

Wrapper

PariDispari

Scrivi la classe `PariDispari` che valuta il valore intero passato come argomento da linea di comando e, se il valore è pari stampa “PARI”, altrimenti stampa “DISPARI”. Nel caso che il valore passato non sia intero è **necessario** gestire l'eventuale errore in maniera appropriata. Qualche esempio:

```
$ java PariDispari 3  
DISPARI
```

```
$ java PariDispari 6  
PARI
```

```
$ java PariDispari 8  
PARI
```

Se non vengono passati argomenti il programma termina:

```
$ java PariDispari  
$
```

Se vengono passati argomenti non appropriati:

```
$ java PariDispari "Testo"  
ERRORE: il valore Testo non è accettabile
```

```
$ java PariDispari 5.3  
ERRORE: il valore 5.3 non è accettabile
```

Hex2Dec

Scrivi la classe `Hex2Dec` che converte in decimale il valore passato come argomento da linea di comando in formato esadecimale. Nel caso che il valore passato non sia adeguato è **necessario** gestire l'eventuale errore in maniera appropriata. Esempio:

```
$ java Hex2Dec A
10
```

```
$ java Hex2Dec FF
255
```

```
$ java Hex2Dec 10
16
```

Se non vengono passati argomenti il programma termina:

```
$ java Hex2Dec
$
```

Se vengono passati argomenti non appropriati:

```
$ java Hex2Dec "EffeEffe"
ERRORE: il valore EffeEffe non è accettabile
```

```
$ java Hex2Dec 5.3
ERRORE: il valore 5.3 non è accettabile
```

Dec2Hex

Scrivi la classe `Dec2Hex` che converte in esadecimale il valore passato come argomento da linea di comando. Nel caso che il valore passato non sia un valore decimale è **necessario** gestire l'eventuale errore in maniera appropriata. Esempio:

```
$ java Dec2Hex 10  
a
```

```
$ java Dec2Hex 255  
ff
```

Se non vengono passati argomenti il programma termina:

```
$ java Dec2Hex  
$
```

Se vengono passati argomenti non appropriati:

```
$ java Dec2Hex "Cento"  
ERRORE: il valore Cento non è un valore decimale  
  
$ java Dec2Hex 5.3  
ERRORE: il valore 5.3 non è un valore decimale
```

SumArgs

Scrivi la classe `SumArgs` che stampa a terminale la somma dei due valori interi passati come argomenti da linea di comando. Nel caso che i valori passati non siano valori interi **è necessario** gestire l'eventuale errore in maniera appropriata. Esempio:

```
$ java SumArgs 1 5
1 + 5 = 6
```

Se l'utente non inserisce uno o entrambi gli argomenti, i valori omessi vengono considerati come 0 nel calcolo della somma. Ad esempio:

```
$ java SumArgs 2
2 + 0 = 2
```

Oppure:

```
$ java SumArgs
0 + 0 = 0
```

Se vengono passati argomenti non appropriati:

```
$ java SumArgs 3 "Cinque"
ERRORE: uno o più valori passati non sono accettabili

$ java SumArgs "Due" 5.4
ERRORE: uno o più valori passati non sono accettabili
```

SumArgsHex

Scrivi la classe `SumArgsHex` che stampa a terminale la somma dei due valori interi passati come argomenti da linea di comando in formato esadecimale. Nel caso che i valori passati non siano valori interi è **necessario** gestire l'eventuale errore in maniera appropriata. Esempio:

```
$ java SumArgsHex 5 5
5 + 5 = a

$ java SumArgsHex A 5
a + 5 = f

$ java SumArgsHex FF FF
ff + ff = 1fe
```

Se vengono passati argomenti non appropriati:

```
$ java SumArgsHex FF "Cinque"
ERRORE: uno o più valori passati non sono accettabili

$ java SumArgsHex 5.3 FF
ERRORE: uno o più valori passati non sono accettabili
```