|  |  |  |  |
| --- | --- | --- | --- |
|  | **Année Universitaire : 2015-2016**  **Examen Java** | | |
| **Module** : Conception orientée objet et programmation Java | | | **Documents autorisés** : Non |
| **Enseignants** : Equipe Java | | | **Nombre de pages** : 4 |
| **Date** : 14/06/2016 | | **Heure :** 08:30 | **Durée** : 1h30 |
| **Classes** : 3INFO A, 4INFINI | | | **Session :** Rattrapage |

**Gestion D’un Zoo d’animaux**

Nous souhaitons développer une application simplifiée de gestion d’un Zoo d’animaux. Pour réaliser notre application, nous vous demandons de compléter les classes présentées ci-dessous.

**Les parties à compléter sont numérotées de 1 à 18.**

**Vous devez *mettre dans vos copies le numéro correspondant à chaque code ajouté //TODO XX :***

**Vous devez *mettre TOUTE la méthode à compléter y compris la signature*.**

**Travail demandé :**

En précisant le numéro de chaque partie à ajouter :

1. Complétez la classe **Animal** avec les méthodes **equals, hashCode** et **toString**, deux animaux sont égaux en cas d’égalité de leurs id et origine. // TODO 1 // 2 points
2. Un animal est peut être soit **AnimalHerbivore** ou **AnimalCarnivore**.
3. Complétez la classe **EnsembleAnimaux** regroupantun ensemble d’Animaux dans un **HashSet.** Complétez les méthodes de cette classe sachant qu’elle implémente l’interface **IEnsembleAnimaux**. // TODO 2 -> TODO 12 // 9 points
4. Complétez la classe **Zoo** qui regroupe pour chaque **Secteur** (voir annexe) l’ensemble de ses Animaux dans un **HashMap**. // TODO 13 -> TODO 18 // 9 points

Remarque 1 : On suppose que la classe **Secteur** contient déjà les méthodes **equals, hashCode** et **toString.** Deux Secteurs sont égaux en cas d’égalité de leurs id et nom.

**Remarque** 2 : **Toutes les classes sont présentées dans l’ANNEXE ci-dessous.**

**ANNEXE :**

|  |
| --- |
| **public abstract class Animal {**  **private int id;**  **private int age;**  **private String origine;**  **public Animal() {**  **}**  **public Animal(int id, int age, String origine) {**  **this.id = id;**  **this.age = age;**  **this.origine = origine;**  **}**  **public int getId() {**  **return id;**  **}**  **public void setId(int id) {**  **this.id = id;**  **}**  **public int getAge() {**  **return age;**  **}**  **public void setAge(int age) {**  **this.age = age;**  **}**  **public String getOrigine() {**  **return origine;**  **}**  **public void setOrigine(String origine){**  **this.origine = origine;**  **}**  // TODO 1 - à compléter //  /\* equals, hashCode et toString \*/  (1 point) (0,5 point) (0,5 point)  **}** |
| **public class AnimalCarnivore extends Animal {**  **private int vitesse;**  **public AnimalCarnivore(int vitesse, int id, int age, String origine) {**  **super(id, age,origine);**  **this.vitesse = vitesse;**  **}**  **public int getVitesse() {**  **return vitesse;**  **}**  **public void setVitesse(int vitesse) {**  **this.vitesse = vitesse;**  **}**  **}** | |
| **public class AnimalHerbivore extends Animal {**  **private String typeHerb;**  **public AnimalHerbivore(String typeHerb, int id, int age, String origine) {**  **super(id, age, origine);**  **this.typeHerb = typeHerb;**  **}**  **public String getTypeHerb() {**  **return typeHerb;**  **}**  **public void setTypeHerb(String typeHerb) {**  **this.typeHerb = typeHerb;**  **}**  **}** | |

|  |
| --- |
| **public interface IEnsembleAnimaux {**  **void ajouterAnimal(Animal animal);**  **void supprimerAnimal(Animal animal);**  **boolean chercherAnimal(Animal a);**  **void afficherAnimeaux();**  **List<AnimalHerbivore> getAnimauxHerbivores();**  **TreeSet<Animal> trierAnimauxParOrigine();**  **double moyenneAgeAnimaux();**  **double SommeVitesseAnimauxCarnivores();**  **}** |

|  |
| --- |
| **public class EnsembleAnimaux**// TODO 2 - à compléter // (0,5 point)  **{**  **Set<Animal> animaux;**  **public EnsembleAnimaux() {**  // TODO 3 - à compléter // (0,5 point)  **}**  **@Override**  **public void ajouterAnimal(Animal animal) {**  // TODO 4 - à compléter // (0,5 point)  **}**  **@Override**  **public void supprimerAnimal(Animal animal) {**  // TODO 5 - à compléter // (0,5 point)  **}**  **@Override**  **public boolean chercherAnimal(Animal a) {**  // TODO 6 - à compléter // (0,5 point)  **}**  **@Override**  **public void afficherAnimeaux() {**  // TODO 8 - à compléter //  /\* avec l'API stream \*/ (1 point)  **}**  **@Override**  **public List<AnimalHerbivore> getAnimauxHerbivores() {**  // TODO 9 - à compléter // (1,5 point)  **}**  **@Override**  **public TreeSet<Animal> trierAnimauxParOrigine() {**  // TODO 10 - à compléter Trier par Origine// (1,5 point)  **}**    **public double moyenneAgeAnimaux() {**  // TODO 11 - à compléter//  /\* avec l'API stream \*/ (1,5 point)  **}**  **@Override**  **public double SommeVitesseAnimauxCarnivores() {**  // TODO 12 - à compléter//  /\* avec l'API stream \*/ (1 point)  **}**  **}** |
|  |

|  |
| --- |
| **public class Secteur {**  **private int id;**  **private String name;**  **public Secteur() {}**  **public Secteur(int id, String name) {**  **this.id = id;**  **this.name = name;**  **}**  **public int getId() {**  **return id;**  **}**  **public void setId(int id) {**  **this.id = id;**  **}**  **public String getName() {**  **return name;**  **}**  **public void setName(String name) {**  **this.name = name;**  **}**  **}** |

|  |
| --- |
| **public class Zoo {**  **Map<Secteur, EnsembleAnimaux> map;**  **public Zoo() {**  // TODO 13 - à compléter // (0,5 point)  **}**  **public void ajouterSecteur(Secteur s) {**  // TODO 14 - à compléter // (0,5 point)  **}**  **public void ajouterAnimal(Animal a, Secteur s) {**  // TODO 15 - à compléter // (2 points)  **}**  **public void afficherZoo() {**  // TODO 16 - à compléter // (2 points)  /\* Utiliser Lambda Expression ou API Stream \*/  **}**  **List<Secteur> contientAnimalCarnivore() {**  // TODO 17 - à compléter -// (2 points)  // retourne les secteurs contenant au moins un animal carnivore//  **}**  **public AnimalCarnivore lePlusRapide () {**  // TODO 18 - à compléter// (2 points)  /\* avec l'API stream \*/  /\* Retourne l'animal carnivore le plus rapide \*/  **}**  **}** |