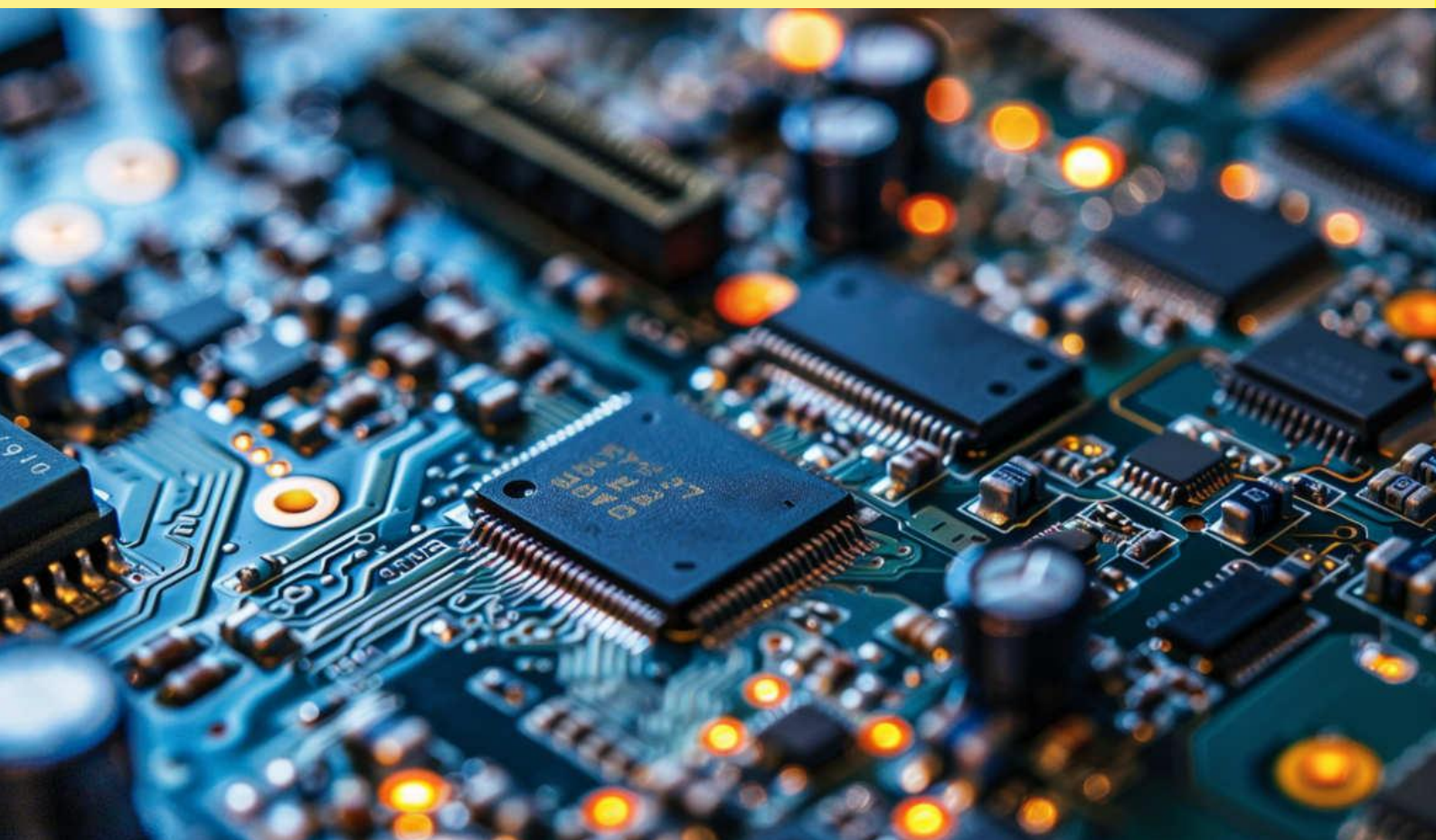


CPI A2 – PROJET P00

PROSIT 4 – A POINTS



CONTEXTE :

Cet exercice vise à améliorer un programme qui calcule la distance euclidienne entre des points et gère des parcours en 2D ou 3D. Actuellement, le code fonctionne mais il y a des redondances entre les classes Point2D/Point3D et Parcours2D/Parcours3D, ce qui complique sa maintenance et son extension. L'objectif est de rendre le programme plus flexible, notamment pour simplifier la conversion entre points 3D et 2D.

INFORMATIONS IMPORTANTES :

```
void main(void)
{
    int pause;

    CLpoint* p1;
    CLpoint* p2;
    CLpoint* p3;
    CLParcours* parcours;

    p1 = new CLpoint(0.0, 0.0);
    p2 = new CLpoint(1.0, 1.0);
    p3 = new CLpoint(2.0, 2.0);
    parcours = new CLparcours2D(3);

    parcours->ajouterPoint(p1);
    parcours->ajouterPoint(p2);
    parcours->ajouterPoint(p3);

    cout << parcours->calculDistance() << endl;
    parcours->message();

    p1 = new CLpoint3D(0.0, 0.0, 0.0);
    p2 = new CLpoint3D(1.0, 1.0, 1.0);
    p3 = new CLpoint3D(2.0, 2.0, 2.2);
    parcours = new CLparcours3D(3);

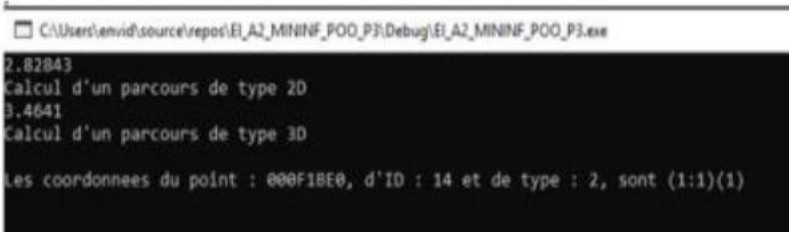
    parcours->ajouterPoint(p1);
    parcours->ajouterPoint(p2);
    parcours->ajouterPoint(p3);

    cout << parcours->calculDistance() << endl;
    parcours->message();

    p1 = new CLpoint(0.0,0.0);
    p2 = new CLpoint3D(1.0, 1.0, 1.0);

    (p1 = p2)->afficherCoordo();

    cin >> pause;
}
```



<https://openclassrooms.com/fr/courses/7137751-programmez-en-orientee-objet-avec-c/7710051-mettez-en-oeuvre-le-polymorphisme>

MOTS INCONNUS :

Polymorphisme : Le polymorphisme permet d'améliorer l'organisation et la lisibilité du code aussi bien que la création de programmes *extensibles* que l'on peut faire croître

non seulement pendant la création originelle du projet, mais également quand de nouvelles caractéristiques sont souhaitées.

Méthode virtuelle : Par défaut, C++ correspond à un appel de fonction avec la définition de fonction correcte lors de la compilation. Il s'agit de la *liaison statique*. Vous pouvez indiquer que le compilateur correspond à un appel de fonction avec la définition de fonction correcte lors de l'exécution ; il s'agit de la *liaison dynamique*. Vous déclarez une fonction avec le mot clé `Virtual` si vous souhaitez que le compilateur utilise une liaison dynamique pour cette fonction spécifique.

Une fonction virtuelle est une fonction membre dont vous souhaitez la redéfinition dans des classes dérivées. Lorsque vous faites référence à un objet de classe dérivée à l'aide d'un pointeur ou d'une référence à la classe de base, vous pouvez appeler une fonction virtuelle pour cet objet et exécuter la version de la classe dérivée de la fonction.

Les fonctions virtuelles garantissent que la fonction correcte est appelée pour un objet, quelle que soit l'expression utilisée pour créer l'appel de fonction.

PROBLEMATIQUE :

Comment pouvons-nous, en manipulant les héritages et le polymorphisme, résoudre le problème de redondance dans le code et correspondre ainsi au main proposé.

PLAN D'ACTION :

Comprendre le concept de polymorphisme [60min]

Modification du code de manière à éviter les tâches répétitives [90min]

REALISATION :

```
2.82843  
CLparcours: 3 points.  
3.4641  
CLparcours3D: 3 points 3D.  
Point(1, 1)
```

