

Esercizio 1.

Implementare l'applicazione per il calcolo di espressioni aritmetiche in notazione postfissa nella forma in cui è illustrata nelle slides della lezione

Esercizio 2.

La classe `String` dispone di un metodo `split`:

```
public String[] split(String regex)
```

che può essere usato per spezzare la stringa in tokens secondo dei delimitatori specificati. Il parametro `regex` è una espressione regolare (sequenza di simboli che identifica un insieme di stringhe – saranno trattate in seguito nel corso). Ad esempio:

```
String[] risultato = "Questo è un test".split("\\s");  
for(int x=0; x<risultato.length; ++x)  
    out.println(risultato[x]);
```

Dove `'\\s'` indica il carattere spazio, produce:

Questo
è
un
test

Utilizzando il metodo `split`, modificare l'applicazione dell'esercizio 1 in modo da leggere tutta l'espressione in input in una singola linea.

Esercizio 3.

Gestire opportunamente tutte le eccezioni nel programma seguente, stampando i messaggi di errore che descrivono il tipo di errore che si è verificato

```
1  import prog.io.ConsoleInputManager;  
2  
3  public class Esercizio1 {  
4      public static void main(String[] args) {  
5          int n = 10;  
6          int[] v = new int[n];  
7  
8          ConsoleInputManager in = new ConsoleInputManager();  
9  
10         int i = 0;  
11         String linea = in.readLine("Inserire un intero: ");  
12         while (!linea.equals("")) {  
13             v[i] = Integer.parseInt(linea);  
14             linea = in.readLine("Inserire un intero: ");  
15             i++;  
16         }  
17     }  
18 }
```