




B

Test 1 java A1 

00	public class A1 {
01	public int f(int k){
02	return k;
03	}
04	public static void main(String[] a){
05	A1 instance=new B();
06	int x=3;
07	x=((A1)instance).f(x)+((B)instance).f(x);
08	System.out.println(x);
09	}
10	class B extends A1 {
11	public int f(int k){
12	return -k*2;
13	}

Test 2 java A1 

00	public class A1 {
01	public int f(int k){
02	return -k;
03	}
04	public static void main(String[] a){
05	B instance=new A1();
06	int x=3;
07	x=((A1)instance).f(x)+((B)instance).f(x);
08	System.out.println(x);
09	}
10	class B extends A1 {
11	public int f(int k){
12	return k*2;
13	}

Test 3 java A1 

00	public class A1 {
01	public void changeSign(int k){
02	k=-k;
03	}
04	public float changeSign(float k){
05	return -k;
06	}
07	public static void main(String[] k) {
08	int x=3;
09	changeSign(x);
10	System.out.println(x+" "+changeSign((float) x);
11	}

B

Test 4 java A



01	public class A {
02	void f(int k) { System.out.print(k*3); }
03	}
04	public static void main (String args[]){
05	Object z = new B();
06	if (z instanceof uno.A) ((A) z).f(3);
07	if (z instanceof uno.B) ((B) z).f(2);
08	} }
09	class B extends A{
10	void f(int k) { System.out.print(k*2); }
11	}

Test 5 java A1



01	public class A1 {
02	static int count=0;
03	A1(){ System.out.print("A"++count); }
04	A1(int k){ System.out.print("C"++count); }
05	public void finalize(){System.out.print("Z"++count); }
06	public static void main(String[] a){
07	B instance=new B(3);
08	B instance2=instance;
09	instance.f();
10	instance=null;
11	System.gc(); System.runFinalization();
12	}
13	}
14	class B extends A1{
15	B(int k){ System.out.print("B"++count); }
16	public void finalize(){ System.out.print("W"++count); }
17	public void f(){new A1();}
18	}

Test 6 java Otto



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

B

Test 7



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

Test 8 java Sette



01	import java.util.*;
02	public class Sette {
03	Sette(){
04	Collection<String> a = new ArrayList<String>();
05	Collection<String> b = new HashSet<String>();
06	for (int k=0;k<10;k++) {
07	String s="A"+(k%5);
08	a.add(s);
09	b.add(s);
10	}
11	int count=0;
12	Iterator<String>i=a.iterator();
13	while (i.hasNext()) {
14	String s=i.next();
15	count++;
16	}
17	i=b.iterator();
18	while (i.hasNext()) {
19	String s=i.next();
20	count++;
21	}
22	System.out.println(count);
23	}
24	public static void main(String[] a) {
25	new Sette();
26	}}



Test 9 – scrivere nel campo per l'output del test la sequenza risultante indicando T per le affermazioni vere e F per quelle false

9.1	In una classe ci può essere un solo metodo main
9.2	In un programma ci può essere un solo metodo public static void main(String[] args)
9.3	Java usa solo la heap perché tutta la memoria è allocata dinamicamente con le new()
9.4	Una Interface e una classe Abstract sono la stessa cosa
9.5	Non è possibile istanziare una Collection
9.6	In Java esiste ereditarietà multipla
9.7	Se di un metodo f faccio overriding non è detto che ci siano più "signatures" (firme) di metodi legate al nome f
9.8	Se di un metodo f faccio overloading non è detto che ci siano più "signatures" (firme) di metodi legate al nome f

B

3(5) 3.5

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.

no

TEST 1	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga <u>05</u> perchè →	
	C	il codice esegue correttamente, e l'output è →	

no

TEST 2	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è → <u>12</u>	

✓

TEST 3		il codice esegue correttamente, e l'output è → <u>3 -3</u>
--------	--	--

✓

TEST 4	A	compile error alla riga _____ perchè →	
	B	runtime error alla riga _____ perchè →	
	C	il codice esegue correttamente, e l'output è → <u>6 4</u>	

no

TEST 5		il codice esegue correttamente, e l'output è → <u>B 1 A 2 W 1</u>
--------	--	---

no

TEST 6		il codice esegue correttamente, e l'output è →
--------	--	--

✓

TEST 7		il codice esegue correttamente, e l'output è → <u>Q S 1</u>
--------	--	---

no

TEST 8		il codice esegue correttamente, e l'output è →
--------	--	--

TEST 9	Riportare la sequenza di V e F	<div style="display: flex; justify-content: space-around;"> <div>9.1 <u>V</u></div> <div>9.2 <u>V</u></div> <div>9.3 <u>F</u></div> <div>9.4 <u>F</u></div> <div>9.5 <u>V</u></div> <div>9.6 <u>F</u></div> <div><u>V</u></div> <div><u>F</u></div> </div>
--------	--------------------------------	--

Risposte errate al punto 9 sottraggono punti!