

Esercizi metodi

Cerchio

Scrivi la classe `Cerchio` contenente il metodo:

```
public static double calcolaCirconferenza(double raggio)
```

Tale metodo ritorna un numero che rappresenta la circonferenza di un cerchio di raggio `raggio`.

Triangolo

Scrivi la classe `Triangolo` contenente il metodo:

```
public static double calcolaAreaTriangolo(double base, double altezza)
```

Tale metodo ritorna un numero che rappresenta l'area di un triangolo di base `base` e di altezza `altezza`.

DivMul

Scrivi la classe `DivMul` contenente i metodo:

- `public static int calcolaMCD(int a, int b)`: calcola il massimo comun divisore di 2 numeri. (Wiki: MCD)
- `public static int calcolaMcm(int a, int b)`: calcola il minimo comune multiplo di 2 numeri. (Wiki: mcm)

Calcolatrice

Scrivi la classe `Calcolatrice` contenente il metodo:

```
public static int calcola(int a, int b, char operatore)
```

Tale metodo esegue un'operazione matematica di base sui due numeri `a` e `b`. Il valore `operatore` ('+', '-', '*', '/') definisce l'operazione da eseguire.

Ad esempio l'invocazione `calcola(5,2,'+')` ritornerà 7 (5+2) mentre l'invocazione `calcola(5,2,'*')` ritornerà 10 (5*2). Se il parametro `operatore` contiene un carattere non valido, il metodo ritorna 0.

StringTool

Scrivi la classe `StringTool` contenente i metodo seguenti:

- `public static String longest(String a, String b)`: ritorna la stringa più lunga tra `a`, `b`. Ad esempio `longest("cane","gatto")` ritorna la stringa `gatto`, mentre `longest("abcdef","gatto")` ritorna `abcdef`.
- `public static String alternate(String a, String b, int times)`: ritorna una stringa di testo contenente le stringhe `a` e `b` alternate `times` volte. Ad esempio `alternate("","-",5)` ritorna la stringa `*-*-*--*--`, mentre `alternate("va","Ja",2)` ritorna la stringa `vaJavaJa`.
- `public static String concatenate(int a, int b, int c, char separator, char terminator)`: ritorna una stringa contenente l'elenco dei valori dei parametri `a`, `b` e `c`. I valori sono separati dal carattere `separator` e la lista è terminata dal carattere `terminator`. Ad esempio `concatenate(1,2,3,';',',','.')` ritorna la stringa `1;2;3.`, mentre `concatenate(10,20,30,'_','!')` ritorna la stringa `10_20_30!`.

NameFramer

Scrivi la classe `NameFramer` contenente il metodo:

```
public static String frameName(String name, char c, int width)
```

che ritorna una stringa contenente la stringa `name` incorniciata nel carattere `c` utilizzato con uno spessore `width`.

Ad esempio `frameName("Piero",'*',2)` ritorna la stringa:

```
*****  
*****  
**Piero**  
*****  
*****
```

mentre `frameName("Andrea",'$',1)` ritorna la stringa:

```
$$$$$$$  
$Andrea$  
$$$$$$$
```

Trenino

Scrivi la classe `Trenino` che contiene i seguenti metodi:

- `public static void stampaLocomotiva()`
- `public static void stampaPasseggeriPiccolo()`
- `public static void stampaPasseggeriGrande()`
- `public static void stampaMerciPiccolo()`
- `public static void stampaMerciGrande()`
- `public static void stampaGancio()`

Nel metodo `main` utilizza i metodi che hai scritto per comporre un treno. Utilizza a piacimento i vagoni creati tramite i metodi, come ad esempio (`Locomotiva`, `PasseggeriPiccolo`, `MerciGrande`):

```
/T\  
/|||\ \  
| 0 |  
LLLLL  
LLLLL  
|  
PPPPP  
P   P  
PPPPP  
|  
MMMMM  
M   M  
M   M  
M   M  
MMMMM
```