

北京大学信息科学技术学院考试试卷

考试科目：Java 程序设计 姓名：_____ 学号：_____

考试时间：2018 年 6 月 26 日 任课教师：黄骏

题号	一	二	三	四	五	六	七	八	总分
分数									
阅卷人									

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机等）不得带入座位，已经带入考场的必须放在监考人员指定的位置，并关闭手机等一切电子设备。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束监考人员宣布收卷时，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准旁窥、交头接耳、打暗号，不准携带与考试内容相关的材料参加考试，不准抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有严重违纪或作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学习纪律管理规定》及其他相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 11 页。

装订线内

不要答题

得分

一、填空题（共 30 分）

1. Java 使用默认、`public`、`protected` 和 `private` 四类关键字控制对类成员的访问权限。请用 **Yes** 和 **No** 补全下表，其中 **Yes** 表示可以在该列指定的范围内访问类成员。（6 分）

	同一类中	同一包中	不同包中的子类	不同包中的非子类
默认	Yes	Yes	No	No
<code>public</code>	Yes	Yes	Yes	Yes
<code>protected</code>	Yes	Yes	Yes	No
<code>private</code>	Yes	No	No	No

2. 给出下列程序的输出。如有编译错误请说明错误类型（共 24 分，每题 4 分）。

（1）下列程序的运行结果是： 16

```
class Main {
    public static void main(String[] args) {
        byte a = -112, b = 87;
        System.out.println(a&b);
    }
}
```

（2）下列程序的运行结果是： $2^{31}-1$

```
class Main {
    public static void main(String[] args) {
        byte a = -1;
        System.out.println(a >>> 1);
    }
}
```

(3) 编译运行下列程序时得到的结果是: 1, 3, 2

```
class A {
    public int id = 1;
    public int id() { return id; }
    static public int sid = 2;
    static public int sid() { return sid; }
}
class B extends A {
    public int id = 3;
    public int id() { return id; }
    static public int sid = 4;
    static public int sid() { return sid; }
}
public class Main {
    public static void main(String args[]) {
        A a = new B();
        System.out.print(a.id);
        System.out.print(a.id());
        System.out.print(a.sid());
    }
}
```

(4) 下列程序的运行结果是: 0

```
class A {
    public void add(int x, int y, int sum) {
        sum = x+y;
    }
    public static void main(String[] args) {
        A a = new A();
        int sum;
        a.add(10, 20, sum);
        System.out.println(sum);
    }
}
```

(5) 假设当前路径下有 test.txt 文件，其内容为”hello world”。

则编译运行下列程序时得到的结果是： 编译错误，子类抛出父类异常

```
import java.io.*;
class A {
    public void foo(String path) throws IOException {
        BufferedReader br = new BufferedReader(new FileReader(path));
        System.out.println("A: "+br.readLine());
    }
}
class B extends A {
    public void foo(String path) throws Exception {
        BufferedReader br = new BufferedReader(new FileReader(path));
        System.out.println("B: "+br.readLine());
    }
}
public class Main {
    public static void main(String args[]) throws Exception {
        A a = new B();
        a.foo("test.txt");
    }
}
```

(6) 依次给出下列程序的输出结果： false, true, true

```
public class Test {
    public static void main(String args[]) {
        String str = new String("Hi");
        String str1 = new String("Hi");
        String str2 = new String("Hi");
        System.out.println(str1 == str2);
        System.out.println(str.equals(str1));
        str1 = str;
        str2 = str;
        System.out.println(str1 == str2);
    }
}
```

得分

二、简答题. (40 分)

1. 请从平台、内存管理和安全的角度分析 Java 语言的三个主要特点，并与 C/C++ 进行比较。(6 分)

平台：C/C++是编译型语言，因此与平台相关。而 Java 使得语言声明不依赖于实现的方面。Java 的字节码对新的硬件平台和操作系统是可移植的。

内存管理：C/C++要求程序员管理内存的分配和维护，不仅增加程序员的负担，而且容易造成错误。而 Java 不支持指针，它消除重写存储和讹误数据的可能性。类似地，Java 支持自动的垃圾回收，预防存储漏泄和其它有关动态存储分配和解除分配的有害错误。

安全：Java 运行系统使用字节码验证过程来保证装载到网络上的代码不违背任何 Java 语言限制。这个安全机制部分包括类如何从网上装载。

2. 请解释重载和覆盖的区别。(4 分)

重载指父类与子类中同名但调用参数类型或个数不同的函数；

覆盖指父类与子类中同名调用参数类型和个数都相同的函数。

3. 请说明以下程序在运行过程中出现的问题并解释造成该问题的原因。(5 分)

```
class A {  
    A() { print(); }  
    void print() { System.out.println("Hello"); }  
}  
class B extends A {  
    int [] array;  
    B() { array = new int[2]; }  
    void print() { System.out.println(array[0]+ " " + array[1]); }  
}  
public class Test {  
    public static void main(String [] args) { A a = new B(); }  
}
```

运行时抛出空指针异常。程序运行过程如下：

- (1) 程序首先创建 B 类的实例；
- (2) 执行 B 的构造函数前，Java 自动调用 super 函数，进行父类 A 的初始化；
- (3) 程序进入 A 的构造函数，调用 print；由于该实例的运行时类型为 B，程序将调用 B 的 print 函数；
- (4) 在执行 B 的 print 函数时，程序尝试打印 array[0] 与 array[1]；
- (5) 由于此时 array 数组尚未被初始化，因此程序将抛出空指针异常。

4. 请说明以下程序在运行过程中出现的问题并解释造成该问题的原因。(5分)

```
public class A {
    private int id;
    public A(int id) { this.id = id; }
    synchronized void work(A other) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
        synchronized (other) {
            System.out.println("Thread "+id+" is running.");
        }
    }
}

public static void main(String[] args) {
    A a1 = new A(1);
    A a2 = new A(2);
    Thread t1 = new Thread(()->{a1.work(a2);});
    Thread t2 = new Thread(()->{a2.work(a1);});
    t1.start();
    t2.start();
}
```

该程序会造成死锁。比如下列情形：

- (1) t1 的 work 函数请求自身对象的实例锁，成功后对自身上锁并执行 sleep 语句进入阻塞状态；
- (2) t1 的 work 函数请求自身对象的实例锁，成功后对自身上锁并执行 sleep 语句进入阻塞状态；
- (3) t1 在 sleep 结束后继续执行，并请求 t2 的实例锁；由于此时 t2 的实例锁为锁住状态，t1 将被阻塞；
- (4) t2 在 sleep 结束后继续执行，并请求 t1 的实例锁；由于此时 t1 的实例锁为锁住状态，t2 将被阻塞；

由此 t1 与 t2 将进入死锁状态。

5. 给定以下程序，请判断字符串实例 a 被回收的具体时间点，并对你的结论进行解释。（4 分）

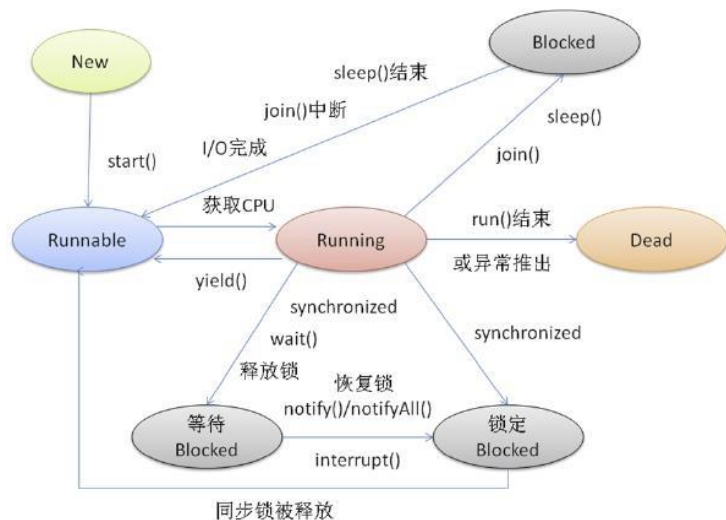
```
public class A {  
    public static void main(String[] args) {  
        String a = new String("hello world");  
        String b = new String("game over");  
        System.out.println(a);  
        a = null;  
        System.out.println(b);  
        b = null;  
        System.gc();  
    }  
}
```

字符串 a 在 a = null 之后可以被回收。但具体回收时间点取决于 JVM 的运行时，因此无法确定。

6. Java 线程在其生命周期中通常经历新建(New)、就绪(Runnable)、运行(Running)、阻塞(Blocked)、和终止(Dead)五种状态。请解释各状态的含义，并说明使各状态之间相互转换的条件和方法。（8 分）

新建：线程尚未开始；就绪：进入排队队列等待 CPU 时间片；运行：正在运行直至时间片用尽或任务完成；阻塞：因人为挂起或资源被占用而终止运行；死亡：完成全部工作或被强制终止。

状态转换条件和方法如下图所示：



7. 下列程序的输出是否永远为 true? 解释原因, 并给出两种方法修改 MyThread 类, 通过线程的同步控制使输出永远为 true。(8 分)

```
class Counter extends Thread {
    private int x, y;
    public void run() { increase(); }
    private synchronized void increase() { while (true) { x++; y++; } }
    public void test() { System.out.println(x==y); }
}

public class Main {
    public static void main(String args[]) throws InterruptedException {
        Counter counter[] = new Counter [10];
        for (int i = 0; i < 10; i++) {
            counter[i] = new Counter();
            new Thread(counter[i]).start();
        }
        while (true)
            for (int i = 0; i < 10; i++) counter[i].test();
    }
}
```

输出并非永远为 true。由于 test 函数不是同步函数, 有可能出现下列情况: 当某线程 counter[i] 在执行 increase 函数的 x++ 指令后时间片用完, 而 main 函数的线程此时调用 counter[i] 的 test 函数比较 x 和 y, 则有可能出现 x 不等于 y 的情况。

修改方法(1): 将 test 函数声明为同步函数, 即

```
public synchronized void test() { System.out.println(x==y); }
```

修改方法(2): 将 main 函数的 while 循环体修改为

```
while (true) {
    for (int i = 0; i < 10; i++) {
        synchronized (counter[i]) { counter[i].test() };
    }
}
```

得分

三、编程题. (共 30 分)

(1) 补全 Fish 类, 使得以下该序输出为 true。(15 分)

```
public class Main {
    public static void main(String[] args) {
        Fish f1 = new fish(20, "Blue");
        Fish f2 = new fish(20, "Blue");
        System.out.println(f1.equals(f2));
    }
}

class Fish {
    private int weight;
    private String color;
    public Fish(int weight, String color) {
        this.weight = weight;
        this.color = color;
    }

    // 在以下位置补充代码

    public int getWeight() { return weight; }

    public String getColor() { return color; }

    public Boolean equals(Object other) {
        if (other == this) return true;

        if (other instanceof Fish && other.getWeight() == this.weight
            && other.getColor() == this.color)
            return true;

        return false;
    }
}
```

(2) 银行金库被盗，共有四个嫌疑人。A 说罪犯在 B、C、D 中。B 说自己是清白的，罪犯是 C。C 说 A 和 B 中有一个是罪犯。D 说 B 说的是真话。事实是四人中有两人说真话，两人说假话。写一段程序求谁是罪犯。(15 分)

```
public class Test {  
  
    public static int countLiers(boolean a, boolean b, boolean c, boolean d) {  
  
        int n = 0;  
  
        if (!b && !c && !d) n++; // A is lying  
        if (b || !c)          n++; // B is lying  
        if (!a && !b)         n++; // C is lying  
        if (b || !c)          n++; // D is lying  
  
        return n;  
    }  
  
    public static void main(String args[]) {  
  
        for (int i = 0; i < 16; i++) {  
  
            boolean a, b, c, d; // 值为 true 时表示对应人为罪犯  
  
            a = ((i>>0)&1)==1; b = ((i>>1)&1)==1;  
            c = ((i>>2)&1)==1; d = ((i>>3)&1)==1;  
  
            if (countLiers(a, b, c, d) == 2) {  
  
                System.out.print("罪犯是:");  
  
                if (a) System.out.print("A ");  
                if (b) System.out.print("B ");  
                if (c) System.out.print("C ");  
                if (d) System.out.print("D ");  
  
                System.out.print("\n");  
  
            }  
  
        }  
  
    }  
}
```