* ***Partie 1 : Retour sur le cour :***
* **Question 1 :**

Qu’est-ce qu’une propriété en C# ? Quelles sont les différentes façons de les déclarer ? Quelles sont les différentes possibilités offertes par chacune d’elles ? (5 %)

* **Réponse :**

**Une propriété est un membre qui fournit un mécanisme flexible pour lire, écrire ou calculer la valeur d'un champ . Les propriétés peuvent être utilisées comme si elles étaient des membres de données publiques, mais ce sont en fait des méthodes spéciales appelées accesseurs . Cela permet d'accéder facilement aux données tout en favorisant la sécurité et la flexibilité des méthodes.**

* **Différentes façon de les déclarer :**
* **Facon 1 :**

privateint champ;  
publicint propriété 1**{**  
            **get { return** champ ; **}**  
**}**

cet accesseur indique que la propriété est en lecture et doit renvoyer un résultat dont le type doit être le même que celui de la propriété.

* **Facon 2 :**

privateint champ;  
public **Int** propriété 2 **{**  
            **set {** champ = ***value*** ; **}**  
**}**

cet accesseur indique que la propriété est en écriture et sert à initialiser ou à modifier la propriété. La propriété propriété 1 est déclarée en écriture seule et stocke une donnée de même type qu'elle dans la variable champ

* **Facon 3 :**

**private int** champ;  
**public int** propriété3**{**  **get { return** champ ; **}**  
**set {** champ = ***value*** ; **}**  
**}**

propriété 3 en lecture et écriture

* **Question 2 :**
* **Question 3 :**
* **Réponse :**

**Le diagramme de classe permet de :**

1. **Faire une représentation graphique de code ou de problème en question afin de détailler toutes les étapes en étude**
2. **Représenter les classe, relations entre les classes et les interfaces**

* **Gestion ressources :**
* **Question 1  :**

Lisez les algorithmes « *Collecter Bois* », « *CollecterMinerai* » et « *CollecterCereales* » de la classe« *MorceauTerrain* » et expliquez leur fonctionnement.

* **Réponse :**

Les trois méthodes prennent un variable s’appelle quantité collecté en paramètre , elle comparent ce variable avec la (quantité de bois, quantité minerai, quantité céréales) pour chaque méthode si la quantité collecte supérieur à la quantité convenable de ces trois quantités ,la quantité collecté pend la valeur de cette dernière . si non la méthode diminue la quantité collecte de la quantité correspondante , si la quantité collecté négative elle retourne une exception , en fin chaque une de ces fonction retourne la quantité collecté

* **Question 2 :**

Y a-t-il des classes qui devraient être abstraites et transformées en classes abstraites ou interfaces ? **Si oui, expliquez pourquoi**. (Ici, on ne vous demande pas d’effectuer la modification, seulement de donner votre avis)

* **Réponse :**

**Oui on le class <<unite>> qui permet de transférer un class abstract paracerque comporte seulement la méthode travailler qui est sans implémentation facile a transformer en abstract**