### 1) 在Java中，如果父类中的某些方法不包含任何逻辑，并且需要有子类重写，应该使用（c）关键字来申明父类的这些方法。 a) Finalc b) Static c) Abstract d) Void 2) 给定两个java程序，如下： public interface Face{ int counter = 40; 40 } public class Test implements Face{ private static int counter; 0 public static void main(String[ ]args){ System.out.println(++counter); +1=1 } } Test.java 的编译运行结果是（d）。 a) 40 b) 41 c) 0 d) 1 3) 给定java代码，如下： public class Test{ static int i; public int aMethod( ){ i++; return i; } public static void main(String [] args){ Test test = new Test( ); test.aMethod( ); 1 System.out.println(test.aMethod( )); 2 } }编译运行后，输出结果是（ c）。 a) 0 b) 1 c) 2 d) 3 4) 给定java代码，如下： abstract class Shape { abstract void draw( ); } 要创建Shape类的子类Circle，以下代码正确的是（b,d）。 a) class Circle extends Shape{ int draw( ){} } b) abstract class Circle extends Shape{ } c) class Circle extends Shape{ void draw( ); d) class Circle extends Shape{ void draw( ){} } 5) 给定java代码，如下： class Parent{ public void count( ){ System.out.println(10%3); 1 } } public class Test extends Parent{ public void count(int i){ System.out.println(10%i); } public static void main(String[]args){ Parent p = new Test( ); p.count(3); 10%3 1 } } 编译运行，结果是（d）。 a) 1 b) 3 c) 3.333333333333335 d) 编译错误 7) 给定java程序，如下： public class Test{ private static final int counter=10; 10 public static void main(String [] args){ System.out.println(++counter); } } 编译运行Test.java，结果是 （c ） a) 10 b) 11 c) 编译错误 d) 运行时出现异常 8) 在java中，以定义了两个接口B和C，要定义一个实现这两个接口的类，以下语句正 确的是 （c） a) interface A extends B,C b) interface A implements B,C c) class A implements B,C d) class A implements B,implements C 9) 给定一个Java程序代码，如下： public class Test{ int count = 9; public void count1(){ int count =10; System.out.println("count1="+count); 10 } public void count2(){ System.out.println("count2="+count); 9 } public static void main(String args[]){ Test t=new Test(); t.count1(); t.count2(); } 行编译后，输出结果是（b） a) count1=9 count2=9 b) count1=10 count2=9 c) count1=10 count2=10 d) count1=9 count2=10 11) 给定如下 java 代码， 以下（a d）访问控制修饰符可以填入下划线处。 class Parent{ protected void eat(){} } class Child extends Parent{ \_\_\_\_\_ void eat(){} } a) Protected b) Private c) 什么也不填 d) Public 12) 在Java中，下面关于抽象类的描述正确的是（bd）。 a) 抽象类可以被实例化 b) 如果一个类中有一个方法被声明为抽象的，那么这个类必须是抽象类 c) 抽象类中的方法必须都是抽象的 d) 声明抽象类必须带有关键字abstract 13) 给定如下Java代码,以下（ab）方法可以加入Child类中。 Public class Parent{ int change(){…} } Class Child extends Parent{ () } a) public int change(){} b) int chang(int i){} c) private int change(){} d) abstract int change(){} 14) 在java中，在定义类时加上修饰符（d）可以实现该类不能在本类被实例化。 a) Final b) Public c) Private d) Abstract 15) 在java中，下面（b）的陈述是正确的。 a) 私有方法不能被重载 b) 静态方法能被重载 c) 公有方法被重写后的访问修饰符可以变成private d) 一个被重写的方法不能抛出一个在基类中不被检查的异常 16) 分析下面的java代码输出结果为（d）。 Class Point{ Int x,y; Point(int x,int y){ This.x=x;//值传递 This.y=y; } Public ststic void main(String[] args){ Point pt1,pt2; Pt1=new Point (3,3); Pt2=new Point(4,4); System.out.print(pt1.x+pt2.x); } } a) 6 b) 34 c) 8 d) 7 17) 在java中，以下程序的输出结果是（c）。 Class Point { Int x; Boolean y; Void output() { System.out.println(x); System.out.println(y); } Public static void main(String[] args) { Piont pt =new Point(); Pt.output(); } } a) 运行错误 b) 0 Ture c) 0 False d) 0 0 18) 给定java代码如下，编译运行结果是（a）。 public class Test{ public int count(){ return 1%9; 0 } public static void main(String[] args){ System.out.println(count()); } } a) 编译错误 b) 运行时出现例外 c) 正常运行，输出1 d) 正常运行，输出0 20) 在Java接口中，下列选项里有效的方法声明是（ab） a) public void aMethod(); b) void aMethod(); c) static void aMethod(); d) protected void aMethod(); 22) 给定java代码，如下：编译运行，结果是（a） public static void main (String [] args){ String s; System.out.println(“s=”+s); } a) 编译错误 b) 编译通过，但出现运行时错误 c) 正常运行，输出s=null d) 正常运行，输出s= // 23) 给定java代码，如下，编译运行后,结果是 (b )。 public class Test{ static String s; publis static void main(String args[ ]){ char c=s.charAt(0); System.out.println(c); } } a) 编译错误 b) 运行期异常,异常类型为 NullpointerException c) 运行期异常,异常类型为 ArithmeticExceptioin d) 运行正常,但无输出内容