UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Lezione 9 - Classi e orientamento agli oggetti

(Parte 1)

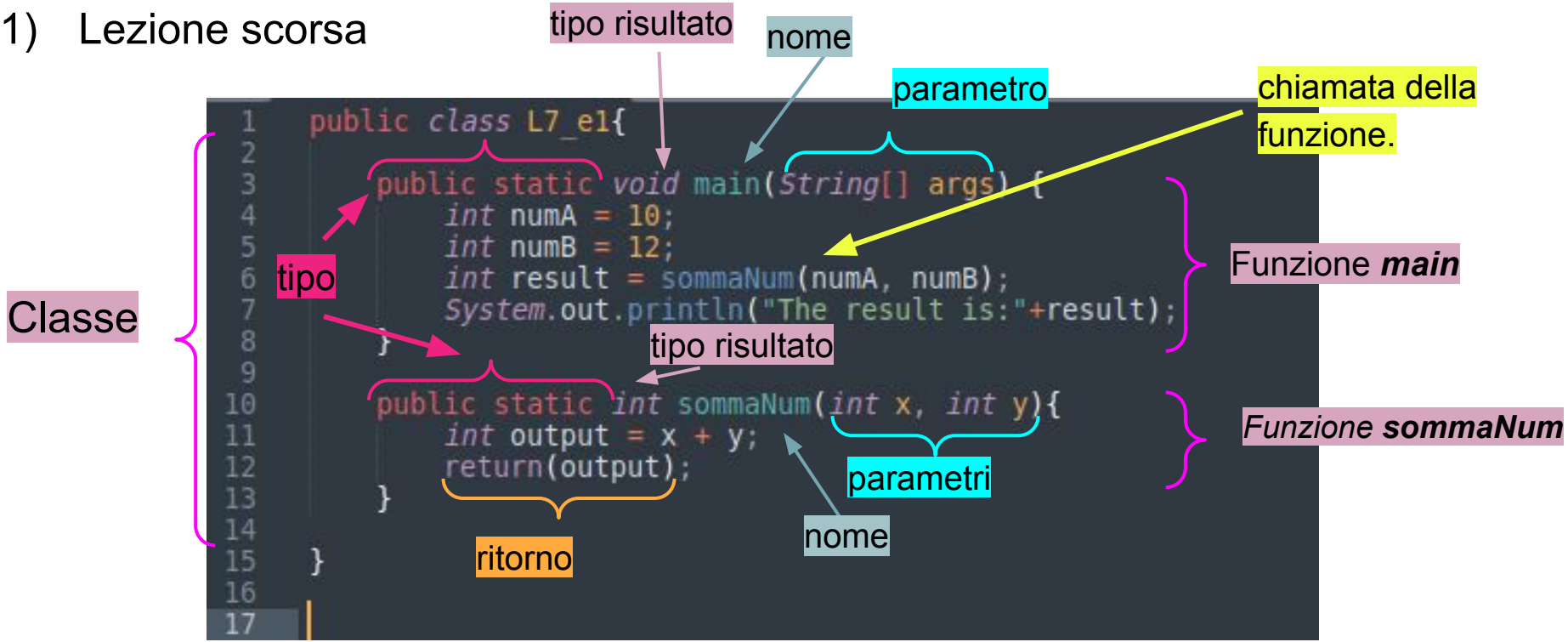
Sylvio Barbon Junior
sylvio.barbonjunior@units.it

Sommario:

1) Lezione scorsa;
2) Classi, variabili e oggetti;
3) Visibilità;
4) Distruttori;
5) Esempio;

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

1) Lezione scorsa



Sylvio Barbon Junior - sylvio.barbonjunior@units.it

2) Classi, variabili e oggetti

- Una classe è un modello che **descrive** una certa categoria di oggetti.
- Essa comprende un certo numero di membri:
  - variabili membro o campi dati;
  - funzioni membro o metodi;
  - costruttori (metodo chiamato per la creazione degli oggetti)

```
class MiaClasse{
     int n;
     MiaClasse(){
          /* ... */
     }
     int fai(int a){
     /* ... */
     }
}
```

dichiarazione della classe

dichiarazione della variabile

dichiarazione del costruttori

dichiarazione del metodo

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

2) Classi, variabili e oggetti

- Una variabile di un tipo classe (brevemente, variabile classe) è un riferimento di oggetti di quella classe;
- Un oggetto di un tipo classe (brevemente, oggetto classe) è un'istanza della classe (oggetto istanza): viene ottenuto per mezzo dell'operatore **new**;

```
MiaClasse varA;

varA = new MiaClasse();

MiaClasse varB = new MiaClasse();
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

2) Classi, variabili e oggetti

classe

metodo costruttore (stesso nome della classe)

```java
public class MiaClasse{

    public MiaClasse(){
        System.out.println("##Costrutore");
    }

    public int getValore(){
        return(10);
    }

    public static int getValoreStatic(){
        return(0);
    }

    public static void main(String args[]){
        System.out.println("#Main 1 \n");
        System.out.println("#"+MiaClasse.getValoreStatic());
        MiaClasse miaClass = new MiaClasse();
        System.out.println("#"+miaClass.getValore());
        System.out.println("#Main \n");
    }
}
```

metodo “non” statico >>> oggetto

metodo statico >>> classe

```
barbon@barbon-PC:~/Scaricati/FI/codici$ java M
#Main 1

#0
##Costrutore
#10
#Main
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

2) Classi, variabili e oggetti

```java
public class Cane{

    String nome;
    int eta;

    public Cane(){

    }

    public Cane(String n){
        nome = n;
        System.out.println("Il cane "+nome);
    }

    public Cane(String n, int e){
        nome = n;
        eta = e;
        System.out.println("Il cane "+nome+" ha "+e);
    }

}
```

```java
public class Main{
    public static void main(String args[]){
        Cane cane1 = new Cane();
        Cane cane2 = new Cane("Chaise");
        Cane cane3 = new Cane("Sky", 4);
    }
}
```

```
barbon@barbon-PC:~/Scaricati/FI/codici$ javac Cane.java
barbon@barbon-PC:~/Scaricati/FI/codici$ javac Main.java
barbon@barbon-PC:~/Scaricati/FI/codici$ java Main
Il cane Chaise
Il cane Sky ha 4
barbon@barbon-PC:~/Scaricati/FI/codici$
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

3) Visibilità
   - L'identificatore della classe e quello dei membri della classe sono sempre visibili all'interno della classe;
   - Un membro di una classe può avere uno dei modificatori:
     - modificatore public : membro visibile alle altre classi;
     - modificatore private: membro non visibile alle altre classi;

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

3) Visibilità

```java
public class Punto{
    private double x, y;

    // costruttore
    public Punto (double a, double b ){
        x = a;
        y = b;
    }

    // metodo: calcola la distanza dall'origine
    double distanza(){
        return Math.sqrt(x*x + y*y);
    }

    // metodo: trasla le coordinate del punto
    // della quantita` t
    void traslazione(double t){
        x += t;
        y += t;
    }
}
```

```java
public class MainPunto{
    public static void main(String[] args){
        double d = Double.parseDouble(args[0]);
        Punto p1 = new Punto(1, 2); // azione 1)
        d = p1.distanza(); // azione 2)
        Punto p2 = p1; // azione 3)
        p2.traslazione(2.5);
        System.out.println(p2.distanza());
    }
}
```

```
barbon@barbon-PC: ~/Scaricati/FI/codici
barbon@barbon-PC:~/Scaricati/FI/codici$ javac Punto.java
barbon@barbon-PC:~/Scaricati/FI/codici$ javac MainPunto.java
barbon@barbon-PC:~/Scaricati/FI/codici$ java MainPunto 4
5.70087712549569
barbon@barbon-PC:~/Scaricati/FI/codici$
```

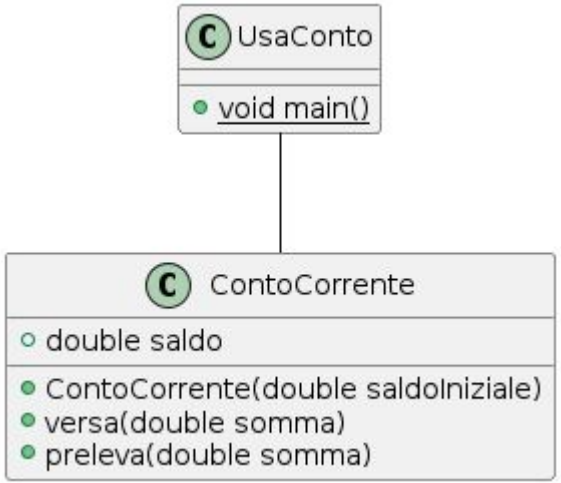

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

4) Distruttori

- In Java **non esistono distruttori** (come per esempio in C++), per cui non avviene (come per la creazione) una distruzione esplicita degli oggetti ad opera del programmatore;
- Il recupero della **memoria dinamica** non più utilizzata viene effettuata dalla piattaforma Java per mezzo di una specifica routine (***Garbage Collector***)

Sylvio Barbon Junior - sylvio.barbonjunior@units.it

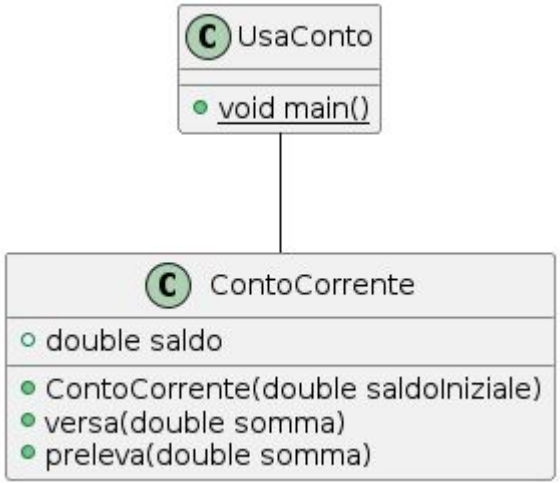
5) Esempi




```java
public class ContoCorrente {

    public double saldo ;

    public ContoCorrente ( double saldoIniziale ) {
        saldo = saldoIniziale ;
    }

    public void versa ( double somma ) {
        saldo += somma ;
    }

    public void preleva ( double somma ) {
        saldo -= somma ;
    }
}
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

5) Esempi




```java
public class UsaConto {
    public static void main ( String [] args ) {
        ContoCorrente cc = new ContoCorrente (1000);

        cc.versa(700);

        if (cc.saldo >200)
            cc.preleva(200);
        if (cc.saldo >900)
            cc.preleva(900);
        System.out.println("Saldo finale:"+cc.saldo);
    }
}
```

```
barbon@barbon-XPS13-9333: ~/Downloads/FI/uml
(base) barbon@barbon-XPS13-9333:~/Downloads/FI/uml$ java UsaConto
Saldo finale:600.0
(base) barbon@barbon-XPS13-9333:~/Downloads/FI/uml$
```


Sylvio Barbon Junior - sylvio.barbonjunior@units.it

Grazie!

Sylvio Barbon Junior - sylvio.barbonjunior@units.it