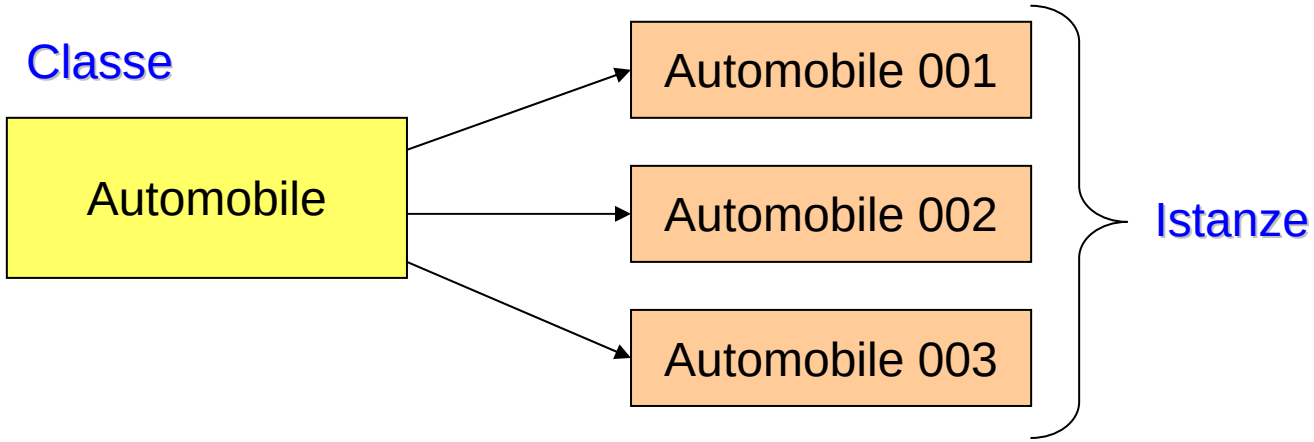
Introduzione alle classi e agli oggetti

Walter Didimo

Classi e oggetti

La classe rappresenta l'unità di base della programmazione ad oggetti:

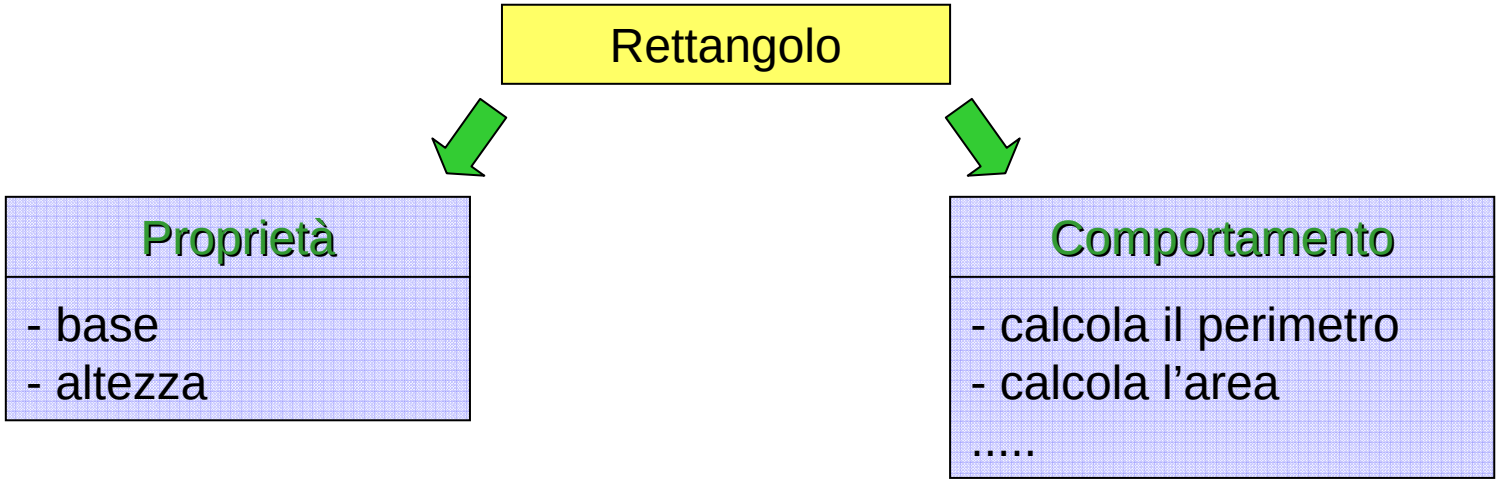
- una classe definisce una *tipologia di elementi* (cioè una categoria di elementi dello stesso tipo)
- un elemento del tipo definito da una classe si chiama oggetto o istanza della classe



Caratteristiche delle classi

Definire una classe significa specificare:

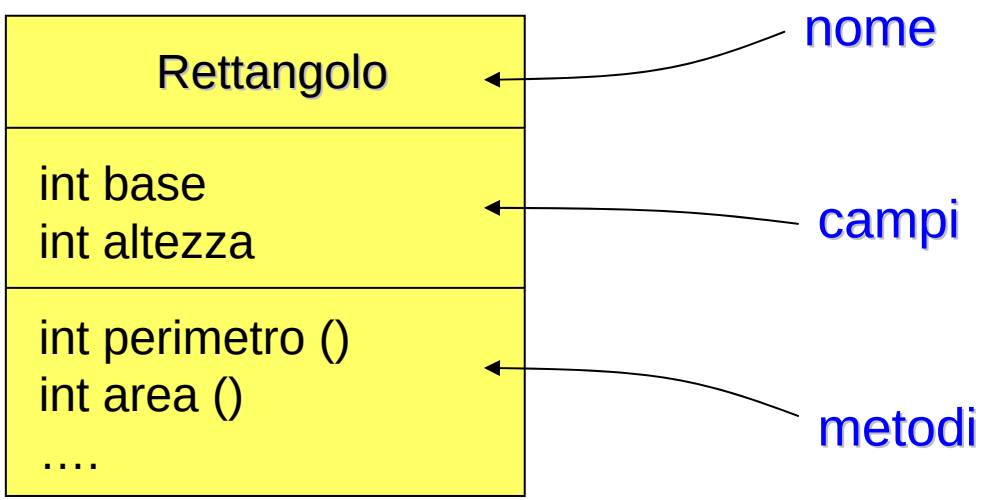
- le proprietà delle sue istanze, cioè *come* esse sono fatte
- il comportamento delle sue istanze, cioè *cosa* esse sanno fare (quali servizi possiamo richiedere loro)



Struttura di una classe

Una classe ha un nome e si compone di:

- un insieme di campi (detti anche attributi), che descrivono le proprietà delle sue istanze
- un insieme di metodi, che descrivono il comportamento delle sue istanze



Nome di una classe

Il nome di una classe descrive in breve la tipologia delle sue istanze

- *regole:*
  - non può contenere spazi bianchi e simboli di punteggiatura
- *convenzioni:*
  - inizia sempre con una lettera maiuscola
  - se si compone di più parole, le parole si scrivono attaccate ed ogni parola inizia con lettera maiuscola

Esempi di nomi

- nomi validi che rispettano le convenzioni:

  Rettangolo
  AutomobileDaNoleggio

- nomi validi che non rispettano le convenzioni:

  rettangolo
  AutomobiledaNoleggio

- nomi non validi:

  Automobile Da Noleggio
  Automobile.Da.Noleggio

Campi di una classe

I campi descrivono le proprietà delle istanze della classe; ogni campo:

- ha un nome
- descrive una proprietà
- può assumere valori di un certo tipo (dominio); esempi
  - *int* numeri interi
  - *double* numeri reali
  - *String* stringhe alfanumeriche

  .....

Nome di un campo

- *regole:*

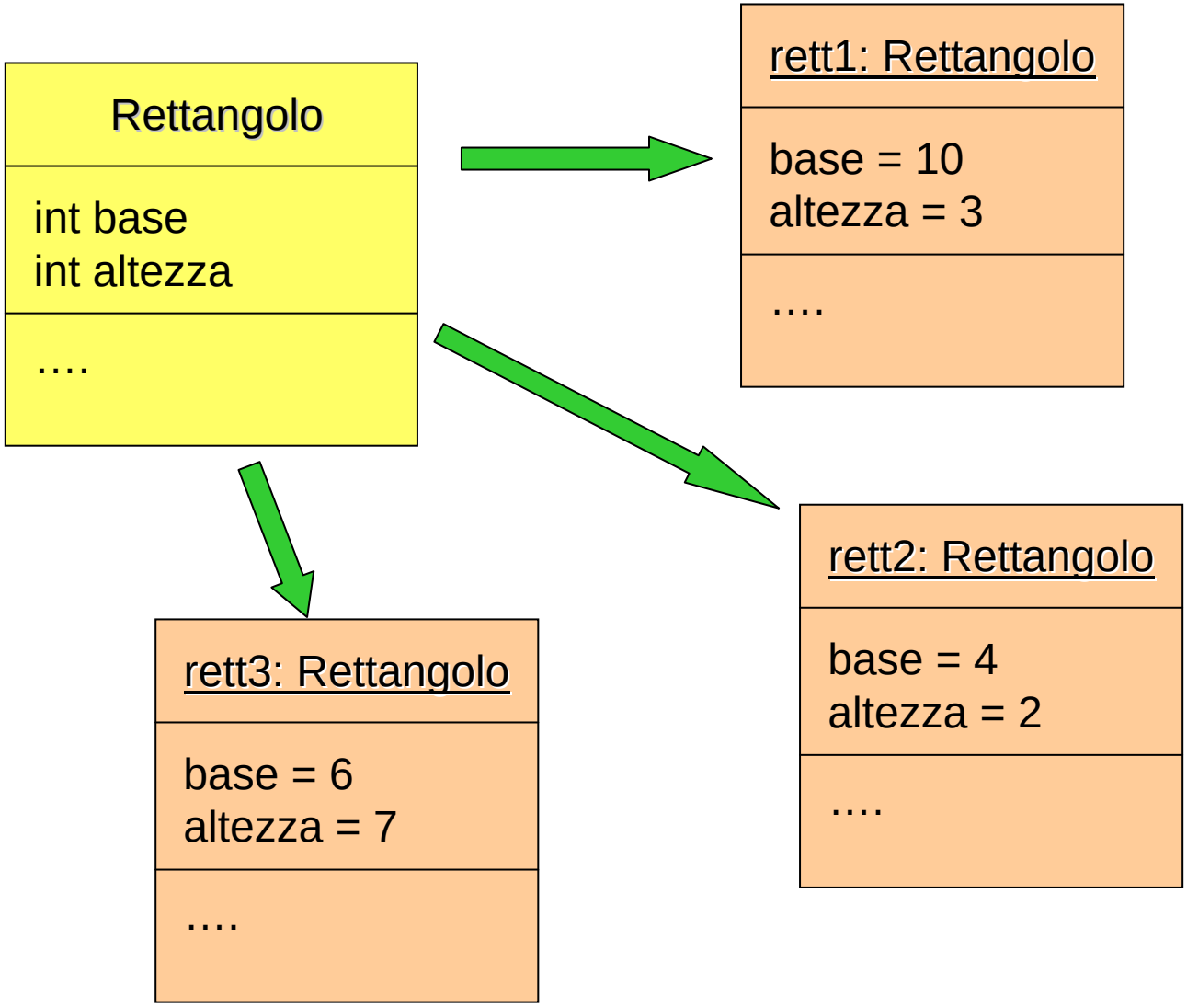
  – non può contenere spazi bianchi e simboli di punteggiatura

- *convenzioni:*

  – inizia sempre con una lettera minuscola

  – se si compone di più parole, le parole si scrivono attaccate e dalla seconda in poi ogni parola inizia con una lettera maiuscola

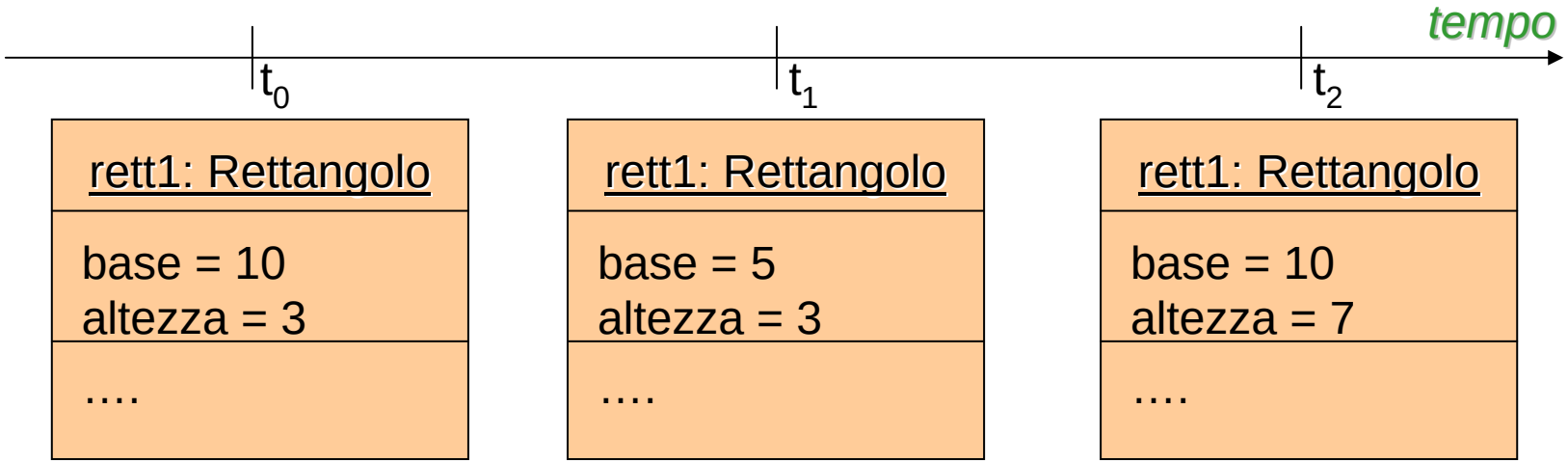
Oggetti (istanze) di una classe



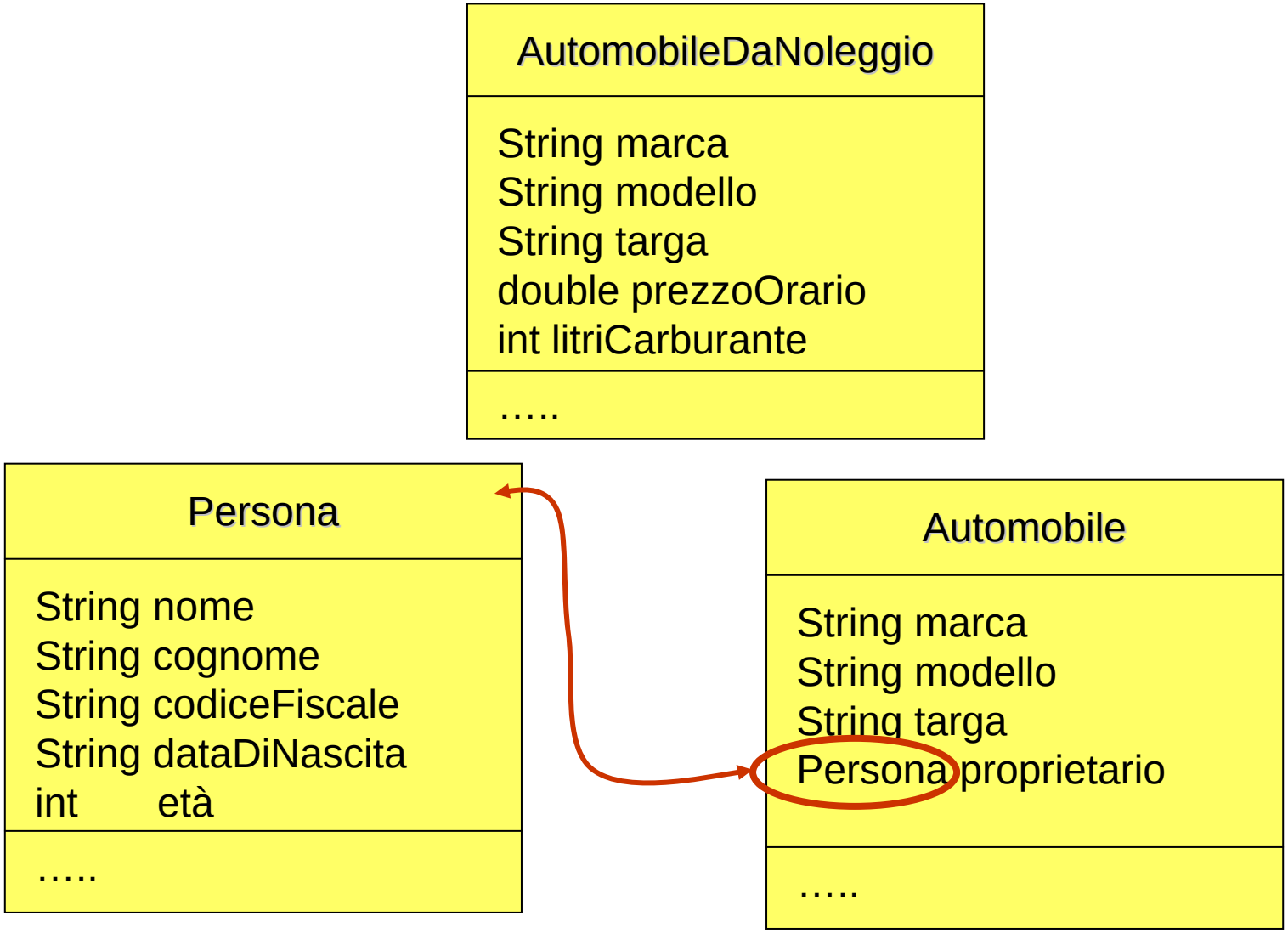
Stato di un oggetto

L'insieme dei valori assunti dai campi di un oggetto istante per istante determina lo stato dell'oggetto

- lo stato di un oggetto può cambiare nel tempo
- un oggetto cambia stato ogni volta che varia il valore di almeno uno dei suoi campi



Altri esempi di classi e campi



Metodi di una classe

I metodi descrivono il comportamento delle istanze della classe; ogni metodo:

- ha un nome
- descrive un servizio che gli oggetti sanno svolgere
- ha un corpo, cioè un insieme di *istruzioni* che dicono come il metodo deve essere svolto
- può prendere dei *dati in ingresso* (detti anche parametri)
- può restituire un dato in uscita *(dato di ritorno)*

Prototipo e signature di un metodo

Il prototipo di un metodo è la specifica del nome, del tipo di parametri, e del tipo di dato in uscita

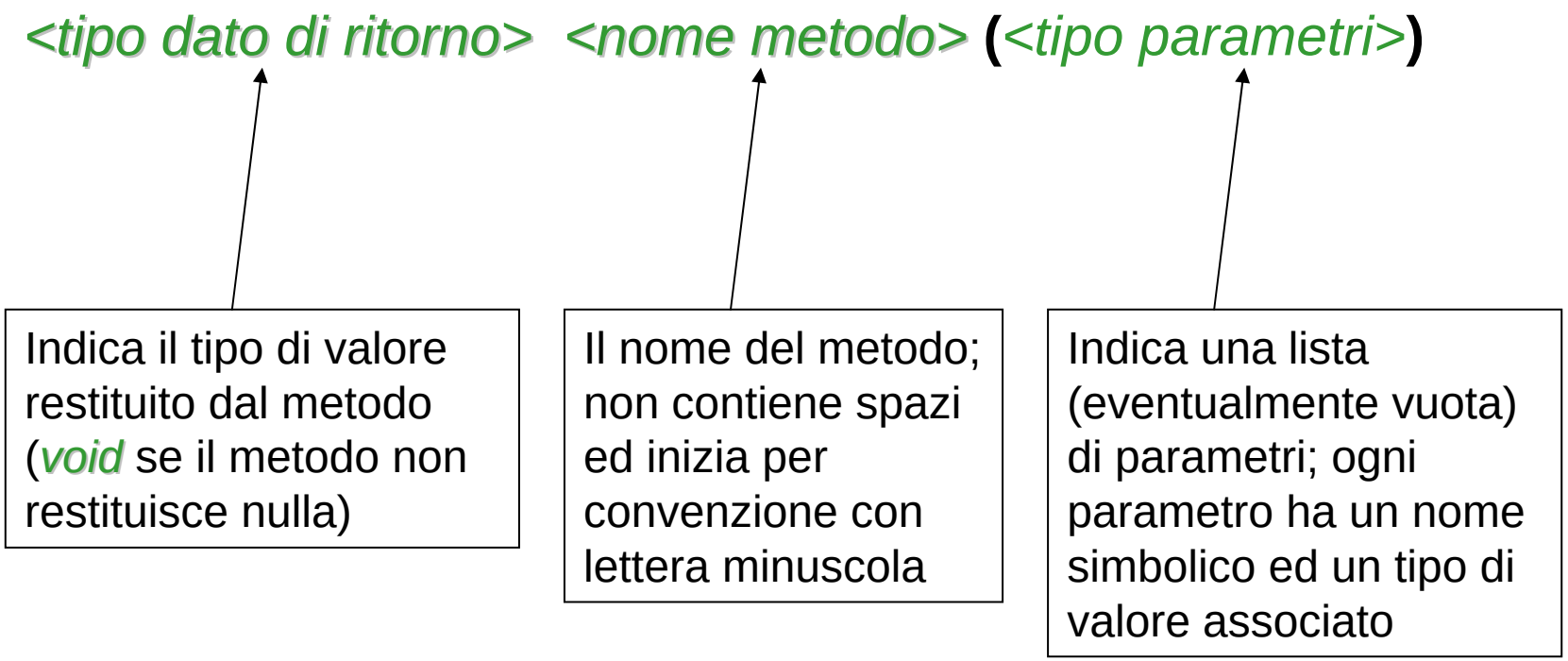
prototipo = *nome* + *tipo parametri* + *tipo dato di ritorno*

La signature di un metodo è la sola specifica del nome e del tipo di parametri

signature = *nome* + *tipo parametri*

Sintassi di un prototipo

Il prototipo di un metodo ha la seguente sintassi



Esempi di prototipi

Consideriamo gli oggetti di una classe Rettangolo; ecco un insieme di possibili metodi, descritti tramite i loro prototipi

tipo dato di ritorno	*nome metodo*	*tipo parametri*
int	perimetro	( )
void	cambiaDimensioni	(int b, int a)
double	frazioneDiArea	(double f)

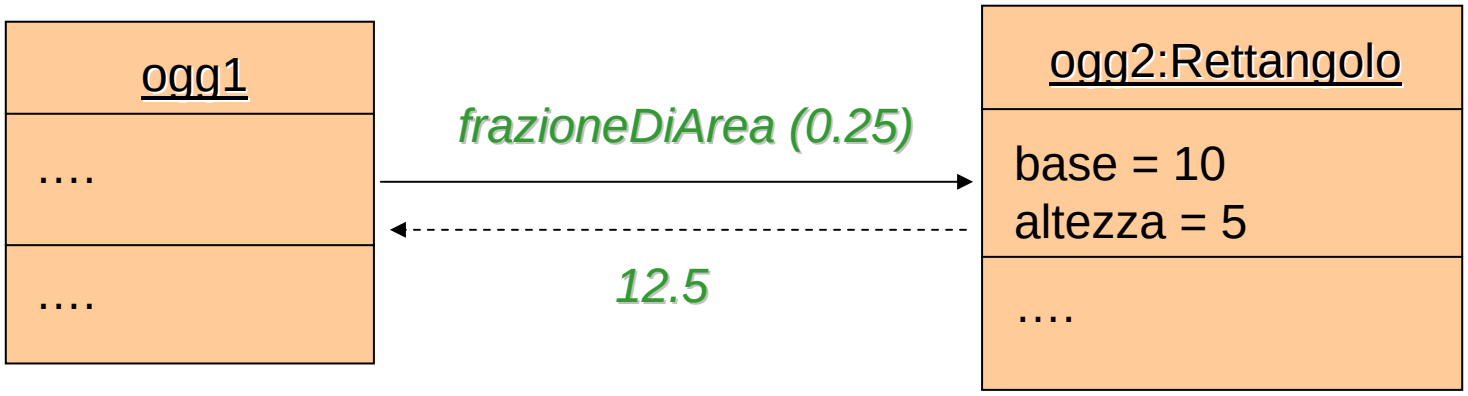
Invocazione di metodi

Un programma ad oggetti consiste nella definizione di classi e oggetti *cooperanti*

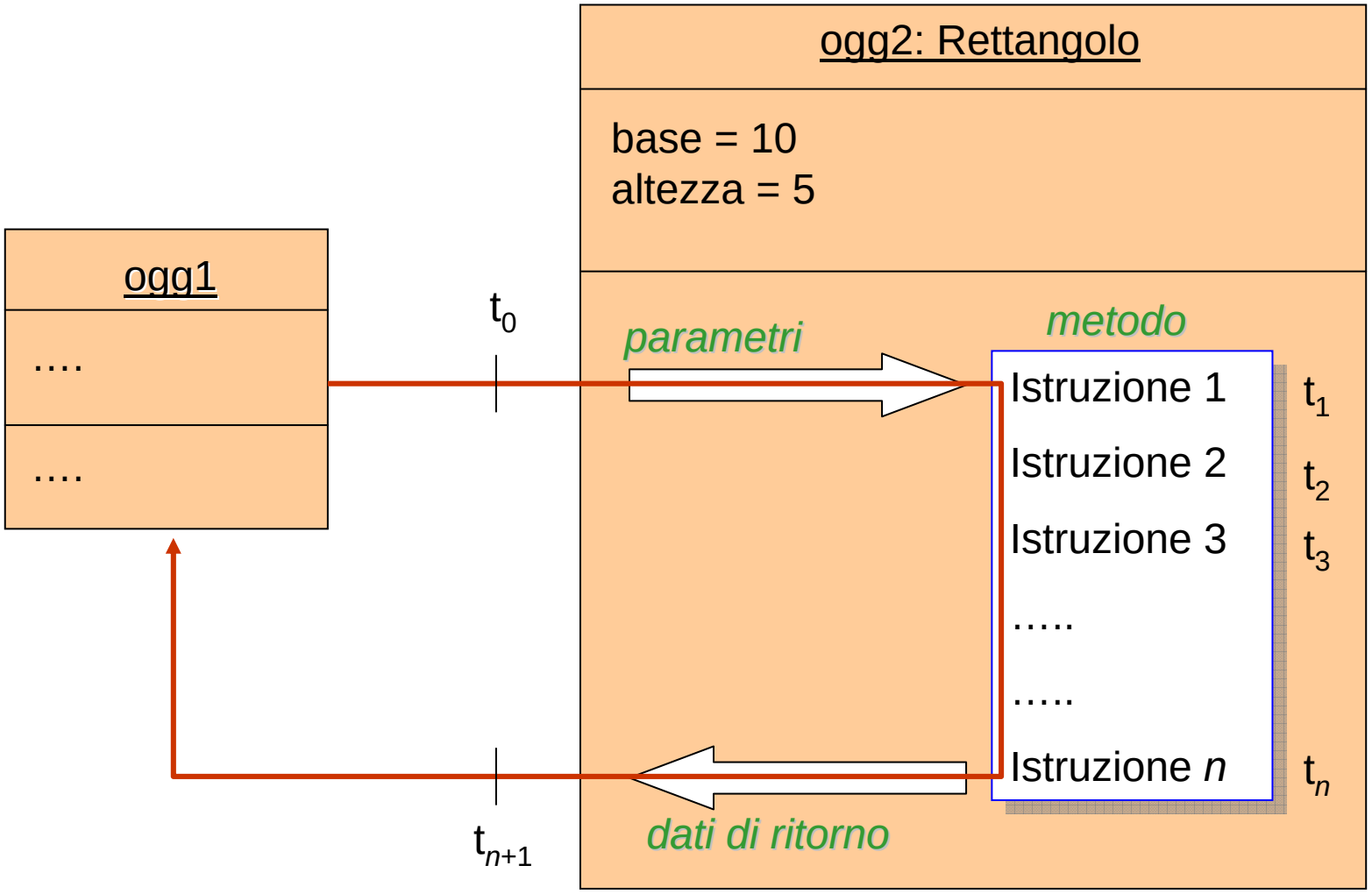
- un oggetto *ogg1* può ottenere un servizio da un oggetto *ogg2*, chiedendogli di eseguire uno dei suoi metodi
- la richiesta di servizio si chiama invocazione di metodo e corrisponde all'invio di un messaggio da *ogg1* ad *ogg2*
  - *ogg1* deve specificare la signature del metodo di *ogg2* che vuole invocare

Schema di invocazione di metodi

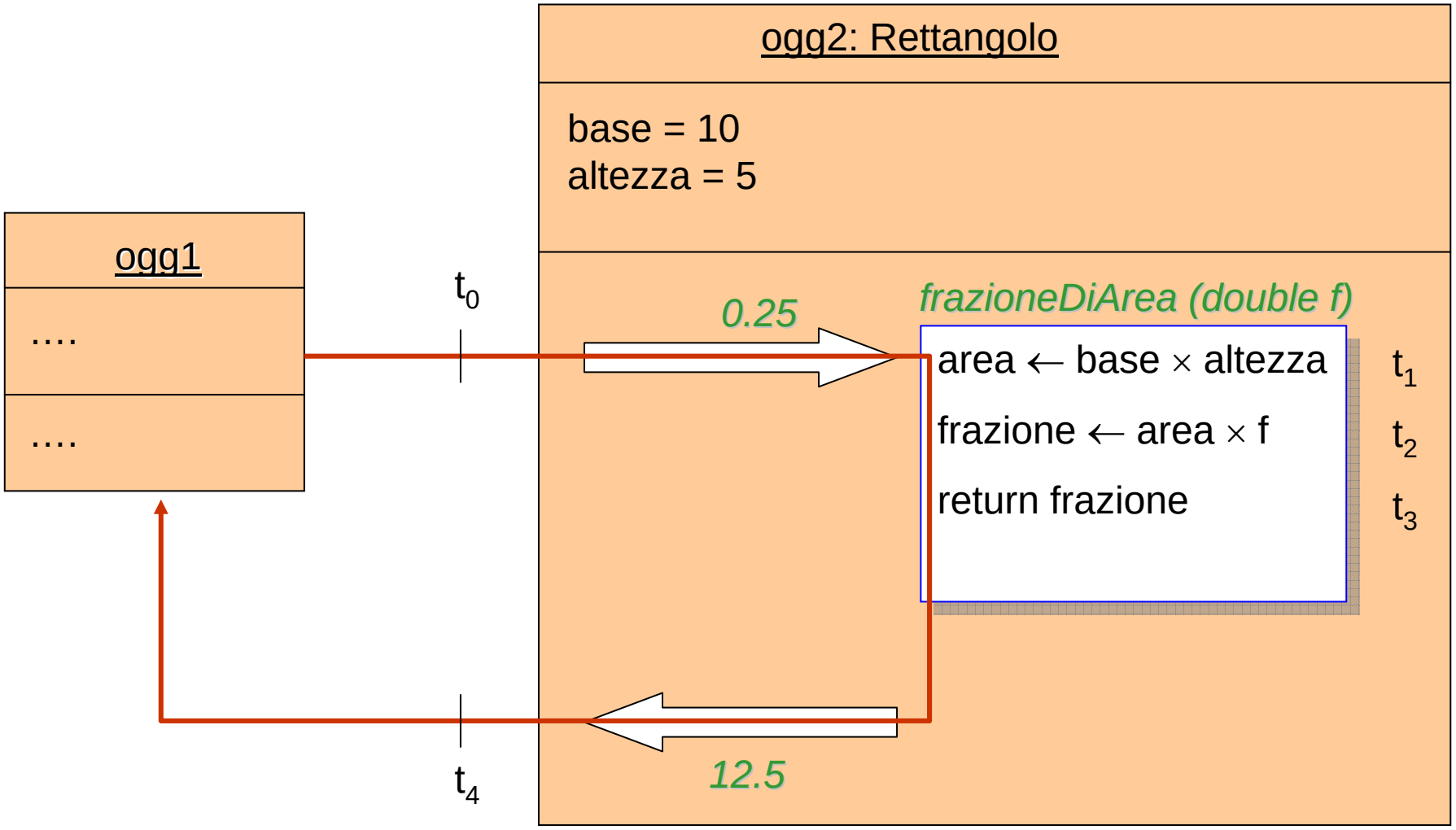
Sia *ogg2* un oggetto della classe Rettangolo e supponiamo che *ogg1* voglia chiedere ad *ogg2* di eseguire il metodo *double frazioneDiArea(double f),* specificando per *f* il valore 0.25



Cosa accade nel tempo

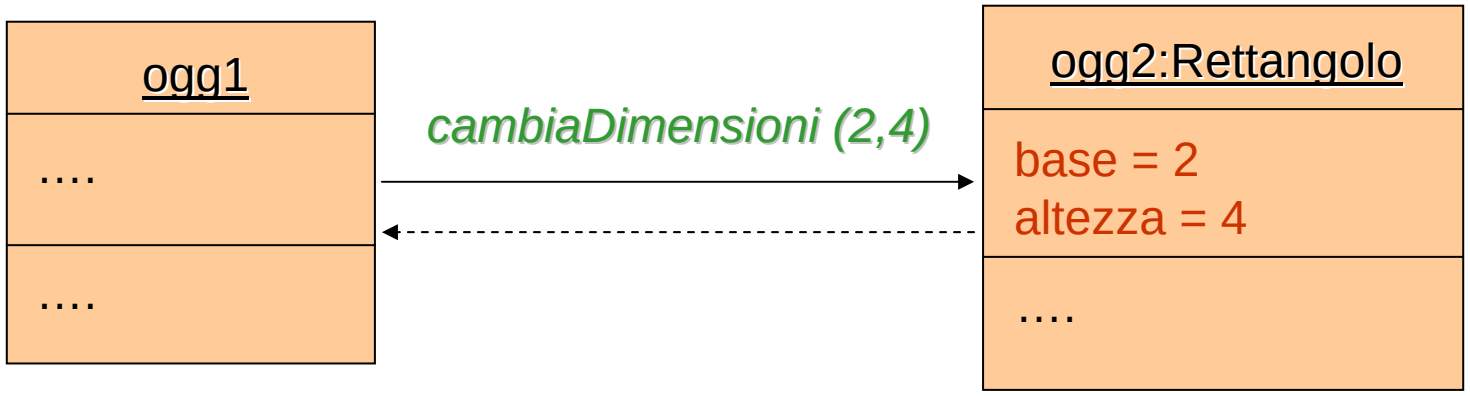


Nell'esempio specifico



Un altro esempio

Sia *ogg2* un oggetto della classe Rettangolo e supponiamo che *ogg1* voglia chiedere ad *ogg2* di eseguire il metodo *void cambiaDimensioni(int a, int b),* specificando per *a e b* i valori 2 e 4



Questo metodo cambia lo stato dell'oggetto!

Glossario dei termini principali

Termine	Significato
Classe	Descrizione di una tipologia di elementi; una classe specifica le proprietà e il comportamento della categoria di elementi che rappresenta
Oggetto (o Istanza)	Uno degli elementi di una classe
Campo (o Attributo)	Costrutto di una classe che definisce una proprietà delle sue istanze; un campo può assumere valori di un certo dominio
Metodo	Descrizione di uno specifico servizio offerto dalle istanze di una classe; un metodo descrive parte del comportamento di una istanza della classe
Stato	Insieme dei valori assunti dai campi di un oggetto istante per istante; lo stato di un oggetto può cambiare nel tempo
Corpo (di un metodo)	Insieme delle istruzioni che specificano come il metodo va svolto
Parametri (di un metodo)	Insieme dei dati che il metodo prende in ingresso
Prototipo (di un metodo)	Specifica del nome, dei parametri e del tipo di dato restituito da un metodo
Signature (di un metodo)	Specifica del nome e dei parametri di un metodo