

# Programmazione Orientata agli Oggetti in Linguaggio Java

## Classi e Oggetti: Introduzione

versione 2.1

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons  
(vedi ultima pagina)

G. Mecca – Università della Basilicata – mecca@unibas.it



Classi e Oggetti: Introduzione >> Sommario



## Sommario

- Il Messaggio Fondamentale
- Un Esempio
  - ⇒ La Morra Cinese in Java
- Concetti Noti
- Concetti Nuovi
  - ⇒ Componenti
- Metodologia Didattica
  - ⇒ Progetti di Riferimento

G. Mecca - Programmazione Orientata agli Oggetti

2



## Il Messaggio Fondamentale

### ○ ATTENZIONE

*la programmazione Orientata agli Oggetti  
è basata su principi e tecniche molto  
diversi rispetto a quelli della programmazione  
procedurale tradizionale*

*è necessario fare uno sforzo per  
cambiare drasticamente mentalità*



## Il Messaggio Fondamentale

### ○ In effetti

- ⇒ i linguaggi di programmazione a oggetti sono basati su alcuni concetti comuni ai linguaggi di programmazione procedurali
- ⇒ in questo corso sono prerequisiti

### ○ Però

- ⇒ ci sono però numerosi concetti completamente nuovi, sia di carattere tecnologico che metodologico



## Il Messaggio Fondamentale

- In particolare
  - ⇒ cambia significativamente il modo di organizzare il codice
- Nella programmazione procedurale
  - ⇒ l'unico strumento di organizzazione del codice è il sottoprogramma (modulo)
  - ⇒ le applicazioni si costruiscono scegliendone ed implementandone i vari moduli



## Il Messaggio Fondamentale

- L'esecuzione dell'applicazione
  - ⇒ sequenza di chiamate di sottoprogrammi
  - ⇒ in ogni momento, la pila di attivazione rappresenta completamente lo stato dell'applicazione
  - ⇒ in particolare, contiene tutti i dati in quel momento utilizzati dal programma



## Un Esempio

>> morraCinese4.cpp

- Ragioniamo su un esempio
  - ⇒ il gioco della Morra Cinese
- La Morra Cinese in “clean C”
  - ⇒ circa 250 linee di codice
  - ⇒ un unico file di codice sorgente
  - ⇒ main + 15 sottoprogrammi
  - ⇒ complessivamente 7 costanti globali, 31 parametri, 16 variabili locali ai moduli



## Un Esempio

- Esecuzione dell'applicazione
  - ⇒ viene avviato il main
  - ⇒ main chiama inizioGioco
  - ⇒ inizioGioco chiama schermolniziale
  - ⇒ ...
  - ⇒ in ogni momento la pila mi dice il nome, il punteggio della partita e il punteggio complessivo del gioco

Classi e Oggetti: Introduzione >> Un Esempio

Un Esempio

```

#100 const int MANI=3;
#101 void main() { // id blocc=0
#102     string nome;
#103     inizioGioco(nome);
#104     gioca(nome);
#105     schermoFinale(nome);
#106 }
#107 void inizioGioco(string& nome) { // id=1
#108     int seme = time(NULL); /**
#109     srand(seme); /**
#110     schermolniziale(nome);
#111     return;
#112 }
#113 void schermolniziale(string& nome) { // id=2
#114     cout << "Immetti il tuo nome ";
#115     getline(cin, nome);
#116     cout << " Benvenuto, " << nome << "\n";
#117     return;
#118 }

```

#1000	MANI	3
#1001	<b>id blocco</b>	<b>0</b>
#1002	rif. codice	#102
#1003	punto rit.	-
#1004	nome	xxx
#1005	p. cat. statica	#1000
#1006	<b>id blocco</b>	<b>1</b>
#1007	rif. codice	#108
#1008	punto. rit.	#104
#1009	nome	#1004
#1010	seme	xxx
#1011	p. cat. statica	#1000
#1012	<b>id blocco</b>	<b>2</b>
#1013	rif. codice	#114
#1014	punto rit.	#111
#1015	nome	#1004
#1016	p. cat. statica	#1000
...	...	...

G. Mecca - Pro...

9

Classi e Oggetti: Introduzione >> Un Esempio

## Un Esempio

>> it.unibas.morracineseswing

- La morra cinese in Java con Swing
  - ⇒ le stesse funzionalità
  - ⇒ varie novità significative
- Principale novità
  - ⇒ interfaccia utente grafica (GUI, "Graphical User Interface")
  - ⇒ l'interazione dell'utente non avviene attraverso la console ma attraverso il mouse
  - ⇒ modalità diversa di controllo (menu, bottoni..)

G. Mecca - Programmazione Orientata agli Oggetti

10



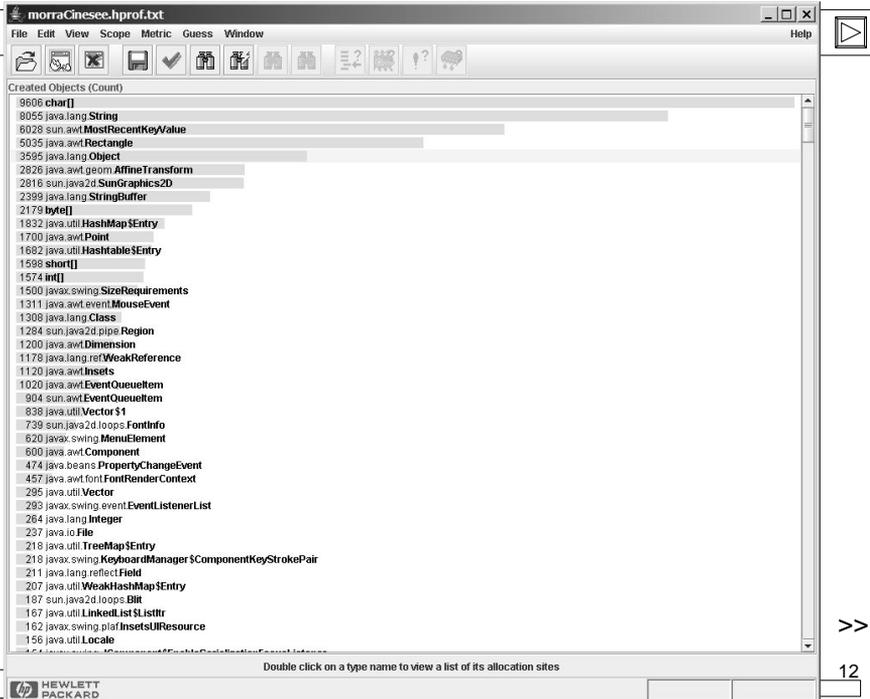
# Un Esempio

## ○ Alcune statistiche

- ⇒ oltre 500 linee di codice sorgente
- ⇒ 5 file di codice sorgente, 12 classi
- ⇒ durante l'esecuzione l'applicazione crea varie migliaia di oggetti
- ⇒ complessità molto maggiore

## ○ Informazioni sull'esecuzione

- ⇒ ottenute attraverso il "profiler" standard della macchina virtuale Java (hprof)





## Un Esempio

>> [it.unibas.morracinese](http://it.unibas.morracinese)

### ○ Attenzione

- ⇒ si potrebbe pensare che questa maggiore complessità sia dovuta all'interfaccia grafica
- ⇒ in effetti la programmazione dell'interfaccia grafica introduce una certa complessità
- ⇒ ma in parte le differenze nel modello di programmazione sono inerenti
- ⇒ nella versione console dell'applicazione Java la sostanza non cambia



## Un Esempio

### ○ Il messaggio

- ⇒ il modello di esecuzione è diverso da quello della programmazione procedurale
- ⇒ l'esecuzione è basata sul concetto di pila
- ⇒ ma lo stato dell'applicazione non è tenuto esclusivamente nella pila
- ⇒ bensì dipende dall'interazione tra le classi e gli oggetti dell'applicazione



## Concetti Noti

- Nel seguito
  - ⇒ cercheremo di interpretare queste novità
- Nel frattempo però
  - ⇒ cerchiamo di identificare alcuni concetti del linguaggio che ci sono già familiari
  - ⇒ lo facciamo raffrontando il codice della morra cinese in “clean C” (in giallo)
  - ⇒ con quello in Java basato su console



## Concetti Noti

- Elementi comuni ai due linguaggi
  - ⇒ variabile, costante, tipo di dato
  - ⇒ espressioni e assegnazioni
  - ⇒ lettura e stampa
  - ⇒ strutture di controllo (if, cicli)
  - ⇒ sottoprogrammi
- Esempio
  - ⇒ schermoMenuPrincipale, giocata



## Concetti Noti

```

int schermoMenuPrincipale() {
    int scelta;
    cout << "+-----+\n";
    cout << "|  1. Nuova Partita      |\n";
    cout << "|  2. Visualizza Vittorie  |\n";
    cout << "|  3. Esci dal Gioco      |\n";
    cout << "+-----+\n";
    cout << "  Scelta --> ";
    cin >> scelta;
    while ((scelta < 1) || (scelta > 3)) {
        cout << "***** Errore. Nuova scelta --> ";
        cin >> scelta;
    }
    return scelta;
}

```



## Concetti Noti

```

private int schermoMenuPrincipale() {
    int scelta;
    System.out.println("+-----+");
    System.out.println("|  1. Nuova Partita      |");
    System.out.println("|  2. Visualizza Vittorie  |");
    System.out.println("|  0. Esci dal Gioco      |");
    System.out.println("+-----+");
    System.out.print("  Scelta --> ");
    scelta = it.unibas.utilita.Console.leggiIntero();
    while ((scelta < 0) || (scelta > 2)) {
        System.out.print("***** Errore: Scelta scorretta. Nuova scelta --> ");
        scelta = it.unibas.utilita.Console.leggiIntero();
    }
    return scelta;
}

```



## Concetti Noti

```
string giocata(int numero) {
    string s;
    if (numero == CARTA) {
        s = "carta";
    } else if (numero == FORBICI) {
        s = "forbici";
    } else if (numero == SASSO) {
        s = "sasso";
    }
    return s;
}
```



## Concetti Noti

```
public static String getGiocata (int mossa) {
    String s;
    if (mossa == Mano.CARTA) {
        s = "carta";
    } else if (mossa == Mano.FORBICI) {
        s = "forbici";
    } else if (mossa == Mano.SASSO) {
        s = "sasso";
    }
    return s;
}
```



## Principali Concetti Nuovi

- I due concetti completamente nuovi
  - ⇒ classi ed oggetti
- In queste lezioni
  - ⇒ cosa sono le classi, cosa sono gli oggetti
  - ⇒ come si scrivono
  - ⇒ come funzionano
  - ⇒ i relativi aspetti metodologici



## Principali Concetti Nuovi

**ATTENZIONE**  
a questi cinque  
concetti

- Programmazione basata sugli oggetti
  - ⇒ cinque concetti fondamentali
- I cinque concetti fondamentali
  - ⇒ componente
  - ⇒ proprietà
  - ⇒ metodo
  - ⇒ costruttore
  - ⇒ riferimento

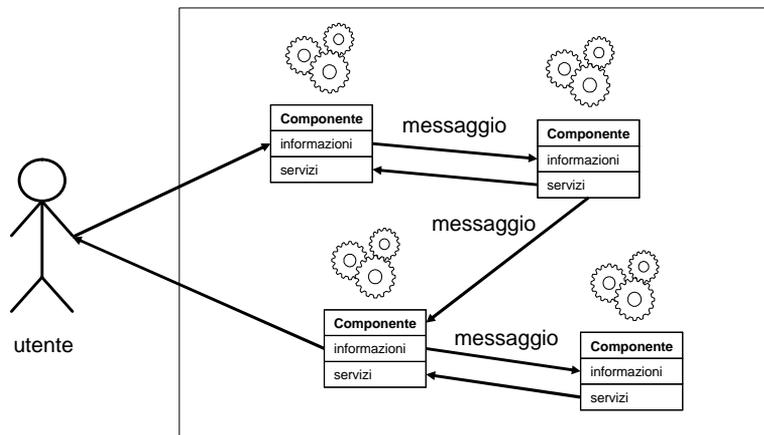


## Componenti

- Applicazione tradizionale
  - ⇒ collezione di funzioni e procedure
  - ⇒ eventualmente organizzate in librerie
- Applicazione a oggetti
  - ⇒ collezione di componenti che si scambiano messaggi
- Componente
  - ⇒ pezzo di software capace di svolgere compiti



## Componenti





## Metodologia Didattica

- Obiettivi del modulo

- ⇒ introdurre i principali concetti relativi ai componenti, alle classi e agli oggetti

- Enfasi

- ⇒ comprensione dei concetti relativi ai componenti e alle modalità di interazione

- ⇒ meno enfasi su sintassi e semantica che saranno affrontate nel prossimo modulo



## Metodologia Didattica

- Lo strumento didattico

- ⇒ per gestire il “rischio didattico” collegato al corso verrà utilizzata una metafora consistente nella presentazione dei concetti

- ⇒ la metafora serve a “raccontare” le tecniche della programmazione a oggetti con riferimento a concetti della vita quotidiana

- ⇒ per semplificare la comprensione dei concetti



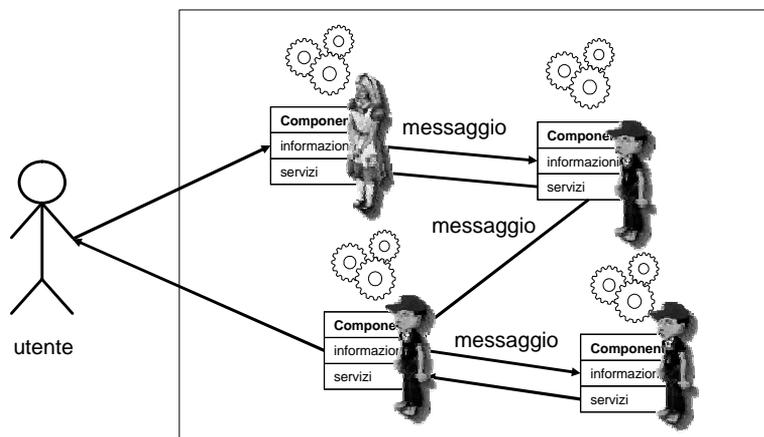
# Metodologia Didattica

## ○ Metafora

- ⇒ vedremo i componenti come piccoli robot di forma umana (li chiameremo anche “omini”)
- ⇒ questi robot si organizzano in una società molto simile a quella umana
- ⇒ le classi sono componenti femmina
- ⇒ gli oggetti sono componenti maschio
- ⇒ le interazioni tra i componenti saranno raccontate con riferimento a questa metafora



# Metodologia Didattica





## Progetti di Riferimento

- Progetti di riferimento
  - ⇒ vari progetti
  - ⇒ tutti molto semplici
  - ⇒ tutti basati su interfaccia a console
  - ⇒ ne verranno presentate varie versioni
- In questo modulo
  - ⇒ i progetti saranno “raccontati” utilizzando la metafora di riferimento



## Progetti di Riferimento

- Calcolatrice
  - ⇒ un programma per effettuare op. aritmetiche
- Circonferenze
  - ⇒ un programma per analizzare circonferenze
- Segmenti
  - ⇒ un programma per analizzare segmenti



## Riassumendo

- Il Messaggio Fondamentale
- Un Esempio
  - ⇒ La Morra Cinese in Java
- Concetti Noti
- Concetti Nuovi
  - ⇒ Componenti
- Metodologia Didattica
  - ⇒ Progetti di Riferimento



## Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.