继承接口，但类只能单根继承。

6.答：

方法的重写和重载是Java多态性的不同表现。

重写是父类与子类之间多态性的一种表现，重载是一个类中多态性的一种表现。

如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。

子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被"屏蔽"了。

如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。

7.答：

&是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。

三、编程题

1.答：

import java.util.ArrayList;

import java.util.ListIterator;

public class test02 {

public static void main(String[] args) {

ArrayList<String> aList = new ArrayList<String>();

aList.add("abc");

aList.add("T1");

aList.add("abcT2");

aList.add("T3");

aList.add("123");

aList.add("a-b-c");

aList.add("abc");

aList.add("abct4");

aList.add("123abc");

ListIterator<String> lit = aList.listIterator();

while(lit.hasNext()){

if("abc".equals(lit.next())){

lit.remove();

}

}

System.out.println(aList);

//输出结果：[T1, abcT2, T3, 123, a-b-c, abct4, 123abc]

}

}

2.答：

import java.util.Arrays;

public class test02 {

public static String splitString(String str, int byteNum) throws Exception {

char[] c = str.toCharArray();

System.out.println(Arrays.toString(c));

return new String(c, 0, getToIndex(c, byteNum));

}

public static int getToIndex(char[] c, int byteNum) throws Exception {

int num = 0;

for (int i = 0; i < c.length; i++) {

num += ((c[i] + "").getBytes("gb2312")).length;

if (num >= byteNum) {

return i + 1;

}

if (i + 1 == c.length) {

return c.length;

}

}

return 0;

}

public static void main(String[] args) throws Exception {

System.out.println(splitString("我ABC们DEF", 9));

}

}

3.答：…