

Esame di Programmazione II, 15 settembre 2014

Una tokenizzazione è un iterabile su stringhe:

```
public interface Tokenization extends Iterable<String> {  
}
```

L'idea è che, iterando su una tokenizzazione, si ottengano, una alla volta, le sottostringhe di una scomposizione di una stringa di partenza *s*, divisa sulla base di un criterio specificato dalla specifica implementazione della tokenizzazione. Per esempio, una tokenizzazione potrebbe usare un insieme di caratteri delimitatori, dopo i quali separare le sottostringhe di *s*.

Esercizio 1 [7 punti] Si scriva un'implementazione `StringTokenization` di `Tokenization`, che divide una stringa *s* subito dopo uno dei caratteri delimitatori forniti al momento della costruzione:

```
public class StringTokenization implements Tokenization {  
    ...  
    public StringTokenization(String s, String delimiters) { ... }  
  
    @Override  
    public Iterator<String> iterator() { ... }  
}
```

Per esempio, iterando su una `new StringTokenization("Questa$e'&una&prova!", "$&")` si devono ottenere una dopo l'altra le stringhe:

```
Questa$  
e'&  
una&  
prova!
```

Se la stringa dei delimitatori è vuota, il costruttore deve lanciare una `NoDelimitersException`, che dovete scrivere.

Esercizio 2 [2 punti] Si scriva un'implementazione `CharacterTokenization` di `Tokenization`, che divide la stringa *s* subito dopo un ben preciso carattere *c* fornito al momento della costruzione:

```
public class CharacterTokenization extends StringTokenization {  
    public CharacterTokenization(String s, char c) { ... }  
}
```

Esercizio 3 [2 punti] Si scriva un'implementazione `SpaceTokenization` di `Tokenization`, che divide la stringa *s* subito dopo uno spazio:

```
public class SpaceTokenization extends CharacterTokenization {  
    public SpaceTokenization(String s) { ... }  
}
```

Esercizio 4 [11 punti] Si scriva un'implementazione `DoubleTokenization` di `Tokenization`, che divide la stringa *s* prima secondo un'altra tokenizzazione fornita al costruttore e poi divide ciascuna di tali stringhe dopo uno dei caratteri delimitatori forniti al costruttore:

```
public class DoubleTokenization implements Tokenization {  
    ...  
    public DoubleTokenization(Tokenization base, String delimiters) { ... }  
  
    @Override  
    public Iterator<String> iterator() { ... }  
}
```

Se tutto è corretto, un'esecuzione del seguente `main`:

```
public class Main {
    public static void main(String[] args) {
        Tokenization tok = new StringTokenizer("Questa$e'&una&prova!", "$&");
        for (String s: tok)
            System.out.println(s);

        Tokenization tok2 = new DoubleTokenization(tok, "ou");
        for (String s: tok2)
            System.out.println(s);
    }
}
```

dovrà stampare:

```
Questa$
e'&
una&
prova!
Qu
esta$
e'&
u
na&
pro
va!
```