Esercizio 1. Si consideri il seguente frammento di codice in Java:

```
int[][] a = b;
int i; int j = 0;
int r = 0;
boolean p = false;
for (i=0; i < a.length; i++) {
    j = 0;
    p = false;
    while (j < a[i].length-1 && !p) {
        if (a[i][j] == a[i][j+1]) {
            r++;
            p = true;
        }
        j++;
    }
}
```

Determinare il valore finale delle variabili *j*, *r* e *p* nei seguenti casi:

- 1) $b = \{\{4,7,3,-3,5\}, \{-1,2,5,2,6\}\}$;
- 2) $b = \{\{3,5,4,2\}, \{-1,2,2,3\}, \{7,7,5,5\}, \{1,1,2,-3\}\}$.

Soluzione

- 1) $j = 4, r = 0, p = \text{false}$.
- 2) $j = 1, r = 3, p = \text{true}$.

Esercizio 2. Si consideri il seguente frammento di codice in Java:

```
class Exc1 extends Exception { }

public class C {
    public static int met(int m,int n)
        throws Exc1 {
        try {
            if (m*m < n) return met(m+1,n);
            else throw new Exc1();
        }
        catch (Exc1 e) {
            if (m*m == n) return m;
            else throw e;
        }
    }
}
```

Determinare il valore restituito dalle seguenti chiamate:

- 1) $C.\text{met}(3,10)$;
- 2) $C.\text{met}(2,25)$;
- 3) $C.\text{met}(4,12)$.

Soluzione

- 1) Viene sollevata l'eccezione *Exc1*.
- 2) Viene restituito il valore 5.
- 3) Viene sollevata l'eccezione *Exc1*.

Esercizio 3. Scrivere un metodo che, data una stringa *s*, restituisce un array di stringhe contenente tutti i prefissi di *s* in *ordine crescente* (inclusa *s*). Ad esempio, data la stringa "abcdef", il metodo restituisce l'array {"a","ab","abc","abcd","abcde","abcdef"}.

Soluzione

```
public static String[] prefissi (String s) {
    int n = s.length();
    String[] a = new String[n];
    for (int i=0; i<n; i++)
        a[i] = s.substring(0,i+1);
    return a;
}
```

Esercizio 4. Scrivere un metodo ricorsivo che, dato un array *a* di interi, restituisce la somma alternante di *a*, ovvero il valore ottenuto aggiungendo gli elementi di *a* in posizione pari e sottraendo gli elementi di *a* in posizione dispari. Ad esempio, dato l'array $a = \{7,3,4,-6,-11\}$, il metodo restituisce 3.

Soluzione

```
public static int alt_sum (int[] a, int sum, int i) {
    if (i == a.length) return sum;
    if (i%2 == 0) return alt_sum(a,sum+a[i],i+1);
    return alt_sum(a,sum-a[i],i+1);
}

public static int alt_sum (int[] a) {
    return alt_sum(a,0,0);
}
```

Esercizio 5. Scrivere una classe `EventoMusicale`, i cui oggetti sono caratterizzati dal tipo (Es. rock, jazz, classico), la città ed il luogo in cui si svolge l'evento. Definire un costruttore per gli oggetti della classe ed i seguenti metodi:

- i metodi che restituiscono i valori delle variabili istanza;
- un metodo `descrizioneEvento` che restituisce le informazioni relative ad un evento musicale, per esempio una stringa della forma

Evento musicale rock, Firenze, Piazzale Michelangelo

nel caso di un evento di tipo rock, ed

Evento musicale classico, Roma, Auditorium Parco della Musica

nel caso di un evento di tipo classico.

Soluzione

```
public class EventoMusicale {

    private String tipo, citta, luogo;

    public EventoMusicale (String t, String c, String p) {
        tipo = t;
        citta = c;
        luogo = p;
    }

    public String get_tipo() {
        return tipo;
    }

    public String get_citta() {
        return citta;
    }

    public String get_luogo() {
        return luogo;
    }

    public String descrizioneEvento() {
        return("Evento musicale "+tipo+", "+citta+", "+luogo);
    }
}
```

Esercizio 6. Un concerto rock è un evento musicale caratterizzato dal nome del cantante o gruppo musicale principale e dal nome del cantante o gruppo musicale di supporto. Un concerto di musica classica è un evento musicale caratterizzato dal nome dell'orchestra e dal nome del direttore.

Scrivere due classi `ConcertoRock` e `ConcertoClassico` con gli opportuni costruttori. Tra i metodi delle nuove classi, oltre a quelli che restituiscono i valori delle variabili istanza, definire un metodo `descrizioneEvento` che restituisce le informazioni relative ad un concerto, per esempio Evento musicale rock, Firenze, Piazzale Michelangelo, Radiohead, Low nel caso di un concerto rock, ed Evento musicale classico, Roma, Auditorium Parco della Musica, Orchestra dell'Accademia di S. Cecilia, Luis Bacalov nel caso di un concerto di musica classica.

Soluzione

```
public class ConcertoRock extends EventoMusicale {

private String band1, band2;

    public ConcertoRock (String t, String c, String p, String b1, String b2) {
        super(t,c,p);
        band1 = b1;
        band2 = b2;
    }

    public String get_band1() {
        return band1;
    }

    public String get_band2() {
        return band2;
    }

    public String descrizioneEvento() {
        return(super.descrizioneEvento()+" "+band1+" "+band2);
    }
}

public class ConcertoClassico extends EventoMusicale {

private String orchestra, direttore;

    public ConcertoClassico (String t, String c, String p, String o, String d) {
        super(t,c,p);
        orchestra = o;
        direttore = d;
    }

    public String get_orchestra() {
        return orchestra;
    }

    public String get_direttore() {
        return direttore;
    }

    public String descrizioneEvento() {
        return(super.descrizioneEvento()+" "+orchestra+" "+direttore);
    }
}
```

Esercizio 7. Scrivere una classe `OrganizzazioneEventiMusicali`, i cui oggetti sono caratterizzati dal nome dell'organizzazione, la sede legale, il responsabile e l'insieme di eventi musicali organizzati (il cui numero massimo viene impostato dal costruttore della classe). Definire un costruttore per gli oggetti della classe con le opportune eccezioni in caso di parametri numerici non corretti. Infine, definire (oltre ai metodi che restituiscono i valori delle variabili istanza) i seguenti metodi:

- un metodo che aggiunge un evento musicale (con eventuali eccezioni);
- un metodo che restituisce l'elenco di tutti gli eventi musicali che si svolgono in una data città c.

Soluzione

```
class ExcNeg extends Exception { }
class ExcFull extends Exception { }

public class OrganizzazioneEventiMusicali {

    private String nome, sede, resp;
    private EventoMusicale[] ev;
    private int cont;

    public OrganizzazioneEventiMusicali (String s, String sl, String r, int max)
    throws ExcNeg {
        nome = s;
        sede = sl;
        resp = r;
        if (max > 0)
            ev = new EventoMusicale[max];
        else throw new ExcNeg();
    }

    public String get_nome() {
        return nome;
    }

    public String get_sede() {
        return sede;
    }

    public String get_resp() {
        return resp;
    }

    public EventoMusicale[] get_eventi() {
        return ev;
    }

    public void agg_evento (EventoMusicale e) throws ExcFull {
        if (cont < ev.length) {
            ev[cont] = e;
            cont++;
        }
        else throw new ExcFull();
    }

    public EventoMusicale[] cartellone (String p) {
        int k = 0;
        for (int i=0; i<cont; i++)
```

```
        if (p.equals(ev[i].get_citta())) k++;
EventoMusicale[] a = new EventoMusicale[k];
int j = 0;
for (int i=0; i<cont; i++)
    if (p.equals(ev[i].get_citta())) {
        a[j] = ev[i];
        j++;
    }
return a;
}
}
```