

# Metodologie di Programmazione 2006 – 2007

## PRIMO APPELLO: 30 Gennaio 2007

### Istruzioni

- Scrivete il vostro nome sul primo foglio.
- Scrivete le soluzioni nello spazio riservato a ciascun esercizio.
- Tempo a disposizione 2 ore e 30.
- No libri, appunti o altro.

LASCIATE IN BIANCO:

JAVA	
DATATYPES	
ITERATORI	
SUBTYPING	
TOTALE	



## Parte I – Java

Considerate la seguente gerarchia di classi:

```
interface M { M m(); }
interface N { void n(); }

class A implements M {
    public M m() { return this; }
}
class B extends A {
    public void k() { }
}
class C extends A implements N {
    public void n() {}
    public void p() {}
}
```

Quale è il risultato della compilazione e della (eventuale, nel caso la compilazione non dia errori) esecuzione dei seguenti frammenti? **Motivate le risposte**

1. `N x = new C(); M y = x.m();`

2. `M x = new A(); B y = (B)x.m();`

3. `A x = new C(); ((C)x).p();`



Considerate le seguenti classi.

```
class A {  
    void test(double x)  
    { System.out.print("A"); }  
}  
class B extends A {  
    void test(double x)  
    { System.out.print("B-double"); }  
    void test(int x)  
    { System.out.print("B-int"); }  
}
```

Data la dichiarazione `B b = new B()`, quale è il risultato della compilazione e della eventuale esecuzione dei seguenti frammenti? **Motivate le risposte**

4. `((A)b).test(1)`

5. `b.test(1.0)`

6. `b.test(1)`



## Parte II – Datatypes

Sia data la seguente specifica della classe `CircList` che realizza il tipo di dato *Lista Circolare*, ovvero una lista in cui il successore dell'ultimo elemento è nuovamente il primo elemento della lista).

```
class CircList {
// OVERVIEW: una CircList e' una lista circolare di Integers.
// Elemento tipico = [x1,...,xn,x1,...,xn, ....]

public CircList()
// POST: costruisce una CircList vuota

public Integer first() throws EmptyException
// POST: se this e' vuota solleva EmptyException, altrimenti
// restituisce il primo elemento di this

public boolean empty()
// POST: se this e' vuota ritorna true, altrimenti ritorna false.

public void insert (Integer x) throws NullPointerException
// POST: se x e' null solleva NullPointerException, altrimenti
// modifica this aggiungendo x come primo elemento

public Integer delete() throws EmptyException
// POST se this e' vuota solleva EmptyException, altrimenti rimuove il
// primo elemento di this, e lo restituisce

public void rightRotate() throws EmptyException;
// EFFECTS: se this e' vuota solleva EmptyException, altrimenti
// modifica this ruotando la lista a destra. Ovvero,
// se this = [x1,...,xn,x1,...,xn,...],
// this_post = [xn,x1,...,xn-1, xn,x1,...,xn-1,...]

public void leftRotate() throws EmptyException;
// EFFECTS: se this e' vuota solleva EmptyException, altrimenti
// modifica this ruotando la lista a sinistra. Ovvero,
// se this = [x1,...,xn,x1,...,xn,...],
// this_post = [x2,...,xn,x1, x2,...,xn,x1,...]
}
```

Completate la definizione della classe `CircList`, ovvero:

- definite la rappresentazione della classe utilizzando una struttura a lista con nodi semplici
- fornite la funzione di astrazione e l'invariante di rappresentazione
- definite l'implementazione del costruttore e di tutti i metodi nella specifica





### Parte III – Iteratori

Estendete la definizione della classe `CircList` con i due iteratori seguenti:

```
public Iterator elements()  
// POST: restituisce un iteratore su this che produce la sequenza  
// finita degli elementi del primo ciclo della lista. Ovvero, se  
// this = [x1,...,xn,x1,...,xn,...] l'iteratore produce la  
// sequenza [x1,...,xn]
```



## Parte IV – Subtyping

Completate seguente definizione del sottotipo `MinCircList` di `CircList`.

```
class MinCircList extends CircList {  
    // OVERVIEW: una CircList con un metodo min() che determina  
    // il minimo intero nella lista.  
  
    public Integer min() throws EmptyException  
        // POST: restituisce il minimo elemento contenuto nella lista  
}
```

La vostra implementazione deve garantire una complessità costante per `min()`, ridefinendo quindi i metodi `insert()` e `delete()` della superclasse.

