

```

1 #include "Point2D.hpp"
2
3 // Costruttori con un numero variabile di argomenti
4 Point2D::Point2D()
5 {
6     std::cout << "Costruisco un punto vuoto\n";
7     x = 0.0;
8     y = 0.0;
9 }
10 Point2D::Point2D(double xCoord, double yCoord)
11 {
12     std::cout << "Costruisco un punto di coordinate"<<xCoord<<","<<yCoord<<"\n";
13     x = xCoord;
14     y = yCoord;
15 }
16
17 // Distruttore
18 Point2D::~Point2D()
19 {
20     std::cout << "Distruggo un punto"<< *this << "\n";
21 }
22
23 // Costruttore di copia
24 Point2D::Point2D(const Point2D& other)
25 {
26     std::cout << "Copio il punto" << other << "\n";
27     x = other.x;
28     y = other.y;
29 }
30 Point2D& Point2D::operator=(const Point2D& other)
31 {
32     std::cout << "Copio il punto tramite assignment operator" << other << "\n";
33     x = other.x;
34     y = other.y;
35     return *this;
36 }
37
38 // Metodi per l'accesso alle coordinate
39 double Point2D::getX()
40 {
41     return x;
42 }
43 double Point2D::getY()
44 {
45     return y;
46 }
47
48 // Operatori somma e differenza di punti nel piano
49 Point2D operator+(const Point2D& A, const Point2D& B)
50 {
51     return Point2D(A.x+B.x, A.y+B.y);
52 }

```

```

53 Point2D operator-(const Point2D& A, const Point2D& B)
54 {
55     return Point2D(A.x-B.x, A.y-B.y);
56 }
57
58 // Metodo per il calcolo della norma
59 double Point2D::norm() const
60 {
61     std::cout << "Calcolo la norma di "<< *this << "\n";
62     return std::sqrt(x*x+y*y);
63 }
64
65 // Metodi di confronto
66 bool operator==(const Point2D& A, const Point2D& B)
67 {
68     return ((std::abs(A.x-B.x)<TOL) and (std::abs(A.y-B.y)<TOL));
69 }
70 bool operator<(const Point2D& A, const Point2D& B)
71 {
72     if(A.x < B.x) return true;
73     if(A.y < B.y) return true;
74     return false;
75 }

```