**GA09班测试题**

**得分：88**

1. **不定项选择题（20分，10题） +18**

1、以下关于类，与对象的说法那些正确？（ADE）

A、所有的类都是Object的子类

B、所有的对象都是Object的实例

C、一个类可以继承多个父类

D、多个类可以实现同一个接口

E、一个接口可以继承多个接口

F、抽象类中，非抽象方法可以调用抽象方法

2、以下关于构造函数的说法那些正确?（BCDEGH）

A、构造函数可以被子类继承

B、子类必需调用父类的构造函数

C、构造函数可以是私有的

D、构造函数没有返回值

E、当没有为类指定构造函数时，类将具有默认的无参构造函数

F、当为类指定了有参数的构造函数后，可以通过无参数的构造函数来建立实例

G、构造函数可以被重载

H、一个类可以有多个构造函数，且它们之间可以相互调用

3、指出下列哪个方法与方法public void method(int a){}为合理的重载方法。（BCD）

A、public intmethod(int a)

B、public void method(long a){}

C、public void method(inta,int b){}

D、public void method(float a){}//是否要加个方法体？？

4、以下关于Java的集合对象的说法那些正确?（ABCDEFH）

A、List是有序的，允许存在空元素，同一对象允许多次添加到同一list中。

B、List.contains是判断某个对象是否在list中，其调用的是对象的equals方法进行比较的。

C、Set是无序的，Set不允许有空元素.

D、所有实现了Collection接口的对象都可以采用java的for…each语句对其进行遍历.

E、Set元素是不可重复的。

F、Map的key不可重复

G、可以使用java的foreach语句遍历Map对象

H、Map的值可以重复

5、给定某java程序片段，如下：

int i=1;

int j=i++;

if((i>++j)&&(i++==j))｛

i+=j;

｝

System.out.println(i);

该程序运行后，i的输出结果为（B）。

A、1 B、2 C、3 D、4

6、给定如下Java代码，编译运行时，以下( AC)语句的值是true.+2

String s = “hello”;

String t = “hello”;

String e = new String(“hello”);

char[] c = {‘h’,’e’,’l’,’l’,’o’};

A、s.equals(t); B、t.equals(c); C、t.equals(e); D、t==c;

7、给定一个java程序的代码片断如下，编译执行后的输出结果是（C）

char c=’a’;

int i=c;

float f=i++;

byte b=(byte)(f+1);

System.out.println(c+”,”+i+”,”+f+”,”+b);

A、a,97 ,97.0 ,98; B、a ,97, 97, 98;

C、a ,98 ,97.0,98; D、a ,98 ,97,98

8、阅读以下的代码，运行后结果输出( B)

public class Demo{

public static void main(String[] args){

String str = new String(“good”);

char[] ch = {‘a’,’b’,’c’};

change(str,ch);

System.out.print(str + “and ”);

System.out.print(ch);

}

public static void change(String str,charch[]){

Str = “test ok”;

ch[0] = ‘g’;

}

}

A、good and abc B、good and gbc//没看明白

C、test ok and abc D、test ok and gbc

9、在下面代码段中，要访问类B中的AddPrint()方法，应添加（ A ）代码

**public class A**{

public void AddPrint(){

//do something

}

}

**public class B extends A**{

public void AddPrint(){

//do something

}

}

------

public static void main(String[] args){

A a = new B();

//此处添加代码；

}

A、a. AddPrint(); B、B.AddPrint(); C、以上都不对

10、在switch(expression) 语句中，expression的数据类型不能是（A）

A) double B) char C) byte D) short

1. **填空题（20分，10题） +15**

1、在空白处填入正确内容

public byte count(byte b1,byte b2){

return \_\_(byte)(b1 + b2)\_\_\_\_;

}

Pubilcint count(char c,inti,double d){

return \_(int)(c + i + d)\_\_\_\_\_;

}

2、下面程序运行会发生什么结果?如果有错误,如何改正? -1

interface A{

int x = 0;

}

class B{

int x =1;

}

class C extends B implements A {

public void pX(){

System.out.println(x); \_\_\_\_\_编译失败，属性名重名\_\_\_\_\_\_\_\_\_\_\_\_\_

}

public static void main(String[] args) {

new C().pX();

}

}

3、下列程序的运行结果是\_\_\_26\_\_\_\_\_\_\_\_\_\_

class A{

class Dog{

private String name;

Privateint age;

Publicint step;

Dog(String s,int a)

{

name=s;

age=a;

step=0;

}

public void run(Dog fast)

{

fast.step++;

}

}

public static void main (String args[])

{

A a=new A();

Dog d=a.newDog(“Tom”,3);

d.step=25;

d.run(d);

System.out.println(d.step);

}

}

4、以下语句输出的结果是什么？

Publicstatic void main(String[] args){

System.out.println("sum:" + getNum(1));

}

Publicstatic intgetNum(int a){

try{

return a++;

}finally{

System.out.println("finally:" + a);

}

}

sum:\_\_1\_\_\_ finally:\_\_\_2\_\_\_\_\_\_

5、有如下代码，完成注解部分的代码: -1

Writer out = new FileWriter(“error.txt”);

try{

intnum = 1/0;

}catch(Exception e){

//a.完成此处代码，请将异常的消息，及堆栈信息保存到error.txt文件中

Out.write(e.toString().getBytes());

//b.将e转换成RuntimeException并附加"计算出错"的错误消息，并抛出。

throw new RuntimeException(“计算出错”);

}

6、给定如下List对象，请写出按学员Student的年龄从低到高进行排序的代码 -2

List<Student> students = new ArrayList<Student>();

//Tom是姓名,20是年龄

students.add(new Student("Tom", 20));

students.add(new Student("Jim", 22));

students.add(new Student("Tony", 18));

students.add(new Student("JJ", 12));

//容器的排序忘记了

1. 创建一个年龄比较器
2. 再用Collections.sort(students,comparator);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7、给定List如下，请将其转换成字符串数组，并将数组转换成字符串序列输出。

List ls = new ArrayList();

ls.add("Tom");

ls.add("Jim");

\_\_\_\_\_String[] s = ls.toArray();\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8、以下代码输出的结果为:\_\_\_\_name:Jim\_\_\_\_\_\_\_\_\_\_\_\_

Student stud = new Student('01','Tom');

List ls = new ArrayList();

ls.add(stud);

stud.name = 'Jim';

System.out.println("name:" + ((Student)ls.get(0)).getName());

9、请完成以下基本类型与字符串的转换代码： -1

"2" -> 2:\_\_Integer.passInteger(“2”)\_\_\_\_

false ->"NO":\_\_boolean?”YES”:”NO”;\_\_\_\_\_(三目运算)

'a' -> 97:\_\_\_int i = ‘a’\_\_\_\_\_\_(取ascii码)

将int a = 0;转换成Integer的封装类型：\_\_new Integer(a);\_\_\_\_\_\_\_\_\_\_

10、假设i=10, j=20, k=30，则表达式 !(i<j+k) || !(i+10<=j) 的值为\_\_false\_\_\_\_\_\_

1. **简答题（20分，5题） +20**
2. 请描述什么是类？类与对象之间的关系？

答：类是对具有共同属性和行为的一列事物(对象)的描述。

对象是类的实例化。

1. 重载与重写有何不同？

答：重载：在同一个类中，方法名相同，参数列表不同的方法称之为重载。

重写：在子父类中，返回值类型、方法名、参数列表都相同的方法称之为重写。子类的访问权限必须大于或等于父类。

重载在同一个类中，重写在子父类的继承关系中。

1. 请指出类、抽象类、接口之间的区别？

答：类是对事物属性和行为的具体描述。是可以实例化对象的。

抽象类不能直接实例化对象，需由抽象类的子类来实例化对象。抽象类可以有抽象方法。抽象方法是只定义了方法名称而没有方法体，抽象方法需由子类进行复写。

接口里面都是抽象方法。实现接口时需要复写接口里所有的抽象方法。

1. 面向对象的三大特征是什么？写出各自的定义？

答：面向对象的三大特征：继承、封装、多态

继承：在子父类中，子类继承父类的属性和行为，可以使用父类中的所有公共成员。继承提高了代码的复用性。

封装：在类内部封装了特有的成员，对外只暴露了公有的方法进行调用。对于类内部如何完成行为的外部并不需要知道。提高了代码的安全性。

多态：多态体现在继承关系中，对于同一行为的不同实现体现了面向对象的多态性。多态提高了代码的扩展性。

1. 请描述线程的同步与异步之间的区别？

答：同步：在同一进程的不同线程中，拥有相同锁的同步代码在同一时刻只能执行持有锁的代码，必须等该代码执行完毕后将锁释放，其他线程才有机会执行。同步线程是协作式。

异步：在同一进程的不同线程中，线程的执行是依靠对cpu执行权的抢夺，谁抢到执行权谁就执行。异步线程是抢占式。

1. **编程题（40分）+35**

**题目1：**完成以下代码，有记事本文件info.txt的内容如下：

"欢迎到到Java的世界，我是Jerry,欢迎与我联系，我的联系地址是jerry@gmail.com，地址：XMBCIT"

请完成使用Java的文件IO操作，读取文件中的内容，将所有的邮件地址改成

"tony@gmail.com"然后写回到原文件的代码。

**package** 题目1;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.io.OutputStreamWriter;

**publicclass** Test {

**publicstaticvoid** main(String[] args) {

BufferedReader br = **null**;

BufferedWriter bw = **null**;

**try** {

br = **new**BufferedReader(**new** InputStreamReader(**new** FileInputStream("info.txt")));

String line = **null**;

line = br.readLine();

System.*out*.println(line);

String reline = line.replaceAll("jerry@gmail.com","tony@gmail.com");

bw = **new** BufferedWriter(**new** OutputStreamWriter(**new** FileOutputStream("info.txt")));

bw.write(reline, 0, reline.length());

bw.flush();

} **catch** (Exception e) {

e.printStackTrace();

}**finally**{

**try** {

**if**(br != **null**)

br.close();

} **catch** (IOException e) {

e.printStackTrace();

}

**try** {

**if**(bw != **null**)

bw.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

}

**题目2：**利用所学的知识（面向对象、异常、多线程、gui）编写一个点名系统。

**package** 题目2;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JTextField;

**publicclass** Test {

JButton start,end;

JTextField name;

**publicstaticboolean***flag* = **false**;

**publicstaticvoid** main(String[] args) {

**new** Test();

}

**public** Test() {

init();

}

**publicvoid** init(){

JFrame frame = **new** JFrame("点名系统");

frame.setBounds(200, 200, 300, 200);

frame.setLayout(**null**);

name = **new** JTextField();

name.setBounds(100, 20, 90, 40);

frame.add(name);

start = **new** JButton("开始点名");

start.setBounds(35, 80, 100, 40);

frame.add(start);

end = **new** JButton("点到谁了");

end.setBounds(145, 80, 100, 40);

frame.add(end);

myEvent();

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

frame.setVisible(**true**);

}

**publicvoid** myEvent(){

start.addActionListener(**new** ActionListener() {

@Override

**publicvoid** actionPerformed(ActionEvent e) {

**if**(!*flag*){

*flag* = **true**;

**new** GetName().start();

}

}

});

end.addActionListener(**new** ActionListener() {

@Override

**publicvoid** actionPerformed(ActionEvent e) {

*flag* = **false**;

}

});

}

**class** GetName **extends** Thread {

String[] n = {"张三","李四","王五","赵六","周七","小八","小九","小明","小王","小白","小李"};

**publicvoid** run(){

**while**(*flag*){

**for**(String s : n){

System.*out*.println(s);

name.setText(s);

**if**(!*flag*){

**break**;

}

**try** {

Thread.*sleep*(5);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

}

}

}