

# Oggetti e dati primitivi

Programmazione  
Corso di laurea in Informatica

## Introduzione agli oggetti

- Interagiamo con oggetti di uso quotidiano, conoscendone le *funzioni*, ma non il *funzionamento interno*
  - Gli oggetti sono **scatole nere** dotate di interfaccia che limita l'accesso ai meccanismi interni
  - Gli oggetti hanno uno **stato**
    - L'insieme delle proprietà che lo caratterizzano in un dato istante
  - e un **comportamento**
    - L'insieme delle azioni che un oggetto può compiere
- Un oggetto sw è un'**astrazione** o un **modello** della realtà che limita il numero dei **dettagli** rappresentati *all'essenziale* per il contesto considerato

AA 2003/04 © Alberti 2 Programmazione 4. Oggetti e dati primitivi

## Astrazione

- L'astrazione nasconde o ignora dettagli inessenziali
- Un oggetto è un'astrazione
  - Non ci preoccupiamo dei suoi dettagli interni per usarlo
  - Non conosciamo come funziona il metodo `println` quando l'invochiamo
- Effettuiamo astrazioni continuamente
  - Possiamo trattare solo poche informazioni contemporaneamente
  - Ma se raggruppiamo le informazioni (come gli oggetti) allora possiamo trattare informazioni più complicate
- Quindi, possiamo anche scrivere software complesso organizzandolo attentamente in classi e oggetti

AA 2003/04 © Alberti 3 Programmazione 4. Oggetti e dati primitivi

## Gli oggetti software

- Lo **stato** di un oggetto sw è descritto e rappresentato da una o più **variabili**
  - Una variabile è un **dato** individuato da un **identificatore**
- il **comportamento** è definito dai **metodi**
- Un oggetto sw è costituito dall'insieme delle variabili e dei metodi

AA 2003/04 © Alberti 4 Programmazione 4. Oggetti e dati primitivi

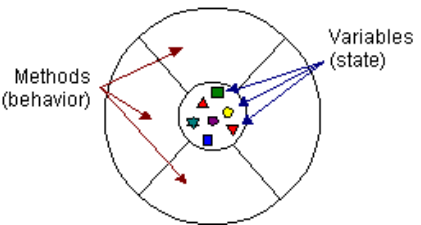
## Oggetti e classi

- Gli **oggetti** sono generati da una **classe**
  - Si dicono anche **istanze** della classe
- La **classe** è uno schema per produrre una categoria di oggetti strutturalmente identici
  - La classe costituisce il **prototipo**
  - Una classe è una fabbrica di istanze: possiede lo schema e la tecnica di produzione



AA 2003/04 © Alberti 5 Programmazione 4. Oggetti e dati primitivi

## Gli oggetti come astrazione



- Un oggetto che modella una bicicletta
  - Una velocità (20 Km/h), il giro dei pedali (15 g/m) e la marcia (5°) sono **variabili di istanza**
  - proprietà rappresentate in ciascuna bicicletta modellata

AA 2003/04 © Alberti 6 Programmazione 4. Oggetti e dati primitivi

### Gli oggetti come astrazione – 2

- Inoltre nel modello rappresentiamo funzioni come **frenare** o **cambiare marcia**, che modificano le variabili d'istanza
- Si chiamano **metodi d'istanza** perché hanno accesso alle variabili d'istanza e le modificano

AA 2003/04 © Alberti 7 Programmazione 4. Oggetti e dati primitivi

### Le istanze

- Definita una classe, si possono creare un numero arbitrario di oggetti appartenenti alla classe

AA 2003/04 © Alberti 8 Programmazione 4. Oggetti e dati primitivi

### Incapsulamento dei dati

- Nascondere le informazioni fornendo **un'interfaccia**
- Le **variabili** di un oggetto, che ne rappresentano lo stato, sono **nascoste** all'interno dell'oggetto, **accessibili** solo ai metodi
- Idealmente i metodi proteggono le variabili
- 4 diversi livelli di accesso a metodi e variabili
  - **Private**: accessibili solo alla classe
  - **Protected**: accessibili a classe, sottoclassi e al pacchetto
  - **Public**: accessibili a chiunque e quindi anche alle sottoclassi
  - **Senza modificatori**: accessibili solo alle sottoclassi del pacchetto
- Consente modularità e flessibilità

AA 2003/04 © Alberti 9 Programmazione 4. Oggetti e dati primitivi

### I messaggi

- Gli oggetti interagiscono tra loro per ottenere funzioni più complesse
  - La bicicletta appesa in garage è un oggetto e basta, ci vuole un ciclista che interagisca con lei perché sia interessante
- Gli oggetti sw per interagire si mandano messaggi
  - Chiedendo di eseguire un certo metodo

AA 2003/04 © Alberti 10 Programmazione 4. Oggetti e dati primitivi

### I messaggi – 2

- Spesso i metodi necessitano di informazioni per poter essere eseguiti: **i parametri**
- Tre componenti:
  - L'oggetto a cui il messaggio è rivolto
  - Il metodo da eseguire per ottenere un certo effetto
  - I parametri se necessari al metodo

changeGears(TowerGear)

AA 2003/04 © Alberti 11 Programmazione 4. Oggetti e dati primitivi

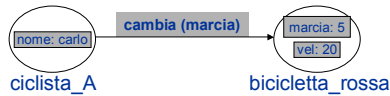
### I messaggi – 3

- Un **oggetto** può essere visto come un insieme di servizi che possiamo chiedere di eseguire
- I servizi sono definiti dai **metodi**
- Il comportamento degli oggetti è definito dai suoi metodi e il meccanismo di invio dei messaggi consente l'interazione tra gli oggetti
- Gli oggetti che si scambiano i **messaggi** possono anche essere **'distanti'** tra loro
  - Su macchine diverse
  - Non appartenenti allo stesso modello

AA 2003/04 © Alberti 12 Programmazione 4. Oggetti e dati primitivi

### Interfaccia

- L'**interfaccia** è l'insieme dei messaggi che un oggetto è in grado di interpretare
- Un oggetto deve soddisfare la richiesta di un messaggio
- Eseguendo il **metodo** si soddisfa la risposta ad un messaggio da parte di un agente



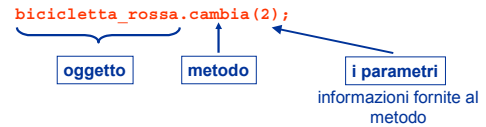
AA 2003/04  
© Alberti

13

Programmazione  
4. Oggetti e dati primitivi

### Inviare messaggi

- Il ciclista **ciclista\_A** che vuole cambiare marcia invia il messaggio all'oggetto **bicicletta\_rossa**



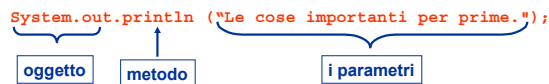
AA 2003/04  
© Alberti

14

Programmazione  
4. Oggetti e dati primitivi

### Invocazione di un metodo

- Molte istruzioni sono invocazioni di metodi su oggetti
- La sintassi della chiamata del metodo:  
`oggetto.nomeMetodo (parametri)`
- Chiediamo il *servizio di stampa*, invocando il metodo `println` dell'oggetto `System.out`



AA 2003/04  
© Alberti

15

Programmazione  
4. Oggetti e dati primitivi

### Metodi e oggetti

- I metodi possono essere invocati su oggetti della classe che hanno quel metodo nella loro interfaccia
  - Il metodo `println` si può applicare a oggetti della classe `PrintStream`  
`System.out.println()`
  - Il metodo `length` si può applicare a oggetti della classe `String`  
`"salute a tutti".length`
  - Quindi causa errore chiamare:  
`"salute a tutti".println`

AA 2003/04  
© Alberti

16

Programmazione  
4. Oggetti e dati primitivi

### I metodi println e print

- L'oggetto `System.out` della classe `PrintStream` fornisce altri servizi
  - Il metodo `print`
  - simile al metodo `println` - non fa avanzare il cursore alla riga successiva
  - Quindi quello che è stampato dopo l'istruzione `print` appare sulla stessa riga
- Esempio [Conto alla rovescia.java](#)

AA 2003/04  
© Alberti

17

Programmazione  
4. Oggetti e dati primitivi

### La classe predefinita String

- Java mette a disposizione oggetti di tipo predefinito `String` per rappresentare stringhe di caratteri
- Ogni stringa letterale, delimitata dai segni `"`, `"` è un oggetto della classe `String`
  - Una stringa letterale non può essere spezzata su più righe nel codice
- Esempio [Fatti.java](#)

AA 2003/04  
© Alberti

18

Programmazione  
4. Oggetti e dati primitivi