

I Compitino 2006- Informatica III.

NOME _____

Concetti Object Oriented *

1. Spiega la differenza tra sottotipazione ed ereditarietà in OO.
2. Che differenza c'è tra le interfacce e le classi astratte in Java?

3. Dinamic Binding in Java ***

Quale è l'output prodotto da: (se c'è un errore in compilazione, segnalalo, ignora l'istruzione e procedi). Spiega, almeno nei casi critici, come la JVM procede nella scelta del metodo da eseguire (early binding e late binding).

```
class A{
    void foo(A x){System.out.println("A");}
class B extends A{
    void foo(B y){System.out.println("B");}
class C extends B{
    void foo(A x){System.out.println("C");}}
```

```
A a1 = new A();
A a2 = new B();
A a3 = new C();
B b4 = new B();
C c5 = new C();
a1.foo(a1); a1.foo(a2); a2.foo(a1);
a2.foo(a2); a2.foo(a3); a2.foo(b4);
b4.foo(a2); b4.foo(b4); c5.foo(a3);
c5.foo(c5); a3.foo(a1);
```

4 Visitor pattern **

Quando è utile il visitor pattern? Supponi di avere le classi `Aula` con due sottoclassi `Laboratorio` e `AulaMagna` e voler implementare l'operazione di calcolo dei posti a sedere della singola aula mediante il visitor pattern, come faresti?

Record attiv.one per funzioni ***

Dato il seguente codice

```
int fun(int x){
    int z = 1;
    if (x > 0) return fun(x - z);
```

```
else return z;}
```

5. Disegna i record di attivazione che si ottengono con la chiamata `fun(2)`, al momento in cui `x` è 0.
6. Si potrebbe ottimizzare utilizzando la tail recursion? Se sì, come?

Passaggio parametri ***

7. Quale è l'output del seguente codice:

```
int a = 0; int b = 5; int c = 3; int d = 7;
void fun(int c, int& e){
    int d = 6;
    c++; d--; e = e*2;
}
void main(){
```

```
    fun(a,b); cout <<a<<b<<c<<d;}
```

8. Dato il seguente codice (supponi x e $y > 0$) che calcola la somma $x + y$.

```
sum(x:int, y:int, z:int){
    z = 0;
    while (x>=0) { z++; x --;}
    while (y>=0) { z++; y --;}
}
```

Per quali tipi di passaggi di parametri la chiamata `sum(a,a,b)` assegna a `b` effettivamente il valore `a+a`?

9. Eccezioni – Java **

Dato il seguente metodo

```
private void foo(int x){
try {
    int y = 7/x;
    System.out.println("A");
} catch (IOException e){
System.out.println("B");
} catch (Exception e){ System.out.println("C");
} finally{ System.out.println("D");} }
```

Quale è l'output prodotto dalle chiamate:

```
foo(9); foo(0);
```

Type Safety Cyclone ***

10. Cosa vuol dire che un linguaggio di programmazione fa il type checking a run time o a compile time? Quali sono i pro e i contro?

11. Elenca i casi (con piccoli esempi) in cui il C non è safe e mostra come ad esempio cyclone risolve il problema