EXAMEN de TP

Documents autorisés : notes de cours et de TP.
Durée : 2 heures

Le barème est donné à titre indicatif et est susceptible de modifications.

Organisation de votre compte :
Créer un dossier pour chaque question. A la fin de l'examen, vous ferez une archive de ces dossiers que vous déposerez sur moodle.
Vous pouvez utiliser le programme `genere` de M Rodin (et c'est fortement conseillé).

Question 1 (5 points) : la classe `Point`.

Écrire une classe `Point` permettant de gérer les coordonnées (des `double`) d'un point. Cette classe devra comporter :

- un constructeur par défaut,
- un constructeur ayant deux doubles en paramètres,
- l'opérateur d'affichage (opérateur <<) qui affiche les coordonnées entre parenthèses et séparées par une virgule,
- les opérateurs de copie et d'affectation (operateur =, constructeur en fonction d'un point),
- les accesseurs (get) et les modificateurs (set),
- les tests d'égalité et d'inégalité (les opérateurs == et !=)
- le destructeur.

Écrire un main de test, ainsi qu'un Makefile, et tester cette classe.

Question 2 (5 points) : la classe `Figure`

Écrire une classe `Figure` caractérisée par ses coordonnées, représentées sous la forme d'un `vector` de `Point`. Par exemple le `vector` de points {(0,0),(0,5),(5,5)} définit un triangle partant du point (0,0) et ayant pour traits le segment de (0,0) à (0,5) ainsi que le segment de (0,5) à (5,5) ainsi que le segment de (5,5) à (0,0). Cette classe devra comporter :

- un constructeur par défaut,
- l'opérateur d'affichage (opérateur <<) qui affiche la phrase « les points de la figure sont « , suivie des coordonnées de chaque point formant la figure,
- les opérateurs de copie et d'affectation (operateur =, constructeur en fonction d'une `Figure`),
- les tests d'égalité et d'inégalité (les opérateurs == et !=) : on testera seulement si les `Point` placés au même endroit dans le vector sont égaux, c'est à dire que par exemple {(0,0),(0,5),(5,5)} sera différent de {(0,5),(5,5),(0,0)},
- une méthode d'ajout d'un `Point` en fin du `vector`,
- la même chose mais avec deux doubles en paramètres,
- le destructeur.

Écrire un main de test, ainsi qu'un Makefile, et tester cette classe.

Question 3 (5 points) : héritage avec les classes `Quadrilatere` et `Triangle`

Écrire une classe `Quadrilatere` dérivant de la classe `Figure`, caractérisée par le fait qu'elle ne contient que 4 points. Cette classe doit comporter :

- un constructeur par défaut,
- un constructeur ayant en paramètres 4 `points`,
- le destructeur,
- l'opérateur << pour l'affichage, qui affiche le mot « Quadrilatere : » suivi de l'affichage de la `Figure`.

Ecrire une classe `Triangle` dérivant de la classe `Figure`, caractérisée par le fait qu'elle ne comporte que 3 points. Cette classe doit comporter :

- un constructeur par défaut,
- un constructeur ayant en paramètres 3 `points`,
- l'opérateur << pour l'affichage, qui affiche le mot « Triangle : » suivi de l'affichage de la `Figure`.
- le destructeur,

Écrire un main de test, ainsi qu'un Makefile, et tester ces classes.

Question 4 (5 points) : union et déplacement

Écrire un (ou plusieurs) opérateur(s) + permettant de réunir deux figures (ces figures pouvant être des `Figure` ou des `Quadrilatere` ou des `Triangle`, on doit pouvoir réunir un `Quadrilatere` et un `Triangle`, une `Figure` et un `Triangle`, deux `Quadrilatere`, etc.) : les coordonnées des points de la deuxième figure seront ajoutées aux coordonnées de la première figure.
Écrire une (ou plusieurs) méthode(s) ayant deux doubles x et y en paramètres et permettant de déplacer une figure (cette figure pouvant être une `Figure` ou un `Quadrilatere` ou un `Triangle`) : les coordonnées des points de la figure seront déplacés de x en abscisse et de y en ordonnée.

Par exemple, le main suivant

```
int main() {
   Quadrilatere q(Point(0,0),Point(5,0),Point(5,5),Point(0,5));
   Triangle t(Point(-0.5,5),Point(2.5,7),Point(5.5,5));
   Figure maison=q+t;
   cout << q << endl;
   cout << t << endl;
   cout << maison << endl;
   maison.deplace(20,0);
   cout << maison << endl;
}
```

doit donner

```
Quadrilatere : les points de la figure sont (0,0) (5,0) (5,5) (0,5)
Triangle : les points de la figure sont (-0.5,5) (2.5,7) (5.5,5)
les points de la figure sont (0,0) (5,0) (5,5) (0,5) (-0.5,5) (2.5,7)
(5.5,5)
les points de la figure sont (20,0) (25,0) (25,5) (20,5) (19.5,5)
(22.5,7) (25.5,5)
```