

D

NOME, COGNOME	
NUMERO DI MATRICOLA	

Istruzioni: leggere il codice dei test sui fogli allegati.

Indicare la risposta sul presente foglio, cerchiando la voce A, B o C. Se si prevede un errore indicare la riga e riportare la motivazione nel campo libero. Se si prevede una corretta esecuzione del codice riportarne l'output nel campo libero.

TEST 1	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 2	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 3	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 4	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 5	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 6	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

TEST 7	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è →	

Test 8.1	Vero o falso? →		Test 8.6	Vero o falso? →	
Test 8.2	Vero o falso? →		Test 8.7	Vero o falso? →	
Test 8.3	Vero o falso? →		Test 8.8	Vero o falso? →	
Test 8.4	Vero o falso? →		Test 8.9	Vero o falso? →	
Test 8.5	Vero o falso? →		Test 8.10	Vero o falso? →	

NOTA: NELL'ESERCIZIO 8 LE RISPOSTE SBAGLIATE SOTTRAGGONO PUNTI. SE SIETE IN DUBBIO MEGLIO LASCIARE IN BIANCO!

D

Test 1

Comment [mr1]: Stampa 6 3

01	package uno;
02	import java.util.*;
03	public class A{
04	A(){
05	Collection l = new LinkedList();
06	Collection s = new HashSet();
07	for (int i = 0; i < 12; i++) if (i % 2 == 0) l.add(new Integer(i%3));
08	s.addAll(l);
09	System.out.println(l.size()+" "+s.size());
10	}
11	public static void main (String args[]){
12	new A();
13	}
14	}

Test 2

Comment [mr2]: Stampa X

01	package uno;
02	class B extends C {
03	void x() {System.out.println("X");}
04	}
05	class C {
06	void x() {System.out.println("Y");}
07	}
08	public class A{
09	A() {
10	C x=new B();
11	x.x();
12	}
13	public static void main (String args[]){
14	new A();
15	}
16	}

Test 3

Comment [mr3]: •clone() has protected access in java.lang.Object at line 10 (10:21) (Compile time)

01	package uno;
02	class B {
03	int x=3;
04	B(int x) {this.x=x;}
05	}
06	public class A {
07	void f() {
08	try {
09	B b1 = new B(5);
10	B b2 = (B) b1.clone();
11	b2.x++;
12	System.out.println(b1.x + " " + b2.x);
13	} catch (Exception e) {e.printStackTrace();}
14	}
15	public static void main(String string[]) {
16	A a=new A();
17	a.f();
18	}}

D

Test 4

00	#include <iostream.h>
01	int x=4;
02	int f(int * x) {
03	x++;
04	return (--(*x));
05	}
06	void g(int & y) {
07	--y;
08	}
09	int main(){
10	int y[]={3,2,1};
11	cout << f(&y[1]);
12	g(x);
13	cout << x << *y << y[1] << *(y+2);
14	}

Comment [mr4]: 0 3 3 2 0

Test 5

01	package uno;
02	class B {
03	int x=4;
04	B(int x) {this.x=x;}
05	}
06	public class A {
07	A() {
08	B b1=new B(3);
09	B b2=new B(3);
10	System.out.println(b1.equals(b2));
11	}
12	public static void main(String Args[]) {
13	new A();
14	}
15	}

Comment [mr5]: Stampa false

Test 6

01	package uno;
02	public class A{
03	static int f(int x) {x++; return x-1;}
04	private int g(int x) {x--; return x+1;}
05	public static void main(String SString[]) {
06	int z=3;
07	System.out.println(f(z));
08	System.out.println(g(z));
09	}
10	}

Comment [mr6]: •non-static method g(int) cannot be referenced from a static context at line 8 (8:24)

D

Test **7**

01	package uno;
02	public class A {
03	C c=null;
04	public A() {
05	System.out.print("+10");
06	C c1=new C();
07	c=new C();
08	}
09	public static void main(String[] args) {
10	A a = new A();
11	System.gc();
12	new C();
13	a = null;
14	System.gc();
15	}
16	}
17	class C {
18	static int count=0;
19	String name=null;
20	public C() {
21	name="2"++count);
22	System.out.print("++name);
23	}
24	protected void finalize() {
25	System.out.print("--name);
26	}
27	}

Comment [mr7]: Stampa
+10+21+22-21+23-22-23

Test **8** - Riportare sul foglio delle risposte V per le affermazioni vere e F per quelle false

8.1	Se A è padre di B la scrittura A a=new B(); genera errore a runtime
8.2	Se A è padre di B la scrittura B a=new A(); genera errore a runtime
8.3	Se A è padre di B la scrittura B a=new A(); genera errore a compile time
8.4	Se A è padre di B la scrittura A a=new B(); genera errore a compile time
8.5	Se A è padre di B la scrittura A a=(A)(new B()); genera errore a runtime
8.6	Se A è padre di B la scrittura B a=(B)(new A()); genera errore a compile time
8.7	JButton ha il metodo equals()
8.8	JButton ha il metodo clone()
8.9	new JButton().clone(); genera errore a compile time anche se i package necessari sono inclusi
8.10	new JButton().clone(); genera errore a runtime

Comment [mr8]:
F F
V F
F F
V V
V F