

Anteprima di test

parte teorica 4 Febbraio 15

Data: Thu Feb 19 11:31:51 2015 Punteggi massimi: 42

1. Dynamic Binding Java (8 Punti)

Date le seguenti dichiarazioni:

```

1 class Computer {
2     void setCPU(int l) {
3         System.out.println("C");
4     }
5 }
6 class NoteBook extends Computer {
7     void setCPU(int l) {
8         System.out.println("N");
9     }
10}
11class Tablet extends Computer {
12    void setCPU(short l) {
13        System.out.println("T");
14    }
15}
16...
17Object oc = new Computer();
18Computer cc = new Computer();
19Computer cn = new NoteBook();
20Computer ct = new Tablet();
21short myfreq = 30;

```

Qual è l'input prodotto dalle seguenti istruzioni (errore se pensi ci sia un errore)?

oc.setCPU(myfreq) errore (2 Punti)

cc.setCPU(myfreq) C (1 Punto)

cn.setCPU(myfreq) N (1 Punto)

ct.setCPU(myfreq) C (4 Punti)

2. Passaggio di array in C (7 Punti)

Data la seguente funzione

```

1 void f(int a[]){
2     printf("%d\n",sizeof(a));
3     a = a +1;
4 }
5
6
7 int main(void) {
8     int p[] = {10,20,30};
9     printf("%d\n",sizeof(p));
10    f(p);
11    printf("%d\n",*p);
12    return EXIT_SUCCESS;
13}

```

Qual'è l'output prodotto dalle seguenti istruzioni (in ordine di esecuzione)? Se pensi contenga un errore, scrivi errore

Assumi che un puntatore vale 4 byte come anche un intero (32 bit).

nel main:

printf("%d\n",sizeof(p));12 (1 Punto)

in f:

printf("%d\n",sizeof(a))4 (4 Punti)

nel main di nuovo

printf("%d\n",*p);10 (2 Punti)

3. C++ virtual functions ed ereditarietà (3) (14 Punti)

Date le seguenti classi

```

1 class A{
2 private:
3     void pri(){cout << "A" << endl;}
4 public:
5     virtual void pub(){ cout << "A" << endl;}
6 };

```

```

7
8 class APRI1: private A{
9 public:
10     void pri(){cout << "APRI1" << endl;}
11
12};
13
14class APRI2: private A{
15public:
16     virtual void pub(){ cout << "APRI2" << endl;}
17};
18
19class APUB1: public A{
20public:
21     void pri(){cout << "APUB1" << endl;}
22};
23
24class APUB2: public A{
25public:
26     virtual void pub(){ cout << "APUB2" << endl;}
27};

```

Scrivi l'output delle seguenti coppie di istruzioni. Se pensi ci sia un errore scrivi errore e ignora l'istruzione (solo quella che dà errore).

A a1; a1.pri();errore (1 Punto)
APRI1 a2; a2.pri();APRI1 (1 Punto)
APRI2 a3; a3.pri();errore (1 Punto)
APUB1 a4; a4.pri();APUB1 (1 Punto)
APUB2 a5; a5.pri();errore (1 Punto)
a1 = a2; a1.pub();errore (1 Punto)
a1 = a3; a1.pub();errore (1 Punto)
a1 = a4; a1.pub();A (1 Punto)
a1 = a5; a1.pub();A (1 Punto)
A* p = &a1; p -> pub();A (1 Punto)
p = &a2;p -> pub();errore (1 Punto)
p = &a3;p -> pub();errore (1 Punto)
p = &a4;p -> pub();A (1 Punto)
p = &a5;p -> pub();APUB2 (1 Punto)

4. Overriding/Overloading (4 Punti)

Dato il seguente codice

```

1 class Value {}
2 class SmallValue extends Value {}
3
4 class Elaboratore {
5     Value getVal() {
6         return new Value();
7     }
8 }
9
10class Phone extends Elaboratore {
11     SmallValue getVal() {
12         return new SmallValue();
13     }
14}

```

Quali di queste affermazioni sono giuste?

- Phone fa overriding del metodo getVal di Elaboratore *(Selezionato = 1 Punto, Non selezionato = 0 Punti)*
- Phone è una sottoclasse di Elaboratore *(Selezionato = 1 Punto, Non selezionato = 0 Punti)*
- Phone contiene un errore: non può definire getVal in questo modo! *(Selezionato = 0 Punti, Non selezionato = 1 Punto)*
- Phone fa overloading del metodo getVal di Elaboratore *(Selezionato = 0 Punti, Non selezionato = 1 Punto)*

5. overriding di equals (1) (5 Punti)

Data la seguente classe e gli oggetti definiti come segue

```

1 public class A {
2     String name;
3
4     public A(String s) {
5         name = s;
6     }
7
8     public boolean equals(A a) {

```

```
9     return this.name.equals(a.name);
10 }
11}
12
13String pippo = "pippo";
14Object o = new A(pippo);
15A a1 = new A("pippo");
16A a2 = new A(pippo);
```

Quanto valgono (metti errore se pensi ci sia un errore)?

```
pippo.equals(o) false (1 Punto)
o.equals(o) true (1 Punto)
o.equals(a1) false (1 Punto)
o.equals(a2) false (1 Punto)
a1.equals(a2) true (1 Punto)
```

6. overriding di equals (4 Punti)

Data la seguente classe e gli oggetti definiti come segue

```
1 public class A {
2     int x;
3
4     public A(int s) {
5         x = s;
6     }
7
8     public boolean equals(A a) {
9         return this.x == a.x;
10    }
11}
12
13Object o = new A(1);
14A a1 = new A(1);
15A a2 = new A(1);
```

Quale è l'output prodotto dalle seguenti istruzioni (metti errore se pensi ci sia un errore)?

```
System.out.println(o.equals(a1));false (1 Punto)
System.out.println(a1.equals(o));false (1 Punto)
System.out.println(a1.equals(a1));true (1 Punto)
System.out.println(a1.equals(a2));true (1 Punto)
```