Esercizi sulla ricorsione

Esercizio 1

Scrivere un metodo di classe **ricorsivo** che riceve in input una stringa **s** ed altri parametri se necessario, e restituisce il carattere più "alto" nell'ordinamento lessicografico. Ad esempio, se la stringa fosse "AcWbL", il carattere restituito sarebbe 'c' essendo esso il carattere che segue tutti gli altri nell'ordinamento lessicografico. **Scrivere inoltre un metodo per l'avvio della ricorsione.**

Note: *(1) Una soluzione non ricorsiva non sarà considerata corretta (2) Il metodo non deve utilizzare variabili che non siano locali*

Esercizio 2

Scrivere un metodo di classe **ricorsivo** che riceve in input una stringa **s** ed altri parametri se necessario, e restituisce la stringa ottenuta da **s** rimuovendo le lettere maiuscole. Se, ad esempio, la stringa **s** fosse "AlbA", la stringa restituita sarebbe "lb". **Scrivere anche un metodo per l'avvio della ricorsione.**

Note: *(1) Una soluzione non ricorsiva non sarà considerata corretta (2) Il metodo non deve utilizzare variabili che non siano locali*

Esercizio 3

Scrivere un metodo di classe **ricorsivo** che riceve in input un array di interi **a** ed altri parametri se necessario, e restituisce il numero di elementi che sono uguali al proprio successore. Ad esempio, se l'array **a** fosse il seguente

0	1	2	3	4	5	6	7
5	2	2	3	4	4	4	6

il metodo dovrebbe restituire 3; infatti gli elementi che sono uguali al proprio successore sono quelli in posizione 1, 4 e 5. **Scrivere anche un metodo per l'avvio della ricorsione.**

Note: *(1) Una soluzione non ricorsiva non sarà considerata corretta (2) Il metodo non deve utilizzare variabili che non siano locali*

Esercizio 4

Scrivere un metodo di classe **ricorsivo** che riceve in input una matrice quadrata **a** ed altri parametri se necessario, e restituisce **true** se tutti gli elementi della diagonale principale di **a** sono tutti uguali a 0. **Scrivere inoltre un metodo per l'avvio della ricorsione.**

Note: *(1) Una soluzione non ricorsiva non sarà considerata corretta (2) Il metodo non deve utilizzare variabili che non siano locali*

Esercizio 5

Si consideri il seguente metodo della classe ***Esercizio***:

```
public static void esercizio2(int[] a, int i){

      if(i<a.length){
            System.out.println(a[i]);
            esercizio2(a,2*i);
            esercizio2(a,2*i+1);
      }
```

```
}
```

Dire che cosa stampa il seguente frammento di codice:

…

```
int[] a={0,1,2,3,4,5,6,7};
Esercizio.esercizio2(a,1);
```

Esercizi su ereditarietà e polimorfismo

Esercizio 6

Si assuma di aver definito le seguenti classi.

```
public class Vertebrato{
	private String nome;

	public Vertebrato(String nome){this.nome = nome;}

	public String getNome(){return this.nome;}

	public String toString(){return "Nome: "+this.nome;}
}

public class Uccello extends Vertebrato{
	private boolean volatile;

	public Uccello(String nome, boolean volatile){
		super(nome);
		this.volatile = volatile;
	}

	public boolean isVolatile(){return this.volatile;}

	public String toString(){
		return "Nome: "+this.getNome()+", volatile: "+this.volatile;
	}
}

public class Mammifero extends Vertebrato{
	private boolean marino;

	public Mammifero(String nome, boolean marino){
		super(nome);
		this.marino = marino;
	}

	public int isMarino(){return this.marino;}

	public String toString(){
		return "Nome: "+this.getNome()+", marino: "+this.marino;
	}
}
```

Dire, nel seguente frammento di codice, quali istruzioni sono errate spiegando brevemente il perché. Per le istruzioni di stampa, qualora siano corrette, dire che cosa viene stampato a video.

```
Vertebrato v = new Mammifero("Balena", true);
Uccello u=new Uccello("Struzzo", false);
System.out.println(v.getNome());
```

```
System.out.println(v.isMarino());
System.out.println(v.toString());
System.out.println(u.toString());
```

Esercizio 7

Si considerino una interface **I** ed una classe **C** definite come segue.

```
interface I{
      public void a();
}
public class C implements I{
      public void a(){...}
      public void b(){...}
}
```

Dire quale dei seguenti frammenti di codice è corretto e quale no.

	Corretto	Non corretto
`I i=new I();`		
`I i=new C();`		
`C c=new C();` `c.a();`		
`I i=new C();` `i.b();`		
`I i=new C();` `i.a();`		

Esercizio 8

Si assuma di avere a disposizione la seguente interface e le seguenti classi che la implementano.

```
public interface Figura{
      public int perimetro();
}

public class Quadrato implements Figura{
      private int lato;

      public Quadrato(int l){ this.lato=l; }
      public int perimetro(){ return this.lato*4;}
}

public class Cerchio implements Figura{
      private int raggio;

      public Cerchio(int r){ this.raggio=r; }
      public int perimetro(){ return this.raggio*2*Math.PI;}
}
```

Scrivere un metodo **`perimetriFigure`** che permetta di stampare il perimetro di un insieme di quadrati e cerchi. L'insieme può contenere un numero qualsiasi (anche 0) di quadrati e un numero qualsiasi (anche

0) di cerchi. Il metodo **perimetriFigure** inoltre deve funzionare senza modifiche anche nel caso in cui si definisca una nuova classe che implementa l'interface **Figura**.

Esercizio 9

Il seguente codice contiene cinque errori. Indicarli spiegando brevemente in che cosa consiste l'errore.

```
public interface MyInterface{
      private int x;
      public MyInterface(int x);
      public int metA();
      public double MetB();
      private void metC();
}

public class MyClass implements MyInterface{
      private int z;
      public MyClass(int z){
            this.z=z;
      }
      public int metA(){
            return z*2;
      }
}

public class MyExtendedClass extends MyClass{
      private double f;

      public MyExtendedClass (int z, double f){
            this.f=f;
            super(z);
      }
      public double metD(){
            return f/2;
      }

}
```

Esercizi sulla pila di attivazione

Esercizio 10

Supponendo di aver definito la seguente classe **Esempio**, mostrare l'evoluzione della pila di attivazione a seguito dell'avvio del programma **Esempio**.

```
public class Esempio{
  public static int metA(int x){
    int a=0;          // metA, 1
    if(x>0)           // metA, 2
        a=metB(x-1);  // metA, 3
    return a;         // metA, 4
```

```
  }

  public static int metB(int y){
    int b=2;          // metB, 1
    if(y>0)           // metB, 2
        b=metA(y-1);  // metB, 3
    return b+2;       // metB, 4
  }

  public static void main(String[] args){
    int a;
    a = metA(3);   // main, 1
  }
}
```

Nella rappresentazione dei record di attivazione si possono omettere lo "stack degli operandi", e il campo "classe".

Esercizio 11

Supponendo di aver definito la seguente classe **Esercizio**, mostrare l'evoluzione della pila di attivazione a seguito dell'avvio del programma **Esercizio**.

```
public class Esercizio{
  public static int metA(int x, int y){
    int p=1;               // metA, 1
    if(y>0)                // metA, 2
        p=x*metA(x,y-1);   // metA, 3
    return p;              // metA, 4
  }


  public static void main(String[] args){
    int a = metA(3,2);   // main, 1
  }
}
```

Nella rappresentazione dei record di attivazione si possono omettere lo "stack degli operandi", e il campo "classe".

Esercizi su argomenti vari

Esercizio 12

Si consideri la seguente classe:

```
public class MyClass{
     private int a;
     private static int b=0;

     public MyClass(int x){
          this.a=x;
          b++;
     }
```

```
	public method(int y){
		int f=y/2;
		return f;
	}
}
```

Indicare le variabili di istanza, le variabili di classe, le variabili locali e i parametri formali (se una variabile rientra in più casi indicarla in entrambi).

variabili di istanza	
variabili di classe	
variabili locali	
parametri formali	

Esercizio 13

Dire cosa stampa il seguente codice

```
public class Esercizio5{
	public static void main(String args[]){
		char[] a={'I','T','A','L','I','A'};
		int n=a.length;
		int i=0;

		while(i<n){
			System.out.print(a[(2*i)%n]);
			i++;
		}
	}
}
```

Esercizio 14

Scrivere ciascuno dei numeri indicati nel formato indicato:

- 623 (base 2)
- -324 (complemento a 2 con 10 bit)
- 892 (base 16)

Esercizio 15

Il seguente codice contiene quattro errori che possono essere errori di logica, di sintassi, o errori a tempo di esecuzione. Indicarli spiegando brevemente perché sono errori.

```
public static double esercizio(double[] a){
	double d;
	i=0;
	while(i<a.length){
		d+=a[i+1];
		i++;
	}
}
```

Esercizio 16

Dire qual è il valore di ciascuna delle seguenti espressioni. Per ognuno dei valori indicati specificare anche qual è il tipo del risultato:

- (‘a’>’b’)&&(2==(1+1)) _____
- ((5/2)==2)||(3!=3) _____
- (2+5)*Math.sqrt(9) _____
- (3+4.0)*(7/3) _____