

MERGE SORT:  $O(n \log n)$

SELECTION SORT:  $O(n^2)$

QUICK SORT:  $O(n \log n)$  media Array. sort

$O(n^2)$  pessime

## Selection Sort

ORDINARE 50.000 NUMERI: 1.100 ms

100.000 " : 4.000 ms

$T(50.000)$

$$T(100.000) \text{ proportional a } (100.000)^2 = (2 \cdot 50.000)^2 = 4 \cdot (50.000)^2$$

tempo che ci mette il Selection Sort

# VOGLIAMO METTORIZZARE NEL NOSTRO PROGRAMMA UN ELENCO DI PERSONE

Ogni persona ha:

- NOME
- COGNOME
- COD. FISCALE
- ETÀ
- STIPENDIO ANNUALE

String [] nome = { "GIANLUCA", "PAOLO", ... }

String [] cognome = { "AMATO", "ROSSI", ... }

double [] stipendio = { 40.000, 38.000, ... }

Se vogliamo scrivere un metodo che operi su questi dati (esempio, stampare tutti i dipendenti) dobbiamo passare come parametro tutti questi array...

Sarebbe comodo aggregare tutte le informazioni di una persona in un oggetto unico.

PASCAL RECORD

C STRUCT

JAVA CLASSE

STATIC non c'è!!

```
public class Persona {  
    String nome;  
    String cognome;  
    int eta;  
    double stipendio;  
}
```

nome della classe

ATTRIBUTI

CAMPI

TIPICI DEGLI ATTRIBUTI

PUBLIC Potremmo metterlo oppure no.

Una classe è un nuovo tipo di dato, che possiamo usare come quelli che conosciamo già. (più o meno).

Cosa fare con il nuovo tipo Persona?

- Creare una variabile di tipo persona

Persona p;

- Creare un oggetto di tipo Persona

new Persona ()

→ Nell'oggetto creato da new tutti i campi assumono il valore di default.

- Accedere ad un attributo di un oggetto di tipo persona

Persona p;

p.name

p.cognome

p.ete

p.stipendio

Attributi dell'oggetto

Oggetto di tipo Persona

↓

|           |           |
|-----------|-----------|
| Nome      | Giulio    |
| Cognome   | Anzani    |
| Età       | 22        |
| Stipendio | 1.000.000 |

```

→ Persona p;
→ p = new Persona();
→ p.cognome = "Amato";
→ p.nome = "Gianluca";
→ p.eta = 22;
→ p.stipendio = 1_000_000;
→ stampaPersona(p);
→ promozione(p);
→ stampaPersona(p);

```

```

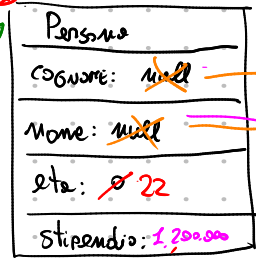
→ Persona p3 = p;
→ p3.nome = "Fabio";
→ stampaPersona(p);

```

PILA RECORD DI ATTIVAZIONE

p: null  
p3:

HEAP



"Amato"

"Gianluca"

"Fabio"

p3 non è una copia di p,  
ma un "alias", cioè un altro  
nome per lo stesso oggetto.

```

public static void promozione(Persona pers) {
    pers.stipendio = 1.2 * pers.stipendio;
}

```

```

public static void prova(double x) {
    x = 1.2 * x;
}

```

```

Run | Debug
public static void main(String[] args) {

```

```

→ double d = 3.4;
→ System.out.println(d);
→ prova(d);
→ System.out.println(d);

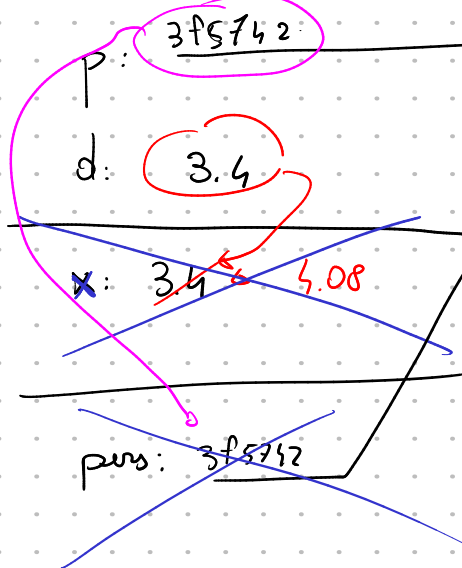
```

```

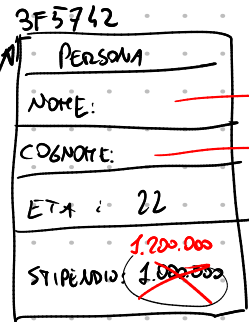
Persona p;
p = new Persona();
p.cognome = "Amato";
p.nome = "Gianluca";
p.eta = 22;
p.stipendio = 1_000_000;
stampaPersona(p);
promozione(p);
stampaPersona(p);

```

PILA RECORD ATTIVAZ.



HEAP



"Amato"

"Gianluca"

```

Persona[] persone;

```

```

persone = new Persona[3];

```

```

persone[0] = new Persona("Gianluca", "Amato", 22, 1.000.000);

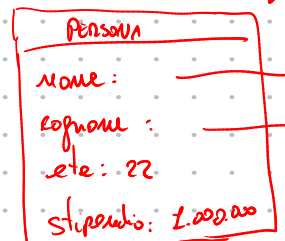
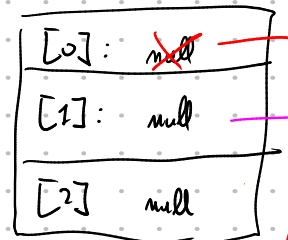
```

```

persone[1] = new Persona();

```

persone: null



persone [2] = persone [1];

