

COGNOME:

NOME:

MATRICOLA:

Secondo Compitino di Programmazione, 8 marzo 2007 – INFORMATICI E MULTIMEDIALI

Esercizio 1 Si progetti una classe di nome `ZeroUno`, la quale istanzia oggetti che modellano successioni formate da zero e uno. La classe mantiene la successione in un campo stringa privato (già dato nel testo) e fornisce:

- un costruttore che, invocato senza parametri, costruisce un oggetto che rappresenta una successione vuota (il costruttore è già dato nel testo);
- un costruttore che, ricevuto un parametro `s` di tipo `String`, costruisce un oggetto che rappresenta la più grande successione di zero e uno contenuta in `s` nell'ordinamento dato dalla stringa (il costruttore è già dato nel testo);

Si aggiunga alla classe:

1. un costruttore che, ricevuto un parametro `n` di tipo `int`, costruisce un oggetto che rappresenta una successione lunga `n` di zero e uno casualmente distribuiti;
2. un metodo `toString()` che restituisce la stringa che rappresenta la successione di zero e uno in cui ogni termine è seguito da virgola: se, ad esempio, la successione è 0100 allora la stringa che la rappresenta sarà "0,1,0,0,";
3. un metodo `toArray()` che restituisce un array di `boolean` in cui l'elemento di indice `i` è `true` o `false` a seconda che il termine `i`-esimo della successione (numerata partendo da zero) sia rispettivamente uguale a uno o zero;
4. un metodo `somma()` che ricorsivamente somma i termini della successione e restituisce un intero contenente la somma. Se ad esempio la successione è 001100010 allora `somma()` restituirà il valore 3.

La classe inoltre contenga un metodo `main()` il quale, costruiti due oggetti `ZeroUno` rispettivamente adoperando una stringa e un intero come parametri per la creazione delle successioni a essi associate, dà successivamente la rappresentazione in stringa e la somma di entrambe le successioni.

```
import prog.io.ConsoleInputManager;  
import prog.io.ConsoleOutputManager;
```

```
public class ZeroUno {  
  
    /////////////////////////////////////////////////// GIA' FATTO  
  
    private String zerouno;  
  
    public ZeroUno () {  
        zerouno = "";  
    }  
  
    public ZeroUno (String s) {  
        this();  
        for (int i=0; i<s.length(); i++)  
            if (s.charAt(i)=='0' || s.charAt(i)=='1')  
                zerouno += s.charAt(i);  
    }  
  
    /////////////////////////////////////////////////// SVOLGERE DA QUI SU FOGLIO PROTOCOLLO
```

(continua)

Esercizio 2 Si consideri la seguente classe per le liste:

```
import prog.io.ConsoleOutputManager;

public class List {
    private int head;
    private List tail;

    public List(int head, List tail) {
        this.head = head;
        this.tail = tail;
    }

    public List swap() {
        if (tail == null) return this;
        else if (head > tail.head)
            return new List(tail.head,new List(head,tail.tail).swap());
        else
            return new List(head,tail.swap());
    }

    public String toString() {
        if (tail == null) return "" + head;
        else return head + "," + tail.toString();
    }

    public static void main(String[] args) {
        List l = new List(5,new List(8,new List(3,new List(5,null))));
        ConsoleOutputManager schermo = new ConsoleOutputManager();

        schermo.println(l);
        l = l.swap();
        schermo.println(l);
        l = l.swap();
        schermo.println(l);
        l = l.swap();
        schermo.println(l);
    }
}
```

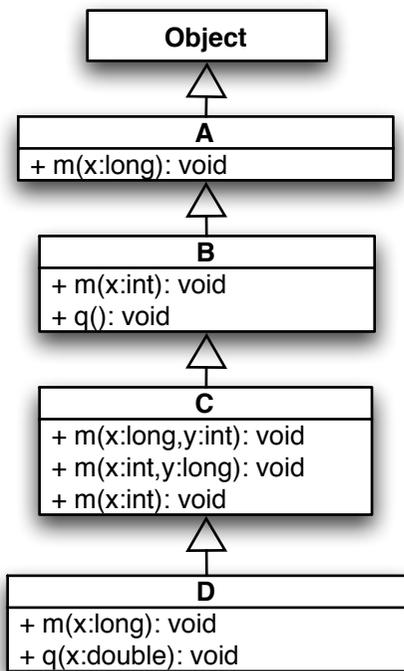
a) Cosa stampa il metodo main() ?

b) Il metodo swap() può generare una NullPointerException? Giustificare la risposta.

Esercizio 3 (Segue alla pagina successiva)

(continua)

Le classi A, B, C, D, E, F, G sono tutte public e sono tutte nella stessa directory



```
public class E{
public static void main(String[] p){
    A alpha = new A();
    A beta = new B();
    B delta = new B();
    alpha.m(5L);
    beta.m(3);
    delta.m(3);
}}
```

```
public class F{
public static void main(String[] p){
    B gamma = new D();
    C csi = new C();
    csi.m(2,2);
    gamma.q(2.3);
}}
```

```
public class G{
public static void main(String[] p){
    B eta = new D();
    B omega = new C();
    D tau = new D();
    eta.m(2L);
    omega.m(2);
    tau.q();
}}
```

Per ognuna delle classi E, F, G si indichi:

- (1) le segnature dei metodi individuate durante la compilazione, indicando gli eventuali errori di compilazione;
- (2) i metodi scelti dalla JVM in esecuzione indicando per ogni metodo la segnature e la classe di appartenenza.

Compilazione di E: elenco delle segnature individuate/ o descrizione errori di compilazione
alpha.m(5L);

beta.m(3);

delta.m(3);

Eventuale esecuzione di E: elenco dei metodi scelti dalla JVM (segnatura+classe appartenenza)
alpha.m(5L);

beta.m(3);

delta.m(3);

Compilazione di F: elenco delle segnature individuate/ o descrizione errori di compilazione
csi.m(2,2);

gamma.q(2.3);

Eventuale esecuzione di F: elenco dei metodi scelti dalla JVM (segnatura+classe appartenenza)
csi.m(2,2);

gamma.q(2.3);

Compilazione di G: elenco delle segnature individuate/ o descrizione errori di compilazione
eta.m(2L);

omega.m(2);

tau.q();

Eventuale esecuzione di G: elenco dei metodi scelti dalla JVM (segnatura+classe appartenenza)
eta.m(2L);

omega.m(2);

tau.q();