

III Appello - 4 giugno 2009

Cognome

Nome

Matricola

1 (4 punti)

Scrivete il programma Java, **Esercizio**, che produce in output in ordine inverso gli argomenti che gli vengono passati sulla riga di comando. Ad esempio:

```
> java Esercizio prova di esame!  
esame!  
di  
prova
```

Produce in output:

2 (3 punti)

Supponete di aver istanziato l'oggetto **v** di classe **Vector** di lunghezza 3, con valori di tipo **String** e che l'output dell'istruzione **println(v)** sia:

```
[oggi, domani, mai]
```

Considerate ora il seguente ciclo, **scorretto**, che dovrebbe premettere a ogni elemento del vettore la stringa **non**

```
for (int i=0; i < v.size(); i++) v.add(i, "non");
```

Se il ciclo funzionasse correttamente l'istruzione **println(v)** dovrebbe dare:

```
[non, oggi, non, domani, non, mai]
```

Putroppo non è così. Individuate il problema e scrivete il codice Java necessario per correggere il ciclo.

3 (3 punti)

Fornite un piccolo esempio concreto di come un metodo può causare l'eccezione **NullPointerException**.

Fornite una concisa spiegazione del perchè **NullPointerException** NON è un'eccezione controllata.

4 (5 punti)

Considerate il seguente spezzone di programma che chiede all'utente di immettere 2 numeri interi e ne calcola la divisione.

```
ConsoleInputManager in = new ConsoleInputManager();  
int n1, n2;  
double r;  
n1 = in.readInt("input un intero: ");  
n2 = in.readInt("input un secondo intero: ");  
r = (double) n1/n2;
```

Inserite il codice in una istruzione **try-catch** con più clausole **catch** in modo da controllare gli errori seguenti:

- tentativo di divisione per 0
- introduzione di dati testuali invece di interi (**InputMismatchException**)

Se queste condizioni si verificano il programma lo segnala all'utente e riprende l'esecuzione con la richiesta di immissione dei dati

5 (6 punti)

Le istruzioni di un programma si sono inspiegabilmente mescolate. Dovete ricostruire l'ordine preciso delle istruzioni perché in esecuzione dia il seguente risultato:

> Java Puzzle Olio

strippa

> Java Puzzle Stanlio

strappa

Ricostruite le righe d'istruzioni nell'ordine corretto. Notate che potrebbero mancare delle parentesi e dei punti e virgola.

```
try {
System.out.print("s");          System.out.print("a");
    System.out.print("i");
} finally {
catch (MiaEccezione e) {
    System.out.print("a");

System.out.print("pp");
throw new MiaEccezione();
    public static void main(String[] arg) {
        String test = arg[0];
rischia(test);
        if "Olio".equals(arg))
            static void rischia(String arg) throws MiaEccezione
            {
                System.out.print("tr");
class MiaEccezione extends Exception{}
    System.out.println()

public class Puzzle {
```

6 (10 punti)

Nella classe **Morse** definite un campo statico di identificatore **morseCode** come array di lunghezza 26 di stringhe:

Il metodo statico **init()** lo inizializzerà secondo la convenzione dell'alfabeto Morse, vedi punto 7 successivo.

Scrivere il metodo statico **String code(String parola)** che codifica i caratteri che compongono la parola passata come argomento secondo l'alfabeto Morse e riportano all'ambiente chiamante la stringa di caratteri codificata. Fate uso della classe **StringBuffer** che offre il metodo **append(String s)** utile allo scopo.

7 Programma da consegnare funzionante alla prova di laboratorio e in stampa all'orale

Scrivete un programma Java che accetta da riga di comando il nome di un file di testo contenente un messaggio da codificare con il codice Morse, che codifica ogni lettera con una sequenza di 2 caratteri, il punto e la linea, ed è archiviato nel campo **morseCode**

A	.-	B	-...	C	-.-.	D	-..	E	.	F	..-	G	--.
H		I	..	J	.---	K	-.-	L	.-..	M	--	N	-.
O	---	P	.---	Q	--.-	R	.-.	S	...	T	-	U	..-
V	...-	W	.-	X	-.-.-	Y	-.-.-	Z	---.				

Il vostro programma deve produrre in output ogni parola del messaggio d'input e la sua codifica Morse su una riga separata. Ad esempio, a fronte di un file di testo contenente le righe:

```
aiuto
siamo naufragando
siamo al largo di pianosa
```

L'output a video dovrebbe essere come segue:

```
aiuto
.-.....
siamo
...-----
naufragando
-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-
siamo
.....
al
.-.-.
largo
.-.-.-.-.-
di
-....
pianosa
.-.-.-.-.-.-.-.-.-.-
File processed.
```

```
public static void init() {
    morseCode[0]= new String(".-");
    morseCode[1]= new String("-...");
    morseCode[2]= new String("-.-.");
    morseCode[3]= new String("-..");
    morseCode[4]= new String(".");
    morseCode[5]= new String("..-");
    morseCode[6]= new String("--.");
    morseCode[7]= new String("....");
    morseCode[8]= new String("..");
    morseCode[9]= new String(".---");
    morseCode[10]= new String("-.-");
    morseCode[11]= new String(".-..");
    morseCode[12]= new String("--");
    morseCode[13]= new String("-.");
    morseCode[14]= new String("---");
    morseCode[15]= new String("...-");
}
```

```
morseCode[16] = new String(" --.-");  
morseCode[17] = new String(" .-.");  
morseCode[18] = new String(" ...");  
morseCode[19] = new String("-");  
morseCode[20] = new String(" .-");  
morseCode[21] = new String(" ...-");  
morseCode[22] = new String(" .--");  
morseCode[23] = new String("-.-");  
morseCode[24] = new String("-.--");  
morseCode[25] = new String("--..");  
}
```

Ovviamente il metodo **String code(String parola)** dell'esercizio precedente fa parte del programma.

Usate la classe `StringTokenizer` di `java.util`. Assumete che il file di testo non contenga segni di punteggiatura, ma solo caratteri alfabetici e spazi vuoti, o meglio che eventuali caratteri di punteggiatura siano presi come caratteri separatori e non codificati.

Si richiede che siano usate le librerie standard di IO per la gestione dei file. Quindi dovrebbero servire le istruzioni d'importazione delle classi:

```
import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.FileReader;  
import java.io.IOException;
```

Si richiede che il ciclo di lettura sia fatto in un'istruzione **try-catch** per controllare eventuali errori di IO e la mancanza del file di input nel file system.