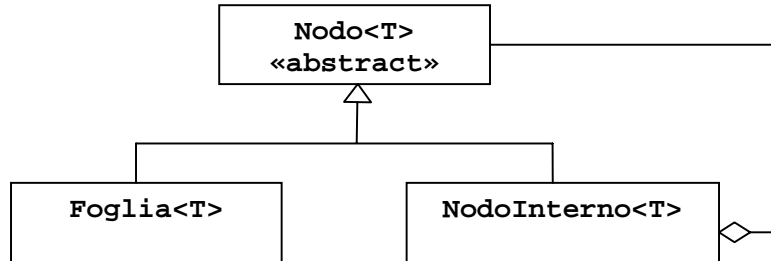


Metodologie di Programmazione AA 2008/ 2009 Secondo Quiz

Esercizio 1. Sia dato il seguente schema UML



Progettate le classi del diagramma, seguendo in modo preciso le specifiche fornite.

Nodo<T>: un nodo ha un campo informazione, privato, di tipo T ed i metodi seguenti:

- // costruisce un nodo con campo informazione uguale a info
protected Nodo(T info);
- // restituisce la rappresentazione testuale del nodo;
public String toString();

Foglia<T>: eredita la struttura ed i metodi dalla superclasse. Definisce il seguente costruttore

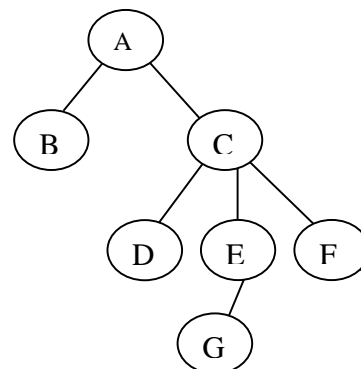
- // costruisce una Foglia con campo informazione uguale a info
public Foglia(T info);

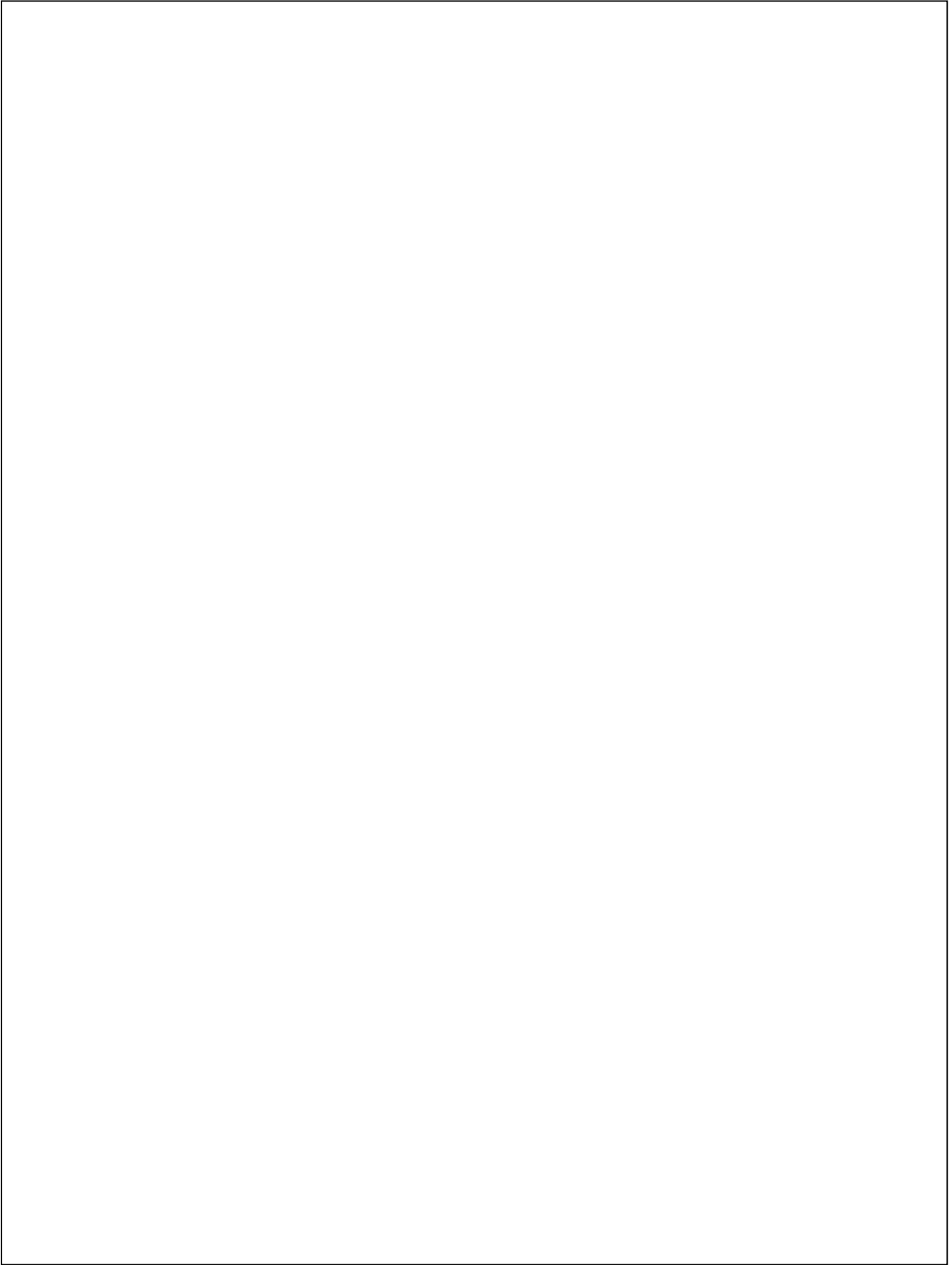
NodoInterno<T>: eredita il campo informazione dalla superclasse e definisce un nuovo campo che contiene una lista con i figli del nodo (dove ciascun figlio può essere una foglia o un nodo interno).. Definisce i seguenti metodi

- // costruisce un NodoInterno con campo informazione info
// e una lista di figli vuota
public NodoInterno(T info);
- // costruisce una NodoInterno con campo informazione info
// e lista di figli che contiene gli elementi nella lista nodi
public NodoInterno(T info, List<Nodo<T>> nodi);
- // sovrascrive il metodo toString() nella superclasse restituendo
// la rappresentazione testuale del campo info seguita dalla rappr.
// testuale della lista dei figli preceduta da "(" e seguita da ")"
public String toString();

Ad esempio l'albero raffigurato qui a destra contiene nodi interni con campo info A, C, E, e foglie B, D, F, G.

L'invocazione del metodo `toString()` sul nodo etichettato da A restituisce la stringa `A (B C (D E (G) F))`





Esercizio 2 Implementate una classe test per la gerarchia di classi realizzata nell'Esercizio 1. Il metodo `main()` utilizza le classi della gerarchia per costruire un albero con la struttura raffigurata in precedenza, e ne stampa la rappresentazione testuale.

Esercizio 3 Considerate le seguenti dichiarazioni:

```
interface I { . . . }
interface J { . . . }
class C implements I, J { . . . }
class D implements I { . . . }
```

Assumete date le seguenti dichiarazioni

```
I i = new D()
J j = new C()
```

Quali dei seguenti comandi causano un errore di compilazione? Quali invece causano un errore a tempo di esecuzione ? (NB: considerate ciascun comando indipendentemente dai precedenti e/o successivi)

	Errore di compilazione	Errore in Esecuzione	Nessun errore
<code>C c = (C) i;</code>			
<code>j = (C) j;</code>			
<code>D d = (C) j;</code>			
<code>i = (C) j</code>			
<code>j = (I) null;</code>			

Esercizio 4. Considerate le seguenti definizioni di classe.

```
class A
{
    public void test(int n) { print(n); }
    public void print(double d) { System.out.println("A"); }
}

class B extends A
{
    public void print(int n) { System.out.println("B.int"); }
    public void print(double d) { System.out.println("B.double"); }
}
```

Cosa stampa la sequenza di istruzioni seguenti? Giustificate la risposta descrivendo le fasi di selezione statica e dinamica che determinano il risultato.

```
A a = new B(); a.test(1); a.test(1.0);
a = new A(); a.test(1); a.test(1.0);
```