姓名：曾泇睷 学号：1820221053 习题5

1. 判断题(叙述正确的,在括号内打√,否则打X)

(1) 子类继承父类的构造方法。（X）

(2) 子类中想使用被子类隐藏的实例成员变量或实例方法,就需要使用关键字 super。（√）

(3) 可以用final修饰构造方法。（X）

(4) 如果在子类的构造方法中，没有显式地写出 super 关键字来调用父类的某个构造方法,那么编译器默认地有“super();”调用父类的无参数的构造方法，如果父类没有这样的构造方法,代码将出现编译错误。（√）

(5) 可以同时用final和abstract 修饰同一个方法。（X）

(6) 子类继承的方法所操作的成员变量一定是被子类继承或隐藏的成员变量。（√）

(7) 如果一个类的所有构造方法的访问权限都是private的，意味着这个类不能有子类，理由是：一个类的private方法不能在其他类中被使用，但子类的构造方法中一定会调用父类的某个构造方法。（√）

(8) 子类在方法重写时，不可以把父类的实例方法重写为类(static)方法，也不可以把父类的类(static)方法重写为实例方法。（√）

(9) abstract类中只可以有abstract 方法。（X）

(10) 接口中的方法也可以用private或protected修饰。（X）

2. 单选题

(1) 下列代码中，（**D**）是正确的。

interface Com {

intMAX;

public void stop();

void start();

abstract void loop();

}

A. “voidstart();”会导致编译出错。

B. “public void stop();”会导致编译。

C. “abstract void loop();”会导致编译出错。

**D. “int MAX;”会导致编译出错。**

(2) 下列选项中,( **A** )代码替换源文件 Com.java 中的【代码】不会导致编译错误。

public interface Com (

intM=200;

int f();

}

class ImpCom implements Com {

【代码】

}

1. **public int f() {**

**return 100 + M;**

**}**

1. int f() {

return 100;

}

1. public double f() {

return 2.6;

1. public abstract int f();

(3) 下列选项中，( **B** )代码放入程序中标注的【代码】处将导致编译错误。

A. public float getNum()(return 4,.0f;)

**B. public void getNum(){ }**

C. public void getNum(double d){ }

D. public double getNum(float d){return 4.0d;}

class A(

public float getNum()(

return 3.0f;

}

}

public class B extends A{

【代码】

}

(4) 下列选项中，( **C** ) 是正确的。

A. final类不可以有父类

B. 可以用new运算符和abstract类的构造方法创建对象

**C. 如果一个类中有abstract 方法，这个类必须是abstract 类**

D. static方法也可以同时是 abstract 方法

(5) 假设C是B的子类,B是A的子类,cat是C类的一个对象,bird是 B类的一个对象。下列选项中，( **D** ) 是错误的。

A. cat instanceof B的值是 true

B. bird instanceof A的值是true

C. cat instanceof A的值是 true

**D. bird instanceof C的值是true**

(6) 下列选项中，( **A** ) 是正确的。

**A. final 类不可以有子类**

B. abstract 类中只可以有abstract方法

C. abstract 类中可以有非abstract方法，但该方法不可以用final修饰

D. bird instanceof C 的值是true

3. 挑错题 (ABCD注释标注的哪行代码有错误?)

(1) **B**

abstract class AAA (

final static void speakHello() {} // A

**final abstract void cry(); // B**

static final int f(){return 0 ;} // C

abstract float g(); // D

}

(2) **D**

abstract class Animal (

int m = 100;

}

class Dog extenda Animal {

double m;

}

public class E {

public static void main(String args[ ]){

Animal animal = null; // A

Dog dog= new Dog( );

animal = dog; // B

dog.m = 3.14; // C

**animal.m = 3.14; // D**

}

}

(3) **D**

public interface Com {

abstract void cry(); // A

public int x=0; // B

static int y=0; // C

}

abstract class Cat implements Com (

**abstract void cry(); // D**

}

(4) **D**

Interface Com {

int MAX = 100; // A

void f();

}

abstract class Animal implements Com {

int MIN; // B

}

Class Dog extends Animal {

Public void f() {

MIN = 10; // C

**MAX = 200; // D**

}

}

4. 阅读程序题(给出【代码】注释标注的代码的输出结果)

(1) **22**

class A (

int f(int x, int y) (

return x+y;

}

}

class B extends A (

int f(int z, int y)(

int m = super.f(x,y)+10;

return m;

}

}

public class E {

public static void main(String args[ ]) {

A a = new B();

System.out.println(a, f(2,10)); //【代码】: **22**

}

}

(2) **6:6:100**

class Animal {

Int m = 100;

public int seeM() {

return m;

}

public int getM() {

return m;

}

}

class Dog extends Animal {

int m = 6;

public int seem() {

return m;

}

}

public class E {

public static void main(String args[]) {

Animal animal = null;

Dog dog = new Dog();

animal= dog;

System. out. printf(" \* d: \* d; % d", dog, seeM(), animal. seeM( ) , animal. getM()); //【代码】: **6:6:100**

}

}

(3) **18:18**

interface Com {

Int add( int a, int b);

}

abstract class People {

Abstract int add(int a, int b);

}

class Student extends People implements Com {

public int add(int a, int b) {

return a + b;

}

}

Public class E {

public static void main(String args[]) {

Student stu = new Student() ;

Com com = stu;

Int m = com.add(12, 6);

People p = stu;

Int n = p.add(12, 6);

System.out.printf(“%d:%d”, m, n); //【代码】: **18:18**

}

}

(4) **18:12:7:6**

Interface Com {

Int add(int a, int b);

public static int get(int n) {

return n;

}

public default int see(int n)(

return n;

}

public default int look(int n)(

return n;

}

}

class A implements Com {

public int add(int a, int b) {

return a + b;

}

public int see(int n){

return n+1;

}

}

public class E (

public static void main(String args[]) {

A a = new A();

Int m = a.add(12, 6);

Int n = Com.get(12);

Int t = a.see(6);

Int q = a.look(6);

System.out.printf(“%d: %d: %d: %d”, m,n,t,q); //【代码】: **18:12:7:6**

}

}

5.编程题

(1)写一个类 A,该类创建的对象可以调用方法 f输出英文字母表;然后编写一个该类的子类 B,要求子类 B 必须继承 A 类的方法 (不允许重写),子类创建的对象不仅可以调用方法 输出英文字母表，而且可以调用子类新增的方法 g 输出希腊字母表。

class A{

public final void f(){

char cStart='a', cEnd='z';

for (char c=cStart; c<=cEnd; c++)

System.out.println(" " + c);

}

}

class B extends A {

public void g() {

char cStart='α',cEnd='ω';

for(char c=cStart; c<=cEnd; c++)

System.out.println(" "+c);

}

}

public class No5\_1 {

public static void main(String args[]){

B b=new B();

b.f();

b.g();

}

}

(2)写一个类,该类有一个方法 public int f(int a，int b),该方法返回a和b的最大公约数;然后编写一个该类的子类,要求子类重写方法 f而且重写的方法将返回 a 和 b的最小倍数。要求在重写的方法的方法体中首先调用被隐藏的方法返回 a 和 b 的最大公约数 m,然后将乘积(a\* b)/m 返回。在应用程序的主类中创建一个父类的对象,例如 a,该对象a 调用方法 {返回最大公约数,然后 a作为子类对象的上转型对象调用方法 返回最小公倍数。

public class lcm extends hcf{

public int f(int a, int b){

hcf tem=new hcf();

return a\*b/tem.f(a, b);

}

}

public class hcf {

public int f(int a, int b){

if(a<b){

int t=a;

a=b;

b=t;

}

while(a%b != 0){

int t= a%b;

a=b;

b=t;

}

return b;

}

}

public class Main {

public static void main(String[] args) {

hcf A=new hcf();

lcm B=new lcm();

System.out.println("12和64的最大公约数="+A.f(12, 64)+"\n12和64的最小公倍数="+B.f(12, 64));

}

}