- Définir les constructeurs de recopie des classes Note et Etudiant.

```
public Note(Note n) {...}
public Etudiant(Etudiant e) {...}
```

- Définir le constructeur suivant:

```
public Etudiant(String code, String nom,int dim) {...}
```

- Définir d'autres constructeurs au besoin.

- Redéfinir pour chacune des classes *Module, Note* et *Etudiant* la méthode suivante:

```
public boolean equals(Object obj) {...}
```

- Modifier la méthode -- ***public void*** *ajouter(Note nt)* -- de telle sorte que la note *nt* n'est ajoutée que si elle n'existe pas. Si une Note *x* ayant le même nom de module que *nt* existe déjà alors *nt* n'est pas ajoutée.

- Définir dans la classe *Etudiant* les méthodes suivantes:

```
  public double noteMoyenneG(){//retourne la moyenne des modules d'un étudiant.}

  public static int compare(Etudiant e1, Etudiant e2){//compare e1 et e2 par rapport
  //aux moyennes des modules}

  public boolean dans(double d1,double d2,String nomModule ){// vérifie si la note
  //du module "nomModule" se trouve dans l'intervalle [d1,d2[}

  public boolean dans(double d1,double d2 ){// vérifie si la moyenne
  //des modules se trouve dans l'intervalle [d1,d2[}
```

Soit la classe LesEtudiants définie comme suit:

```
public class LesEtudiants extends LinkedList{
// Cette classe est conçue dans l'objectif de gérer un ensemble d'étudiants

    public LesEtudiants() {//décrire l'effet de ce constructeur

    }
    .....
    .....

}
```

- **Définir un constructeur de recopie.**
- **Définir la méthode toString()**
- **Définir les méthode suivantes:**

```
public boolean add(Object e) { // redéfinir cette méthode pour qu'elle n'accepte
    //pas les doublons (deux objets identiques). La méthode add telle qu'elle est
    //définit dans la classe LinkedList permet l'existence de doublons dans la
    //liste. }

public LesEtudiants lire(File f, String nomModule) throws Exception{

    // cette méthode retourne dans une liste l'ensemble des informations sur les
    //étudiants se trouvant dans le fichier f. Les informations dans le fichier f
    //concernent le module "nomModule" .

    }

public LesEtudiants trier() {

    //retoune une liste triée des étudiants par rapport à la moyenne des modules
    }

    public LesEtudiants[] groupes(String nomModule){

    // cette méthode retourne un tableau de listes qui contiennent des groupes
    //d'étudiants, un groupe dont les notes sont dans l'intervalle [0,10[, un autre
    //groupe dont les notes sont dans l'intervalle [10,15[ et un dernier groupe dont
    //les notes sont dans l'intervalle [15,20]. Les notes sont celles du module
    //nomModule.
    }
```

À savoir:

Outre les méthodes de la classe *LinkedList* vues lors de la séance de cours, les méthodes suivantes sont aussi des méthode de la classe *LinkedList*.

Boolean contains(Object O)
Retourne true si la liste contient l'élément O spécifié.

int indexOf(Object O)
Retourne l'indice de la première occurrence, dans la liste, de l'élément O spécifié, ou -1 si la liste ne contient pas l'élément.

Object set(int index, Object O)
Remplace par l'objet O, l'objet se trouvant sur la liste à la position *index* spécifiée