周考-JAVASE试题

一 .选择题(每题3分,共45分)

1.(单选题)在异常处理中，如释放资源、关闭文件、关闭数据库连接等由（ c）来完成。

A.try子句

B.catch子句

C.finally子句

D.throw子句

2.(多选题)下列异常类中，属于RuntimeException的子类的是：（d, c）。

A.FileNotFoundException

B.IOException

C.ClassCastException

D.ArrayIndexOutOfBoundsException

3.(单选题)对于catch子句的排列，下列哪种是正确的( a)

A.父类在先，子类在后

B.子类在先，父类在后

C.有继承关系的异常不能在同一个try程序段内

D.只能有子类异常

4.(单选题)String[] arr = new String[3];

System.out.println(arr[0].length());

运行结果是（ d）。

A.会出现编译错误。

B.运行期会抛出ArithmeticException异常

C.运行期会抛出IOException异常

D.运行期会抛出NullPointerException异常

5.(单选题)下列关于多线程中锁机制正确的是（ c）。

A.关键字synchronized只能用于方法声明上。

B.成员方法使用synchronized，那么当前方法的锁对象为当前方法所属的对象

C.静态方法使用synchronized，那么当前方法的锁对象为当前方法所属对象

D.为了尽量避免多线程带来的安全问题，尽可能的扩大synchronized修饰的代码范围。

6.(单选题)下列代码编译和运行的结果是：（b）。

public static void main(String[] args) {

try {

String[] s=null;

s[0] = "test";

System.out.println(s[0]);

} catch (Exception ex) {

System.out.println("Exception");

} catch (NullPointerException npe) {

System.out.println("NullPointerException");

}

}

A.输出：test

B.输出：Exception

C.编译失败

D.输出：NullPointerException

7.(单选题)下列代码编译和运行的结果是：（c）。

public class Threads5 {

public static void main(String[] args) {

new Thread(new Runnable() {

public void run() {

System.out.print("bar");

}

}).start();

}

}

A.编译错误

B.抛出运行时异常

C.输出：bar

D.代码正常运行，但是无输出

8.(单选题)下列选项中关于synchronized的使用正确的是：（c）。

A.private synchronized Object o;

B.void go() {

synchronized() { /\* code here \*/ }

}

C.public synchronized void go() { /\* code here \*/ }

D.private synchronized(this) void go() { /\* code here \*/ }

9.线程调用了sleep()方法后，该线程将进入(  c)状态

A.就绪状态     B.运行状态      C.阻塞状态      D.终止状态

10.(多选题)下面哪些是Thread类的方法（a,b）

A start() B run() C exit() D getPriority()

11.哪个关键字可以对对象加锁？( b )

A.transient B.synchronized C.serialize D.static

12.下列不属于线程生命周期状态的是(a  )

A. 新建状态       B.就绪状态  C. 运行状态       D.解锁状态

13. 以下对TCP和UDP描述正确的是（d）  
A. TCP不能提供数据的可靠性  
B) UDP能够保证数据库的可靠性  
C) TCP数据传输效率高于UDP  
D) UDP数据传输效率高于TCP

14.(多选题)关于java.io.Serializable接口说法正确的是:（b,c）。

A.java.io.Serializable中有一个serialID属性，但是没有方法

B.类通过实现java.io.Serializable 接口以启用其对象的序列化功能

C.java.io.Serializable中有一个run方法，但是没属性

D.java.io.Serializable接口没有方法或属性，仅用于标识可序列化的语义。

15.(多选题)下列代码说法正确的是：（a,c,d）。

public class DaemonThread extends Thread{

public DaemonThread(String name){

super(name);

}

public void run() {

while (true) {

System.out.println("Daemon thread running...");

}

}

public static void main(String[] args) {

DaemonThread dt = new DaemonThread("dtThread");

dt.setDaemon(true);

dt.start();

} }

A.名为dtThread的线程为守护线程

B.运行上述代码，main 线程一退出，则虚拟机退出

C.运行上述代码，程序会一直跑下去，因为main线程结束了，但dt线程是一个无限循环，还没有结束，虚拟机不能退出

D.main线程是守护线程

二 .简答题(共15分)

1. 创建线程有几种方式,有什么区别呢?（10分）

有三种方式可以用来创建线程： 1、继承Thread类； 2、实现Runnable接口； 3、应用程序可以使用Executor框架来创建线程池。

区别：使用Runnable接口创建线程 1.可以将CPU，代码和数据分开，形成清晰的模型 2.线程体run()方法所在的类可以从其它类中继承一些有用的属性和方法 3.有利于保持程序的设计风格一致 直接继承Thread类创建对象

1.Thread子类无法再从其它类继承（java语言单继承）。 2.编写简单，run()方法的当前对象就是线程对象，可直接操作。 在实际应用中，几乎都采取第一种方式

1. Collection和Collections的区别?（5分）

Collection与Collections的根本区别是： 1、Collection 是一个集合接口。它提供了对集合对象进行基本操作的通用接口方法。[Collection接口](https://wenwen.sogou.com/s/?w=Collection%E6%8E%A5%E5%8F%A3&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)在Java 类库中有很多具体的实现。Collection接口的意义是为各种具体的集合提供了最大化的统一操作方式。

1. Collections 是一个包装类。它包含有各种有关集合操作的静态多态方法。此类不能实例化，就像一个工具类，服务于Java的Collection框架。
2. Collections 是一个包装类，Collection 表示一组对象，这些对象也称为 collection 的元素。一些 collection 允许有重复的元素，而另一些则不允许，一些 collection 是有序的，而另一些则是无序的。

三 .程序题（每题20分，共40分）

1. 采用匿名内部类的方式创建2个线程，一个线程为计时线程，每隔2s输出当前时间的毫秒表示，

另一个线程为前台线程，任务是线程开始，休眠10秒结束，当前台线程结束，计时线程也随之结束。请完成以上案例。

//林永健

Thread lin = **new** Thread () {

**public** **void** run () {

System.***out***.println("睡一会吧");

**try** {

Thread.*sleep*(10000);

}**catch**(InterruptedException e) {

System.***out***.println("干嘛呢");

}

System.***out***.println("醒了");

}

};//黄宏

Thread huang = **new** Thread () {

**public** **void** run () {

System.***out***.println("砸墙");

**for**(**int** i =0; i<5;i++) {

Date now = **new** Date( ) ;

System. ***out*** . println(now);

SimpleDateFormat sdf = **new** SimpleDateFormat( "ss");

String line = sdf. format (now);

System. ***out***. println(line) ;;

**try** {

Thread.*sleep*(1000);

}**catch**(InterruptedException e) {

}

}

System.***out***.println("哐当");

System.***out***.println("搞定");

// lin.interrupt();//中断lin线程的睡眠效果

}

};

huang.start();

lin.start();

}

2. 创建2个线程对象表示2个人，二人都要完成在客厅看电视5s，然后去卫生间5s的任务，已知二人同时开始执行任务。

要求：2个人可以同时在客厅看电视，但是上厕所不能同时上，只能排队上

提示：synchronized

**public** **static** **void** main(String[] args) {

Shop shop = **new** Shop();

Thread t1 = **new** Thread() {

**public** **void** run () {

shop.buy();

}

};

Thread t2 = **new** Thread() {

**public** **void** run () {

shop.buy();

}

};

t1.start();

t2.start();

}

}

**class** Shop{

/\*

\* 在一个成员方法上直接使用synchronized后，

\* 同步监视器对象就是该方法所属对象this

\*/

//public synchronized void buy() {

**public** **void** buy() {

**try** {

Thread t=Thread.*currentThread*();

System.***out***.println(t.getName()+"看电视");

Thread.*sleep*(5000);

/\*

\* 若希望多个线程可以同步执行该同步块中的代码，则必须保证指定的

\* 同步监视器对象多个线程看到的是“同一个”

\*/

//synchronized(new Object（）)没有同步效果

**synchronized**(**this**) {

System.***out***.println(t.getName()+"去厕所");

Thread.*sleep*(5000);

}

System.***out***.println(t.getName()+"完事了");

} **catch** (Exception e) {

}

}