

Programmazione a Oggetti  
Modulo B  
PROVA II Compitino  
28 Aprile 2014

Nome: \_\_\_\_\_

Matricola: \_\_\_\_\_

Programmazione a Oggetti 12 crediti [ ] I Compitino+II compitino

**Istruzioni**

- Scrivete in modo leggibile nome e matricola sul primo foglio.
- Scrivete le soluzioni nello spazio riservato a ciascun esercizio e **non** sui fogli quadrettati.

LASCIATE IN BIANCO:

1	2	3	4	TOT



## 1 Esercizio

Dato il seguente codice disegnate il relativo diagramma degli oggetti *creati dal main*.

(24)

```
public class List<T> {  
  
    public class Element<T> {  
  
        Element<T> next;  
        T info;  
  
        public Element(T t) {  
            info = t;  
        }  
    }  
    Element<T> first = null;  
  
    public void add(Element<T> e) {  
        e.next = first;  
        first = e;  
    }  
  
    public static void main(String[] args) {  
        List<String> list = new List<String>();  
        list.add(list.new Element<String>("zero"));  
        list.add(list.new Element<String>("uno"));  
        list.add(list.new Element<String>("due"));  
        //disegnate il diagramma degli oggetti in questo punto  
    }  
}
```



## 2 Esercizio

[2pt]

Disegnate il diagramma degli stati della classe Camera dal codice seguente.

```
public class Camera {  
  
    enum Stato { libera, prenotata, occupata, pulizia};  
    Stato stato=Stato.pulizia;  
  
    public void prenotazione() {  
        if (stato == Stato.libera) {  
            stato = Stato.prenotata;  
        }  
    }  
  
    public void arrivo(Persona p) {  
        if (stato == Stato.libera || stato == Stato.prenotata) {  
            stato = Stato.occupata;  
        }  
    }  
  
    public void annulla() {  
        if (stato == Stato.prenotata) {  
            stato = Stato.libera;  
        }  
    }  
  
    public void libera() {  
        if (stato == Stato.occupata) {  
            stato = Stato.pulizia;  
        }  
    }  
  
    public void pulita() {  
        if (stato == Stato.pulizia) {  
            stato = Stato.libera;  
        }  
    }  
}
```



### 3 Esercizio

Dato il seguente codice diviso nei relativi file:

- illustrare l'applicazione del Design Pattern Type Object utilizzando gli opportuni diagrammi UML; [2pt]
- scrivere l'implementazione relativa. [2pt]

```
public abstract class Animale {
    public abstract String verso();
}
public class Cane extends Animale{
    @Override
    public String verso(){
        return "bau";
    }
}
public class Gatto extends Animale {
    @Override
    public String verso(){
        return "miau";
    }
}
public class Topo extends Animale {
    @Override
    public String verso(){
        return "squit";
    }
}
public class Main {
    public static void main(String [] args){
        Topo topolino = new Topo();
        Cane pluto = new Cane();
        Gatto Silvestro = new Gatto();
    }
}
```



#### 4 Esercizio

- Che pattern è stato rappresentato nel seguente diagramma?
- Scrivete i nomi delle classi e i tipi dei parametri dei metodi.

[1pt]

[2pt]

