D ESAME 03/07/4

Test 1	l	java A	\bigcirc
00	•		

00	import java.util.*;
01	public class A {
02	A(int m) {
.03	<pre>Collection<string> b = new TreeSet<string>();</string></string></pre>
04	for (int k=0;k<10;k++) {
05	String s="W"+(k%m);
06	b.add(s);
07	}
80	int count=0;
09	<pre>Iterator<string> i=b.iterator();</string></pre>
10	while (i.hasNext()) {
11	String s=i.next();
12	System.out.print(s);
13	}
14	}
15	<pre>public static void main(String[] a) { new A(3); }}</pre>

Test 2 java B

02

03

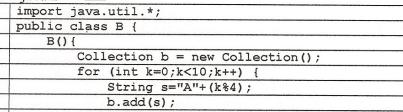
04

05

06

07

08



15 System.out.println(count);
16 }
17 public static void main(String

public static void main(String[] a) { new B(); new B(); }

public static void main(String a) { new B(); }}

Test 3 java C

00	class C{
01	<pre>public static int x;</pre>
02	C(int s) {x=s;}
03	<pre>void f() {System.out.print(x);}</pre>
04	<pre>public static void main(String a[]){</pre>
05	C b=new C(4);
06	C c=new C(1);
07	b.f();
08	c.f();
09	} }

Test 4: java E 01 class E { static int s=0; 02 E(int i) {s=i;} 03 public static void main(String[] args) { 04 E b1=new E(3); 05 E b2=new E(3); 06 E b3=new E(1); 07 if (b1.equals(b2)) System.out.print("K"); else 08 System.out.print("I"); if (b1.s==b3.s) System.out.print("D"); else 09 System.out.print("A"); 10

Test 5 01 #include <iostream> 02 using namespace std; 03 void g(char x[], int y) { y--; 04 x[y]--; 05 06 void f(char *x, int * y) { (*y)++; 07 x[*y]++; 08 09 int main(){ char x[2]; 10 int y; 11 x[0]='G'; x[1]='G';y=0;12 f(x,&y); 13 14 g(x,y);cout<<x[0]<<" "<<x[1]<<" "<<y; 15 return 0; } 16

Test 6 java G public class G implements Cloneable{ 01 int k=0; 02 public G clone() { 03 G copia=null; 04 05 try { copia=(G) super.clone(); 06 } catch (CloneNotSupportedException ex) { 07 System.exit(0); 08 09 copia.k++; 10 return copia; 11 12 public boolean equals(Object x) { 13 if (! (x instanceof G)) return false; 14 return k == ((G)x).k;15 16 public static void main(String[] args) { 17 G b= new G(); 18 G c=b.clone(); 19 G d=new G(); 20 if (b.equals(c)) System.out.print("A"); 21 if (c.equals(d)) System.out.print("J"); 22 if (d.equals(b)) System.out.print("K"); 23 24 }}

Test 7 java F

105	i / java i
01	public class F{
02	int x=2;
03	F(int x) {
04	f(x);
05	f();
06	System.out.println(x);
07	}
08	<pre>void f() { x++; System.out.print(x);}</pre>
09	<pre>void f(int x) { this.x++; x;System.out.print(x);}</pre>
10	<pre>public static void main(String arg[]) {</pre>
11	F x=new F(3);
12	}}

Test 8 java D

	is java D
01	public class D {
02	static int x=1;
03	class String5 {
04	String s;
05	String5() {this.s=""+(++x);}
06	<pre>public void finalize() { System.out.print(s); }</pre>
07	}
08	String5 a=new String5();
09	<pre>void f() {String5 a=new String5();</pre>
10	}
11	<pre>public void finalize() { System.out.print("S"); }</pre>
12	<pre>public static void main(String[] args) {</pre>
13	D a5=new D(); a5.f();
14	a5=new D(); a5.f();
15	System.gc(); System.runFinalization();
16	}}

Test 9 – scrivere nel campo per l'output del test la sequenza risultante indicando V per le affermazioni vere e F per quelle false

9.1	Se A è padre di B la scrittura B a=(B)(new A()); genera errore a compile time
9.2	Se B estende A la scrittura B a=new A(); genera errore a compile time
9.3	Un oggetto ed un suo clone sono identici.
9.4	Poichè Java usa sempre dynamic binding, esso usa sempre la heap e mai lo stack.
9.5	Ereditarietà multipla è permessa con le interfacce e le classi astratte.
9.6	Il main può accedere a qualunque variabile di istanza della classe in cui è
	contenuto.
9.7	Di default l'operatore = e il metodo equals fanno la stessa cosa.
9.8	Il costruttore di una classe può non essere visibile all'esterno della classe stessa.