**《专高三第19单元》练习手册**

编写人员：

编写时间：

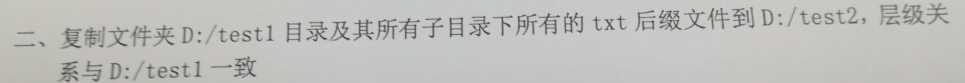
专高三\_分布式架构\_第19单元

练习手册

一、判断题

1. (判断)System.out.println(1^1);输出为0A. 正确B. 错误2. (判断)System.out.println(true^true);输出trueA. 正确B. 错误3. (判断)有两个整数a=3,b=4，在不使用第三个变量前提下是不可以将a和b的数值互换的A. 正确B. 错误4. (判断)JDK的动态代理只能代理接口，Cglib类库可以动态代理类A. 正确B. 错误5. (判断)能够存储在TreeSet中的对象，必须实现java.lang.Comparable接口A. 正确B. 错误6. (判断)如果两个对象hashCode相同，并且比较它们Equals方法返回true，一定是同一个对象A. 正确B. 错误7. (判断)String s1="Java",String s2="Java"，s1和s2的hashCode一定相同A. 正确B. 错误8. (判断)类A的hashCode=100,类B的hashCode=100,A a = new A(),B b = new B();a.equals(b)=true,Set set = new HashSet(),set.add(a),set.add(b),则set中有两个对象A. 正确B. 错误9. (判断)public class TestThread { public void run() { System.out.println("Sub Thread..."); } public static void main(String[] args) { try { TestThread tt = new TestThread(); tt.sleep(1000\*60); System.out.println("Main Thread..."); } catch (Exception e) { e.printStackTrace(); } } } tt.sleep(1000\*60)是代表子线程休眠1分钟A. 正确B. 错误10. (判断)Java程序运行后启动很多线程，其中包括Main主线程，垃圾回收线程，事件监听线程等。A. 正确B. 错误二、单选题1. (单选)public class ItemTest { private final int id; public ItemTest(int id){this.id=id;} public void updateId(int newId){id=newId;} public static void main(String[] args) { ItemTest fa = new ItemTest(42); fa.updateId(69); System.out.println(fa.id); } } 下面描述正确的是：A. 编译错误B. 抛出运行时异常C. ItemTest的属性id没有被修改D. ItemTest的属性被修改为新的值2. (单选)class Foo { private int x; public Foo(int x){this.x = x;} public void setX(int x){this.x = x;} public int getX(){return x;} } public class Gamma { static Foo fooBar(Foo foo) { foo = new Foo(100); return foo; } public static void main(String[] args) { Foo foo = new Foo(300); System.out.print(foo.getX()+"-"); Foo fooFoo = fooBar(foo); System.out.print(foo.getX()+"-"); System.out.print(fooFoo.getX()+"-"); foo = fooBar(fooFoo); System.out.print(foo.getX()+"-"); System.out.print(fooFoo.getX()); } } 输出正确的是：作用是（）A. 300-100-100-100-100B. 300-300-100-100-100C. 300-300-300-100-100D. 300-300-300-300-1003. (单选)public class KungFu { public static void main(String[] args) { Integer x = 400; Integer y = x; x++; StringBuilder sb1 = new StringBuilder("123"); StringBuilder sb2 = sb1; sb1.append("5"); System.out.println((x==y)+" "+(sb1==sb2)); } } 输出正确的是：A. true trueB. false trueC. true falseD. false false4. (单选)1.class A 2.{ 3. public String doit(int x,int y) 4. { 5. return "a"; 6. } 7. 8. public String doit(int...vals) 9. { 10. return "b"; 11. } 12.} 25.A a = new A(); 26.System.out.println(a.doit(4,5)); 输出正确的是：A. 第26行打印"a"到控制台B. 第26行打印"b"到控制台C. 抛出运行时异常D. 因为第8行编译出错5. (单选)class Plant { private String name; public Plant(String name){this.name=name;} public String getName(){return name;} } public class Tree extends Plant { public void growFruit(){} public void dropLeaves(){} } 输出正确的是：A. 代码编译没有问题B. 如果在Tree中添加public Tree(){Plant();}就可以编译C. 如果在Plant中添加public Plant(){Tree();}就可以编译D. 如果在Plant中添加public Plant(){this("fern");}就可以编译6. (单选)public class Pizza { ArrayList toppings; public final void addTopping(String topping) { toppings.add(topping); } public void removeTopping(String topping) { toppings.remove(topping); } } class PepperoniPizza extends Pizza { public void addTopping(String topping) { System.out.println("Cannot add Toppings"); } public void removeTopping(String topping) { System.out.println("Cannot remove pepperoni"); } } Pizza pizza = new PepperoniPizza(); pizza.addTopping("Mushrooms"); pizza.removeTopping("Pepperoni"); 输出结果是：A. 编译错误B. 输出Cannot add ToppingsC. 没有输出D. 抛出NullPointerException7. (单选)public class Venus { public static void main(String[] args) { int[] x = {1,2,3}; int y[] = {4,5,6}; new Venus().go(x,y); } void go(int[]...z) { for(int[] a:z) System.out.println(a[0]); } } 输出结果是：A. 1B. 12C. 14D. 1238. (单选)1.public interface A{ 2. public void doSomething(String thing); 3.} 1.public class AImpl implements A{ 2. public void doSomething(String msg){} 3.} 1.public class B{ 2. public A doit(){ 3. //some code here 4. } 5. 6. public String execute(){ 7. //some code here 8. } 9.} 1.public class C extends B{ 2. public AImpl doit(){ 3. //some code here 4. } 5. 6. public Object execute(){ 7. //some code here 8. } 9.} 输出结果是：A. 所有接口和类都可以编译成功B. C类的第2行编译出错C. C类的第6行编译出错D. AImpl类第2行编译出错9. (单选)1.package test; 2. 3.class Target{ 4. public String name = "hello"; 5.} 哪些类可以读取或修改name属性A. 任何类B. 只有Target类C. 在test包中的任何类D. Target的所有子类10. (单选)abstract class Vehicle{public int speed(){reutrn 0;}} class Car extends Vehicle{public int speed(){return 60;}} class RaceCar extends Car{public int speed(){return 150;}} RaceCar racer = new RaceCar(); Car car = new RaceCar(); Vehicle vehicle = new RaceCar(); System.out.println(racer.speed()+","+car.speed()+","+vehicle.speed()); 输出结果是:A. 0,0,0B. 150,60,0C. 1.5015015E8D. 运行时抛出异常11. (单选)5.class Building{ 6. public class Barn extends Building{ 7. public static void main(String[] args{ 8. Building build1 = new Building(); 9. Barn barn1 = new Barn(); 10. Barn barn2 = (Barn)build1; 11. Object obj1 = (Object)build1; 12. String str1 = (String)build1; 13. Building build2 = (Builing)barn1; 14. } 15. } 16.} 下面描述正确的是：A. 如果删除第10行，可以编译成功B. 如果删除第11行，可以编译成功C. 如果删除第12行，可以编译成功D. 如果删除第13行，可以编译成功12. (单选)21.class Money{ 22. private String country = "Canada"; 23. public String getC(){return country;} 24.} 25.class Yen extends Money{ 26. public String getC(){return super.country;} 27.} 28.public class Euro extends Money{ 29. public String getC(){return super.getC();} 30. public static void main(String[] args){ 31. System.out.print(new Yen().getC()+"-"+new Euro().getC()); 32. } 33.} 输出什么结果：A. CanadaB. null CanadaC. Canada nullD. 第26行编译出错13. (单选)interface Foo{} class Alpha implements Foo{} class Beta extends Alpha{} class Delta extends Beta{ public static void main(String[] args){ Beta x = new Beta(); //哪行代码放在此处，会抛出java.lang.ClassCastException } }A. Alpha a = x;B. Foo f = (Delta)x;C. Foo f = (Alpha)x;D. Beta b = (Beta)(Alpha)x;14. (单选)1.package com.company.application; 2. 3.public class MainClass{ 4. public static void main(String[] args){} 5.} 假定环境变量CLASSPATH指向.,类MainClass在目录/apps/com/company/application下 下面哪两个Java命令可以运行MainClassA. 服务器拒绝请求B. 请求的网页不存在 C. 服务不可用 D. 服务器作为网关或代理，从上游服务器（例如Tomcat）收到无效响应。15. (单选)1.package animals.mammals; 2. 3.public class GrizzlyBear extends Bear{ 4. void hunt(){ 5. Salmon s = findSalmon(); 6. s.consume(); 7. } 8.} 1.package animals.fish; 2. 3.public class Salmon extends Fish{ 4. public void consume(){} 5.} 如果两个类在正确的目录下，Mammal类正确定义findSalmon()方法， 下面哪个修改可以保证代码正确编译A. 在Salmon类第2行添加import animals.mammals.\*;B. 在GrizzlyBear类第2行添加import animals.fish.\*;C. 在GrizzlyBear类第2行添加import animals.fish.Salmon.\*;D. 在Salmon类第2行添加import animals.mammals.GrizzlyBear.\*;16. (单选)class Foo{ public int a = 3; public void addFive(){a+=5;System.out.print("f ");} } class Bar extends Foo{ public int a = 8; public void addFive(){this.a+=5;System.out.print("b ");} } 这样调用代码 Foo f = new Bar(); f.addFive(); System.out.println(f.a); 输出的结果是：A. b 3B. b 8C. b 13D. 编译错误17. (单选)下面六个方法 addListener addMouseListener setMouseListener deleteMouseListener removeMouseListener registerMouseListener 下面有几个方法符合JavaBean Listener命名规范A. 1B. 2C. 3D. 418. (单选)1.interface A{public void aMethod();} 2.interface B{public void bMethod();} 3.interface C extends A,B{public void cMethod();} 4.class D implements B{ 5. public void bMethod(){} 6.} 7.class E extends D implements C{ 8. public void aMethod(){} 9. public void bMethod(){} 10. public void cMethod(){} 11.} 下面描述哪些是正确的；A. 在第3行存在编译错误B. 在第7行存在编译错误C. 在第9行存在编译错误D. 如果定义 D e = (D)(new E()),e.bMethod()是调用第9行代码19. (单选)1.package geometry; 2.public class Hypotenuse{ 3. public InnerTriangle it = new InnerTriangle(); 4. class InnerTriangle{ 5. public int base; 6. public int height; 7. } 8.} 下面哪个描述的类可以引用变量base?A. 任意类都可以B. 没有任何类可以C. 在geometry包下面的类可以D. 必须是Hypotenuse的子类可以20. (单选)10.interface Data{public void load();} 11.abstract class Info{public abstract void load();} 下面哪个选项使用Data接口和Info类是正确的：A. public class Employee extends Info implements Data { public void load(){} }B. public class Employee implements Info extends Data { public void load(){} }C. public class Employee extends Info implements Data { public void load(){} public void Info.load(){} }D. public class Employee implements Info extends Data { public void Data.load(){} public void load(){} }21. (单选)public interface A{public void m1();} class B implements A{} class C implements A{public void m1()} class D implements A{public void m1(int x){}} abstract class E implements A{} abstract class F implements A{public void m1(){}} abstract class G implements A{public void m1(int x){}} 下面选项正确的是：A. 编译错误B. 有一个类编译错误C. 有两个类编译错误D. 有三个类编译错误22. (单选)class TestA{ public void start(){System.out.println("TestA");} } public class TestB extends TestA{ public void start(){System.out.println("TestB");} public static void main(String[] args){ ((TestA)new TestB()).start(); } } 输出的结果是:A. TestAB. TestBC. 编译错误D. 运行时异常三、多选题1. (多选)class One { public One foo(){return this;} } class Two extends One { public One foo(){return this;} } class Three extends Two { //哪些代码可以放在这里编译运行 }A. public void foo(){}B. public Two foo(){return this;}C. public One foo(){return this;}D. public Object foo(){return this;}2. (多选)1.package com.company.application; 2. 3.public class MainClass{ 4. public static void main(String[] args){} 5.} 假定环境变量CLASSPATH指向.,类MainClass在目录/apps/com/company/application下 下面哪两个Java命令可以运行MainClassA. 在当前目录/apps下运行java MainClassB. 在当前目录/apps下运行java com.company.application.MainClassC. 从任意目录运行java -classpath /apps com.company.application.MainClassD. 从/apps/com/company/applicaton目录运行java com.company.application.MainClass3. (多选)大数据\_专高3\_《分布式架构》第10单元：不能用来修饰interface方法的有（）A. privateB. publicC. protectedD. static4. (多选)abstract class A{ abstract void a1(); void a2(){} } class B extends A{ void a1(){} void a2(){} } class C extends B{void c1(){}} 下面代码，哪些是合法的方法多态调用 A x = new B(); C y = new C(); A z = new C();A. x.a2();B. z.a2();C. a.a1();D. x.a1();5. (多选)下面哪三个选项的描述是正确的：A. 当且仅当类X是抽象类时，X中的一个final方法可以是抽象的B. 类X中的保护方法，可以被X的子类重写C. 类X中的公共静态方法，可以被X的子类直接调用D. 类X中的私有final方法，可以被X的子类重写6. (多选)public class Car { private int wheelCount; private String vin; public Car(String vin) { this.vin = vin; this.wheelCount = 4; } public String drive() { return "zoom-zoom"; } public String getInfo() { return "VIN:"+vin+"wheels:"+wheelCount; } } public class MeGo extends Car { public MeGo(String vin){ this.wheelCount = 3; } } 上面代码中哪两个地方需要修改，才可以编译成功A. 在Car的构造方法中添加调用this()B. 在MeGo的构造方法中添加调用this()C. 在MeGo的构造方法中添加调用super(vin)D. 修改Car类中属性wheelCount为保护属性7. (多选)package com.sun.scjp; public class Geodetics { public static final double DIAMETER = 12756.32; } 下面正确使用变量DIAMETER的代码是：A. import com.sun.scjp.Geodetics; public class TerraCarta { public double halfway() { return Geodetics.DIAMETER/2.0; } }B. import static com.sun.scjp.Geodetics; public class TerraCarta { public double halfway(){return DIAMETER/2.0;} }C. import static com.sun.scjp.Geodetics.\*; public class TerraCarta { public double halfway(){return DIAMETER/2.0;} }D. package com.sun.scjp; public class TerraCarta { public double halfway(){return DIAMETER/2.0;} }8. (多选)10.class Inner{ 11. private int x; 12. public void setX(int x){this.x=x;} 13. public int getX(){return x;} 14.} 15. 16.class Outer{ 17. private Inner y; 18. public void setY(Inner y){this.y=y;} 19. public Inner getY(){return y;} 20.} 21. 22.public class Gamma{ 23. public static void main(String[] args){ 24. Outer o = new Outer(); 25. Inner i = new Inner(); 26. int n = 10; 27. i.setX(n); 28 o.setY(i); 29. //选择代码放在这里 30. System.out.println(o.getY().getX()); 31. } 32.} 哪三个选项中的代码放在29行，程序可以打印出100A. n=100;B. i.setX(100);C. o.getY().setX(100);D. i=new Inner();i.setX(100);o.setY(i);9. (多选)11.public class Commander{ 12. public static void main(String[] args){ 13. String myProp = /\*选择代码放在这里\*/ 14. System.out.println(myProp); 15. } 16.} 运行命令 java -Dprop.custom=gobstopper Commander 下面哪两个选项放在第13行，程序可以输出gobstopperA. System.load("prop.custom");B. System.getenv("prop.custom");C. System.getProperties().getProperty("prop.custom");D. System.getProperty("prop.custom");10. (多选)class Foo{ static void alpha(){} void beta(){} } 下面选项中哪两个描述是正确的：A. Foo.beta()可以正确调用方法beta()B. Foo.alpha()可以正确调用方法alpha()C. 方法beta()可以直接调用方法alpha()D. 方法alpha()可以直接调用方法beta()

四、技能题



要求：

合理正确设计类，

正确使用递归穷举文件或目录，

正确使用流赋值文件内容，

正确实现结果输出，