Genre # nom : string Aliment Loft {abstrait} # identifiant : int # nom : string # dimensions : int[2] # quantiteEnergetique : int # population : LinkedList<Aliment> # position : int[2] # cases : Aliment[][] # representationGraphique : int[3] # limiteNbTours: int # loft : Loft + Loft(string nom, int[2] dim, LinkedList<Aliment> pop, int limTours) + Aliment(int qteEnerg, int[2] pos) {abstrait} + dessiner(): void + afficher(): void + ajouter(Aliment objet) : void + estDetruit(): void + retirer(int id) : void + lancerTour(): void Neuneu Banane {abstrait} + Banane(int qteEnerg, int[2] pos) # nom : string # prenom : string # genre : Genre # habitudesAlimentaires : Aliment[] + Pates(int qteEnerg, int[2] pos) + Neuneu(string nom, string prenom, Genre genre, int qteEnerg, int[2] pos) {abstrait} + seDeplacer() {abstrait} + manger(Aliment bouffe) {abstrait} Chips + seReproduire(): void + Chips(int qteEnerg, int[2] pos) + etreExclu(): void + vivre(): void Biere + Biere(int qteEnerg, int[2] pos) Erratique Eau + Erratique(string nom, string prenom, Genre genre, int qteEnerg, int[2] pos) + Eau(int qteEnerg, int[2] pos) + seDeplacer() : void + manger (Aliment bouffe) : void CordonBleu Cannibale + CordonBleu(int qteEnerg, int[2] pos) + Cannibale(string nom, string prenom, Genre genre, int qteEnerg, int[2] pos) + seDeplacer(): void + manger (Aliment bouffe): void Lapin + Lapin(string nom, string prenom, Genre genre, int qteEnerg, int[2] pos) + seDeplacer(): void + manger (Aliment bouffe) : void Vorace + Vorace(string nom, string prenom, Genre genre, int qteEnerg, int[2] pos) + seDeplacer(): void + manger (Aliment bouffe): void MorganMagnin + MorganMagnin(string nom, string prenom, Genre genre, int qteEnerg, int[2] pos) + seDeplacer(): void + manger (Aliment bouffe): void + seTransforme() : PowerRanger PowerRanger - toursRestants : int + PowerRanger(string nom, string prenom, Genre genre, int qteEnerg, int[2] pos) + seDeplacer(): void + manger (Aliment bouffe) : void + seTransforme(): MorganMagnin