

Contrôle de Programmation C++

Durée : 0h30

Aucun document autorisé

Nom :

Prénom :

On possède la classe suivante pour représenter les points du plan cartésien.

```
class Point {
private:
    double x, y;
public:
    Point(const double a=0, const double b=0) : x(a), y(b) {}
    double getX() const;
    double getY() const;
    void setX(const double x);
    void setY(const double y);
    double distance(const Point &p) const;
    friend std::ostream&operator <<(std::ostream &f, const Point &p);
};
```

Notez que l'implémentation de ces méthodes ne vous est pas donnée, car inutile. Le nom de chaque méthode est suffisamment significatif.

On souhaite représenter des polygones dans le plan cartésien. On rappelle qu'un polygone est constitué d'une suite finie de points du plan appelés sommets et de segments, appelés côtés, reliant les couples de sommets consécutifs ainsi que d'un segment reliant le premier et le dernier point.

► 1. Écrivez en C++ la classe `Polygone` avec :

- une variable privée `sommets` type vecteur de `Point` ;
- un constructeur pour initialiser l'objet courant ;
- l'accessor `getSommet` qui renvoie le i ème sommet du polygone courant ;
- le mutateur `setSommet` qui modifie le i ème sommet du polygone courant ;
- la fonction `périmètre` qui renvoie le périmètre du polygone courant ;
- la surcharge de l'opérateur `<<` pour écrire sur un `ostream` un polygone au format $[(x_1, y_1)(x_2, y_2) \dots (x_i, y_i) \dots (x_n, y_n)]$.
