Invocare i metodi

Metodi

- Si usa un oggetto richiamando i suoi metodi.
- Tutti gli oggetti di una data classe condividono un insieme comune di metodi.
- La classe PrintStream fornisce metodi per i suoi oggetti come:
  - println
  - print

Classe String

La classe String fornisce metodi che puoi applicare a tutte le stringhe oggetti.

Esempio: length

```
String saluto = "Hello, World!";
int numberOfCharacters = saluto.length();
```

Esempio: `toUpperCase`

```
String fiume = "Mississippi";
String bigFiume = fiume.toUpperCase();
```

Classe String

La classe String dichiara molti altri metodi oltre alla lunghezza e toUpperCase metodi.

Insieme, i metodi formano l'interfaccia pubblica della classe.

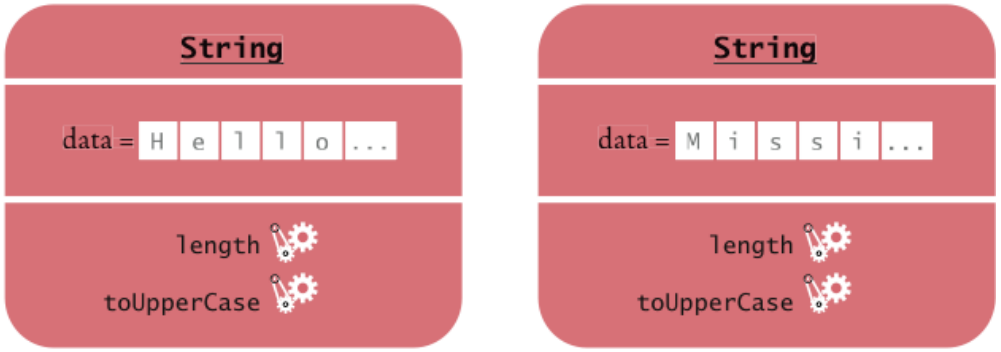
L'interfaccia pubblica di una classe specifica cosa puoi fare con i suoi oggetti.

L'implementazione nascosta descrive come vengono eseguite queste azioni.

Classe String

Ogni oggetto String archivia i propri dati.

Entrambi gli oggetti supportano lo stesso insieme di metodi. Questi metodi formano l'interfaccia pubblica.
L'interfaccia pubblica è specificata dalla classe String.



Argomenti dei metodi

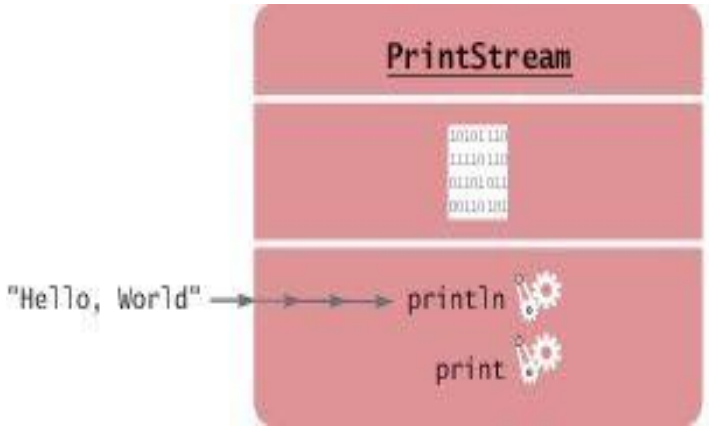
La maggior parte dei metodi richiede valori che forniscano dettagli sul lavoro che il metodo deve svolgere.

Devi fornire la stringa che dove essere stampata quando chiami il metodo println.

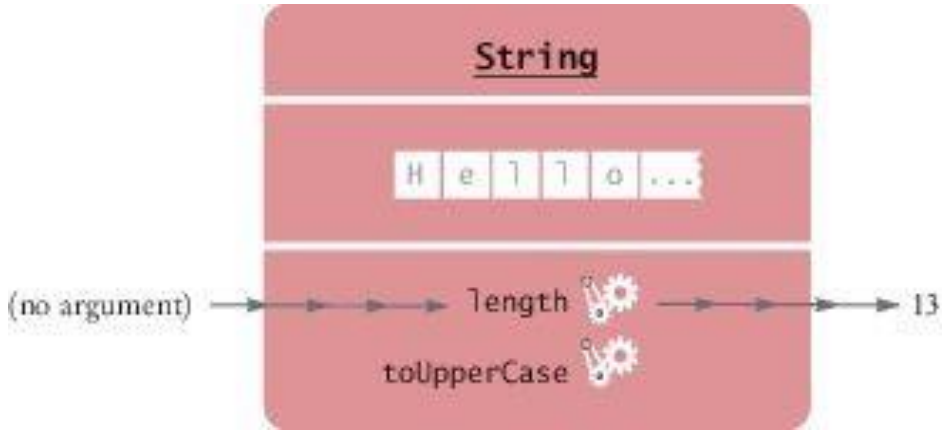
Il termine tecnico per gli input dei metodi: argomenti o parametri.

'saluto' della stringa è un argomento di questa chiamata al metodo:

```
System.out.println(saluto);
```



Argomenti dei metodi



Alcuni metodi richiedono più argomenti.

Altri metodi non richiedono alcun argomento.

Esempio: il metodo length della classe String

Tutte le informazioni che il metodo length richiede per svolgere il proprio lavoro sono memorizzate nell'oggetto

Valori restituiti

Alcuni metodi eseguono un'azione per te.

Esempio: metodo println

Altri metodi calcolano e restituiscono un valore.

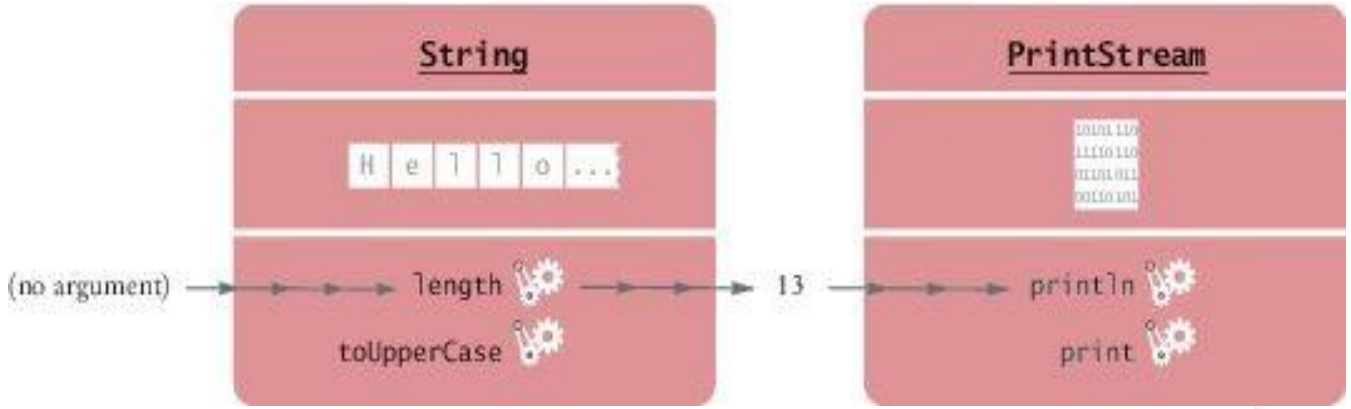
Esempio: il metodo length della stringa restituisce un valore: il numero di caratteri nella stringa.

Puoi memorizzare il valore restituito in una variabile:

```
int numberOfCharacters = saluto.length()
```

Il valore restituito di un metodo è un risultato che il metodo ha calcolato.

Valori restituiti



Puoi anche utilizzare il valore restituito di un metodo come argomento di un altro metodo:

```
System.out.println(saluto.length());
```

La chiamata al metodo saluto.length() restituisce un valore, l'intero 13. Il valore restituito diventa un argomento del metodo println.

Valori restituiti

```
fiume = fiume.replace("issipp", "our");
```

Costruisce una nuova stringa:

sostituendo tutte le occorrenze di "issipp" in "Mississippi" con ”our"

Restituisce l'oggetto String costruito "Missouri"

E salva il valore restituito nella stessa variabile

Potresti passare il valore restituito a un altro metodo:

```
System.out.println(fiume.replace("issipp", "our"));
```

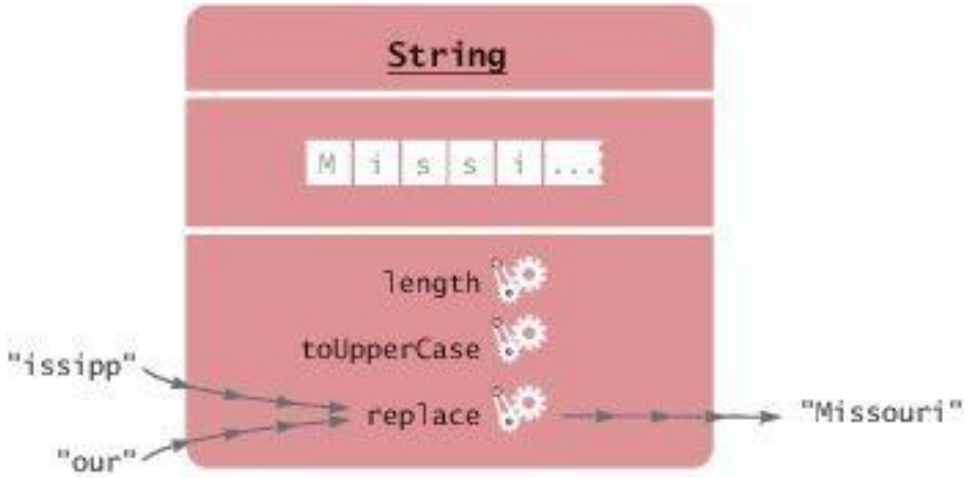
Valori restituiti

```
fiume.replace("issipp", "our"))
```

è invocato su un oggetto String: "Mississippi"

Ha due argomenti: le stringhe "issipp" e "our"

Restituisce un valore: la stringa "Missouri



Dichiarare un metodo

Per dichiarare un metodo in una classe, bisogna specificare:

- i tipi degli argomenti
- il valore restituito

Esempio:

```
public int length()
```

Non ci sono argomenti

Il valore restituito ha il tipo int

Dichiarare un metodo

Esempio:

```
public String replace (String target, String replacement)
```

- Ha due argomenti: target e replacement
- Entrambi gli argomenti hanno il tipo String
- Il valore restituito è un'altra stringa

Esempio:

```
public void println(String output)
```

- Ha un argomento di tipo String
- Nessun valore di ritorno
- Usa la parola chiave void

Dichiarare un metodo

Una classe può dichiarare due metodi con lo stesso nome e diversi tipi di argomento.

La classe PrintStream dichiara un altro metodo println

```
public void println(int output)
```

Usato per stampare un valore intero

Il nome println è ”overloaded” perché fa riferimento a più di un metodo.

ESERCIZIO

Scrivere un programma che inizializzi una stringa con ”Mississippi”, poi vi sostituisca tutte le ”i” con ”ii” e visualizzi la lunghezza della stringa risultante. Nella stringa così ottenuta, sostituire tutte le occorrenze di ”ss” con ”s”, poi visualizzate la lunghezza della stringa risultante.

Gli oggetti della classe Rectangle descrivono forme rettangolari.

COSTRUIRE OGGETTI

COSTRUIRE OGGETTI

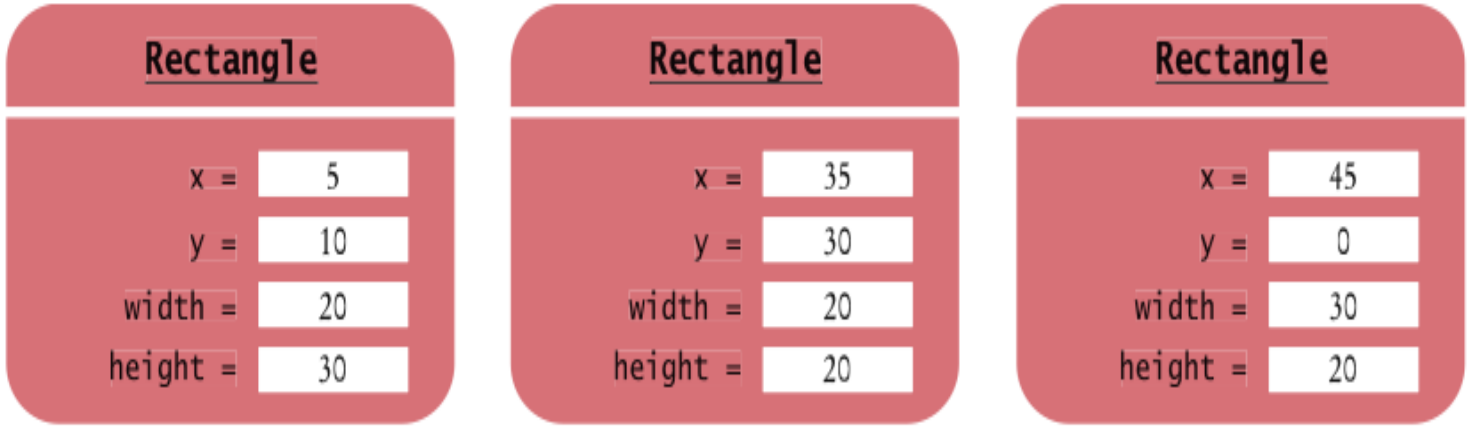
L'oggetto Rectangle non è una forma rettangolare.

È un oggetto che contiene un insieme di numeri.

I numeri descrivono il rettangolo

Ogni rettangolo è descritto da:

- Le coordinate x e y del suo angolo in alto a sinistra
- La sua larghezza
- E la sua altezza.



COSTRUIRE OGGETTI

Nel computer, un oggetto Rectangle è un blocco di memoria che contiene quattro numeri.

COSTRUIRE OGGETTI

Si utilizza l'operatore new, seguito da un nome di classe e argomenti, per costruire nuovi oggetti.

new Rectangle(5, 10, 20, 30);

In dettaglio:

Il nuovo operatore crea un oggetto Rectangle

Utilizza i parametri (in questo caso, 5, 10, 20 e 30) per inizializzare i dati dell'oggetto

Restituisce l'oggetto

Il processo di creazione di un nuovo oggetto si chiama costruzione.

I quattro valori 5, 10, 20 e 30 sono chiamati argomenti di costruzione. Di solito l'output del nuovo operatore è memorizzato in una variabile:

```
Rectangle box = new Rectangle(5, 10, 20, 30);
```

COSTRUIRE OGGETTI

Il processo di creazione di un nuovo oggetto si chiama costruzione.

I quattro valori 5, 10, 20 e 30 sono chiamati argomenti di costruzione.

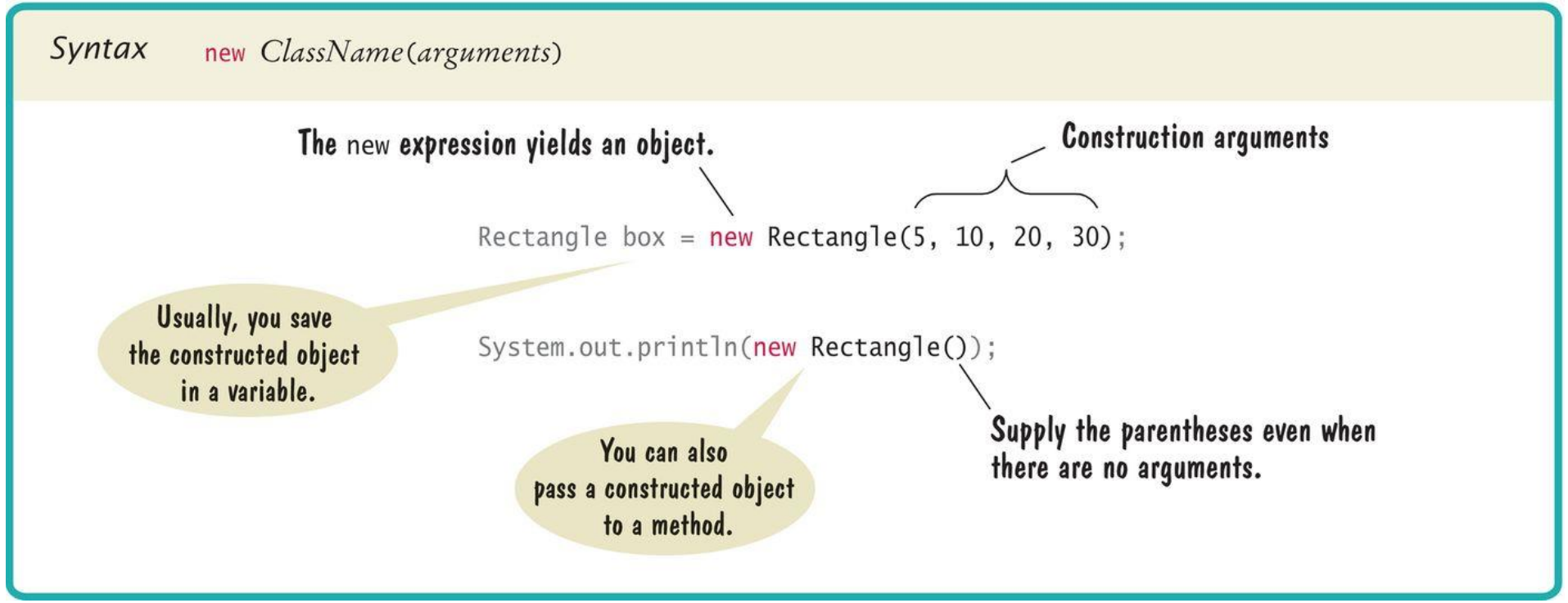
Di solito l'output del nuovo operatore è memorizzato in una variabile:

```
Rectangle box = new Rectangle(5, 10, 20, 30);
```

Un altro possibile costruttore è:

```
new Rectangle()
```

COSTRUIRE OGGETTI



COSTRUIRE OGGETTI

- Come si costruisce un quadrato con centro (100, 100) e lato 20?
- Il metodo getWidth restituisce la larghezza di un oggetto Rectangle. Cosa stampa la seguente dichiarazione?

```
System.out.println(new Rectangle().getWidth());
```

COSTRUIRE OGGETTI

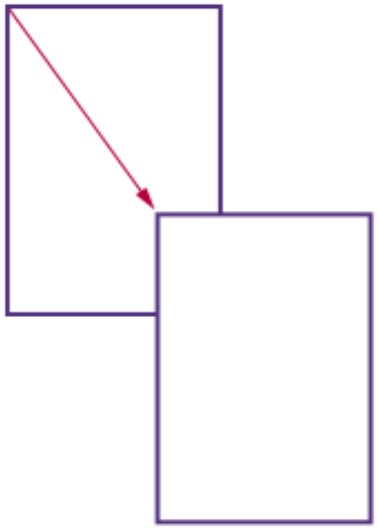
- Metodo accessor: non modifica i dati interni dell'oggetto su cui viene invocato.

  Restituisce informazioni sull'oggetto

  Esempio: metodo length della classe String

  Esempio: double width = box.getWidth();
- Metodo Mutator: cambia i dati dell'oggetto

  box.translate(15, 25);



COSTRUIRE OGGETTI

```
Rectangle box = new Rectangle(5, 10, 20, 30);
System.out.println("Before: " + box.getX());
box.translate(25, 40);
System.out.println("After: " + box.getX());
```

Che cosa produce?

COSTRUIRE OGGETTI

```
String greeting = "Hello";
System.out.println(greeting.toUpperCase());
System.out.println(greeting);
```

Che cosa produce?

API

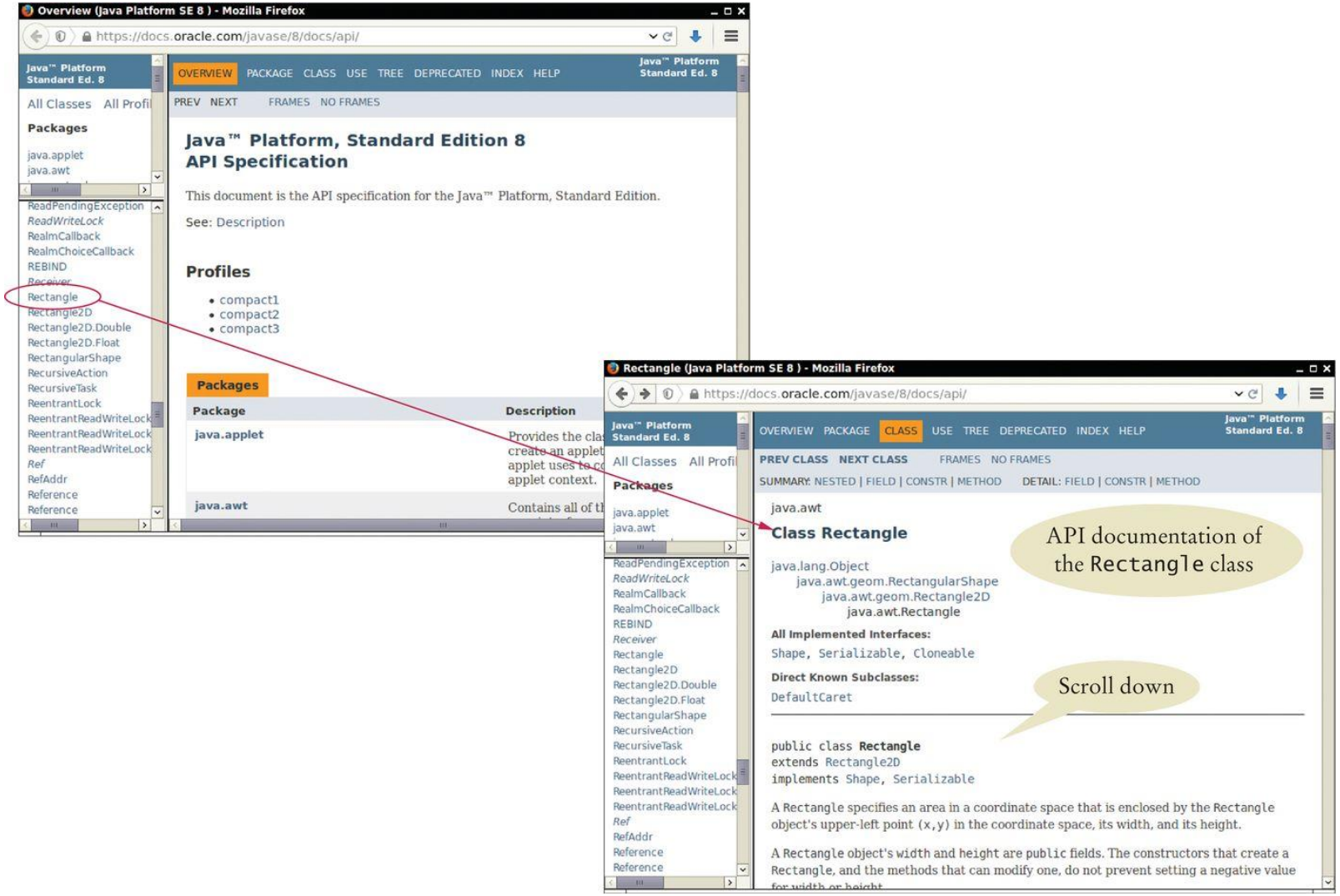
- API: Application Programming Interface
- Documentazione API: elenca classi e metodi della libreria Java

- Programmatore di applicazioni: un programmatore che utilizza le classi Java per creare un programma (o un'applicazione) per computer

- Programmatore di sistemi: un programmatore che progetta e implementa classi di librerie come PrintStream e Rectangle

- http://docs.oracle.com/javase/7/docs/api/index.html

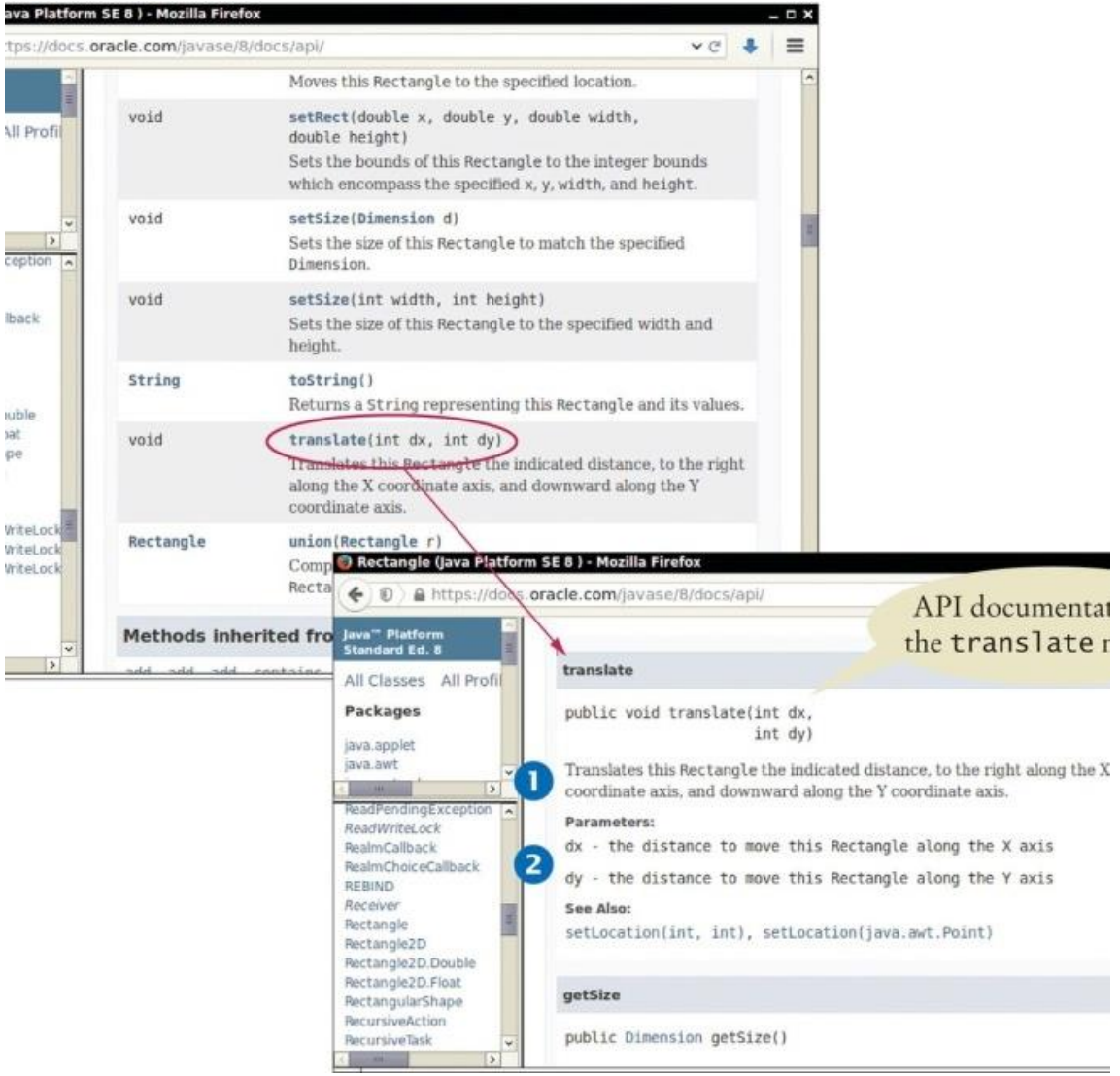
API


Overview (Java Platform SE 8) - Mozilla Firefox
https://docs.oracle.com/javase/8/docs/api/
OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP
PREV NEXT FRAMES NO FRAMES
Java™ Platform, Standard Edition 8
API Specification
This document is the API specification for the Java™ Platform, Standard Edition.
See: Description
Profiles
compact1
compact2
compact3
Packages
Package
Description
java.applet
java.awt
Rectangle (Java Platform SE 8) - Mozilla Firefox
https://docs.oracle.com/javase/8/docs/api/
OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
java.awt
Class Rectangle
java.lang.Object
java.awt.geom.RectangularShape
java.awt.geom.Rectangle2D
java.awt.Rectangle
All Implemented Interfaces:
Shape, Serializable, Cloneable
Direct Known Subclasses:
DefaultCaret
public class Rectangle
extends Rectangle2D
implements Shape, Serializable
A Rectangle specifies an area in a coordinate space that is enclosed by the Rectangle object's upper-left point (x,y) in the coordinate space, its width, and its height.
A Rectangle object's width and height are public fields. The constructors that create a Rectangle, and the methods that can modify one, do not prevent setting a negative value
API documentation of the Rectangle class
Scroll down

API

La documentazione API per ogni classe ha:

- Una sezione che descrive lo scopo delle tabelle
- Summary della classe per i costruttori e i metodi
- Facendo clic sul collegamento di un metodo si accede a una descrizione dettagliata del metodo



API:

La descrizione dettagliata di un metodo mostra:

- L'azione che svolge il metodo
- I tipi e i nomi delle variabili dei parametri che ricevono gli argomenti quando viene chiamato il metodo
- Il valore che restituisce (o la parola riservata void se il metodo non restituisce alcun valore).

Pacchetti

Le classi Java sono raggruppate in pacchetti.

La classe Rectangle appartiene al pacchetto java.awt. Per usare la classe Rectangle devi importare il pacchetto:

```
import java.awt.Rectangle;
```

Inserisci questa istruzione nella parte ‘superiore’ del tuo programma.

Non è necessario importare classi nel pacchetto java.lang come String e System.

Pacchetti

Syntax `import` *packageName.ClassName*;