

```

/* Classe Punto
   Classe derivata PuntoSpazio
*/
// ---- Punto.h
#include <iostream>
using namespace std;

class Punto
{
    double x;
    double y;
public:
    Punto(); // 1° costruttore
    Punto(double s, double t); // 2° costruttore

    double getX();
    double getY();

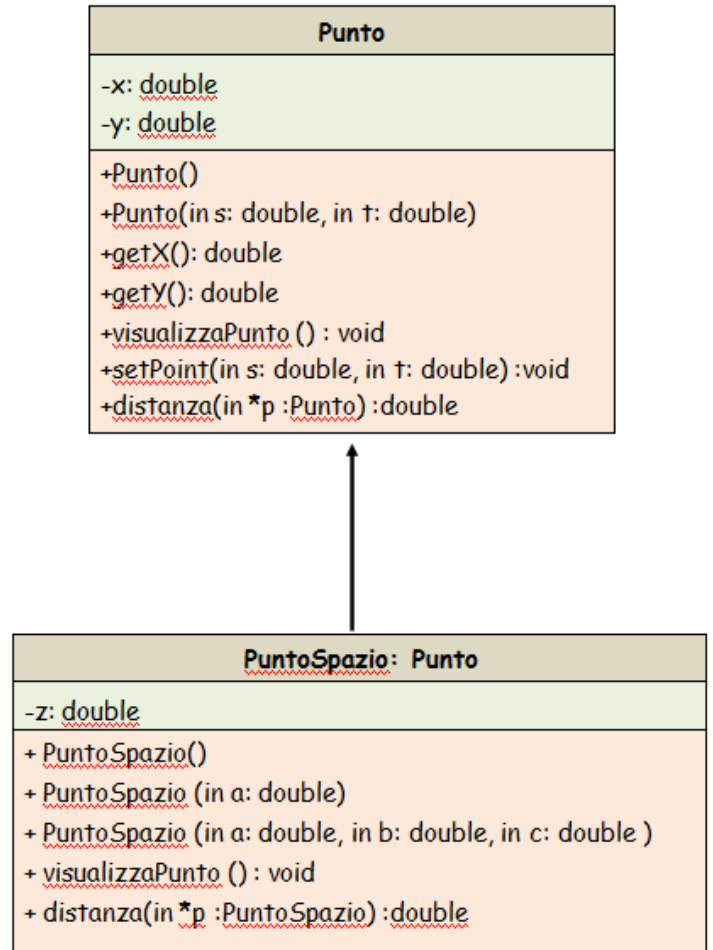
    void visualizzaPunto();
    void setPoint(double s, double t);
    double distanza(Punto *p);
};

// --- Sezione implementativa della classe Punto
Punto::Punto()
{
    x = y = 0;
}
Punto::Punto(double s, double t)
{
    setPoint(s, t);
}
double Punto::getX()
{
    return x;
}
double Punto::getY()
{
    return y;
}
void Punto::visualizzaPunto()
{
    cout << "x = " << x << "\t" << "y = " << y << endl;
}

void Punto::setPoint(double s, double t)
{
    x=s;
    y=t;
}
double Punto::distanza(Punto *p)
{
    return sqrt((x - p->x)*(x - p->x) + (y - p->y)*(y - p->y));
}

```

UML: Diagramma delle classi



```
//---- OOP-PuntoSpazio.cpp
```

```
#include <iostream>
#include <math.h>
#include "Punto.h"
using namespace std;
```

```
class PuntoSpazio : public Punto
```

```
{
    double z;
public:
    PuntoSpazio( )          /* primo costruttore crea un punto nell'origine*/
    {
        setPoint(0,0);
        z=0;
    }
    PuntoSpazio(double a) /* secondo costruttore crea un punto sull'asse delle ascisse*/
    {
        setPoint(a,0);
        z=0;
    }
    PuntoSpazio(double a, double b, double c) /* terzo costruttore */
    {
        setPoint(a,b);
        z=c;
    }
    void visualizzaPunto()
    {
        cout << "x = " << getX() << '\t' << "y = " << getY() << "z = " << z << endl;
    }
    double distanza(PuntoSpazio *P)
    {
        double x=getX(), y=getY();
        return(sqrt( (x - P->getX())*(x - P->getX())+ (y - P->getY())*(y - P->getY()) + (z - P->z)*(z - P->z) ));
    }
}; /* fine classe PuntoSpazio*/
```

```
int main()
```

```
{
    PuntoSpazio P0 = PuntoSpazio(2,3,1); // crea un nuovo punto dello spazio
    cout<<"\nPunto P0 \n";
    P0.visualizzaPunto();

    PuntoSpazio P1 = PuntoSpazio(4,5,6); // crea un nuovo punto dello spazio
    cout<<"\nPunto P1 \n";
    P1.visualizzaPunto();

    cout << "\nDistanza tra i due punti: " << P1.Distanza(&P0) << endl;

    system ("pause");
}
```