**HAI717I - Programmation par objets (Examen de substitution)**
**30 mars 2022
17h-19h**

Université de Montpellier – Faculté Des Sciences

Master informatique (ICO), géomatique, bioinformatique, Physique numérique

Dans ce sujet, nous élaborons autour des notions de programme et d’émissions de radio.

Une énumération vous est donnée ci-dessous pour décrire les types de récurrences temporelles des émissions.

**public enum TypeRecurrence {hebdomadaire, quotidienne, enSemaine, enWeekEnd}**

Une classe **Personne** et le début d’une classe **Emission** vous sont également données ci-après.

**public class Personne {**

 **private String nom, prenom;**

 **public Personne(String nom, String prenom) {this.setNom(nom); this.setPrenom(prenom);}**

 **public String getNom() {return nom;}**

 **public void setNom(String nom) {this.nom = nom;}**

 **public String getPrenom() {return prenom;}**

 **public void setPrenom(String prenom) {this.prenom = prenom;}**

 **@Override**

 **public String toString() {return "Personne [nom=" + nom + ", prenom=" + prenom + "]";}**

**}**

**public class Emission {**

 **private final String titre;**

 **private Personne producteur, realisateur;**

 **private final TypeRecurrence recurrence;**

 **public Emission(String titre, Personne producteur, Personne realisateur,**

 **TypeRecurrence recurrence) {**

 **this.titre=titre; this.setProducteur(producteur);**

 **this.setRealisateur(realisateur);this.recurrence=recurrence;**

 **}**

 **public Personne getProducteur() {return producteur;}**

 **public void setProducteur(Personne producteur) {this.producteur = producteur;}**

 **public Personne getRealisateur() {return realisateur;}**

 **public void setRealisateur(Personne realisateur) {this.realisateur = realisateur;}**

 **public String getTitre() {return titre;}**

 **public TypeRecurrence getRecurrence() {return recurrence;}**

**}**

**Q1 (0,5)**. Peut-on ajouter l’accesseur **public void setTitre(String t){…}** dans la classe **Emission** ? Justifiez votre réponse.

**Le titre est *final*, donc on ne peut l’initialiser qu’une fois, et ici c’est fait par le constructeur. On ne peut pas ajouter cet accesseur, qui permettrait de changer sa valeur de multiple fois.**

**Q2 (3).** Deux sous-classes d’**Emission** seront considérées : **Fiction** et **Journal**.

Une fiction est une émission de radio en plusieurs épisodes d’une même durée (en minutes) : on stockera dans des attributs le nombre et la durée de ces épisodes.

Un journal est une émission de radio revenant régulièrement et donnant des informations sur l’actualité : on stockera dans des attributs la personne qui présente le journal, la durée du journal (en minutes), et une information permettant de savoir si le journal accueille des invités.

Ecrire pour les sous-classes **Fiction** et **Journal** : leur entête, leur(s) attribut(s), leurs constructeurs avec des paramètres permettant d’initialiser tous leurs attributs.

On admettra pour la suite que les accesseurs de forme **get** et **set** existent pour les attributs de ces deux classes. N’écrivez pas ces accesseurs.

public class Fiction extends Emission {

 private int dureeEpisode, nbEpisode;

 public Fiction(String titre, Personne producteur, Personne realisateur, TypeRecurrence recurrence, int

dureeEpisode, int nbEpisode) {

super(titre, producteur, realisateur, recurrence);

 this.setDureeEpisode(dureeEpisode);

 this.setNbEpisode(nbEpisode);

 }

}

public class Journal extends Emission{

 private Personne presentateur;

 private int duree;

 private boolean présenceInvité;

 public Journal(String titre, Personne producteur, Personne realisateur, TypeRecurrence recurrence,

 Personne presentateur, int duree, boolean présenceInvité) {

super(titre, producteur, realisateur, recurrence);

this.presentateur = presentateur;

this.duree = duree;

 this.présenceInvité = présenceInvité;

 }

}

**Q3 (2,5).** Ecrire dans les trois classes une méthode de signature **public int dureeAnnuelle()** qui retourne la durée annuelle d’une émission.

Pour une fiction, il s’agit du nombre d’épisodes multiplié par la durée.

Pour un journal, la valeur est calculée en multipliant le nombre de fois où a lieu l’émission par sa durée. Par exemple, si l’émission est hebdomadaire, comme il y a 52 semaines dans une année, la valeur sera 52 multiplié par la durée.

**Emission**

public abstract int dureeAnnuelle();
**Fiction**

public int dureeAnnuelle() {return this.dureeEpisode\*this.nbEpisode;}

**Journal**

 public int dureeAnnuelle() {

 if (this.getRecurrence()==TypeRecurrence.hebdomadaire)

 return 52\*duree;

 if (this.getRecurrence()==TypeRecurrence.quotidienne)

 return 365\*duree;

 if (this.getRecurrence()==TypeRecurrence.enSemaine)

 return 52\*5\*duree;

 if (this.getRecurrence()==TypeRecurrence.enWeekEnd)

 return 52\*2\*duree;

 return 0;

 }

**Q4 (2).** Ecrire dans les trois classes une méthode de signature **public** **String toString()** qui retourne une chaîne de caractères incluant :

- Pour une émission : le titre, le producteur, le réalisateur, la récurrence et la durée annuelle.

- Pour une fiction : rien de plus que pour une émission.

- Pour un journal : comme pour une émission en ajoutant le présentateur et la chaîne « présence d’invités » si c’est le cas.

**Emission**

public String toString() {

return "\n\nTitre "+this.getTitre()+

 " \nProducteur "+this.getProducteur()+

 " \nRéalisateur "+this.getRealisateur()+

 " \nRecurrence "+this.getRecurrence()+

 " Duree annuelle "+this.dureeAnnuelle()+"";

}

**Fiction**

On n'écrit pas toString car elle n'affiche rien de plus
**Journal**

 public String toString() {

 String res = super.toString()+"Présentateur "+this.getPresentateur();

 if (présenceInvité) res += "Présence d'un invité";

 return res;

 }

**Q5 (2).** Ecrire l’entête, les attributs et le constructeur avec paramètres d’une classe **ProgrammeRadio**, qui représente le programme annuel d’une radio avec les indications suivantes : le nom de la radio, l’année concernée et une liste d’émissions.

public class ProgrammeAnnuelRadio {

 private int année;

 private String nomRadio;

 private ArrayList<Emission> listeEmissions = new ArrayList<>();

 public ProgrammeAnnuelRadio(int année, String nomRadio) {

 this.année = année;

 this.nomRadio = nomRadio;

 }

}

**Q6 (2).** Ecrire, dans la classe **ProgrammeRadio,** une méthode **public int indexEmissionTitre(String titre)**, qui prend en paramètre une chaîne de caractères représentant le titre d’une émission et retourne l’indice de cette émission dans la liste des émissions, ou -1 si aucune émission ne porte ce titre.

public int indexEmissionTitre(String titre) {

 for (int i=0; i < this.listeEmissions.size(); i++)

 if (this.listeEmissions.get(i).getTitre().equals(titre))

 return i;

 return -1;

}

**Q7 (2).** Ecrire, dans la classe **ProgrammeRadio,** une méthode **public void gestionEmission(Emission e)**, qui prend en paramètre une émission **e**. Cette méthode appelle **indexEmissionTitre.** Si aucune émission de la liste ne porte le même titre que **e** (la méthode **indexEmissionTitre** a retourné -1), **e** est ajoutée. Si une émission de la liste porte le même titre que **e** (la méthode **indexEmissionTitre** a retourné l’indice de cette émission), alors elle est remplacée par **e**. Pour cela vous utiliserez la méthode **public void set(int i, T e)** de la classe **ArrayList<T>** qui remplace l’élément de la liste à l’indice **i** par **e**.

public void gestionEmission(Emission e) {

 int i = this.indexEmissionTitre(e.getTitre());

 if (i!=-1)

 this.listeEmissions.set(i,e);

 else

 this.listeEmissions.add(e);

}

**Q8 (2).** Ecrire, dans la classe **ProgrammeRadio,** une méthode **dureeAnnuelleProducteur**, qui retourne la somme des durées annuelles des émissions réalisées par un producteur passé en paramètre.

 public int dureeAnnuelleEmissionProducteur(Personne p) {

 int res = 0;

 for (Emission e : this.listeEmissions)

 if (e.getProducteur().getNom()==p.getNom()

 && e.getProducteur().getPrenom()==p.getPrenom())

 res+=e.dureeAnnuelle();

 return res;

 }

**Q9 (2).** Ecrire, dans la classe **ProgrammeRadio,** une méthode **producteurEmissionsQuotidiennes**, qui retourne la liste (sans doublons) des producteurs des émissions quotidiennes.

 public ArrayList<Personne> producteurEmissionsQuotidiennes(){

 ArrayList<Personne> res = new ArrayList<>();

 for (Emission e : this.listeEmissions)

 if (e.getRecurrence()==TypeRecurrence.quotidienne

 && !res.contains(e.getProducteur()))

 res.add(e.getProducteur());

 return res;

 }

**Q10 (2).** Ecrire les instructions d’un **main** commençant par :

**Personne maud = new Personne("Maud","Laforet");**

**Personne paul = new Personne("Paul","Jacques");**

**Personne astrid = new Personne("Astrid","Cheminbleu");**

Ce programme **main** doit ensuite :

1. créer un programme radio, puis une fiction de titre « le petit Nicolas » et un journal produits par Maud.
2. afficher la durée annuelle des émissions produites par Maud
3. créer une fiction qui serait une nouvelle version de la fiction de titre « le petit Nicolas » avec une nouvelle durée.
4. Appeler **gestionEmission** pour mettre à jour cette fiction.

//1

ProgrammeAnnuelRadio p = new ProgrammeAnnuelRadio(2022,"France Culture");

Fiction lpn = new Fiction("Le petit Nicolas", maud,

 astrid, TypeRecurrence.quotidienne, 5, 11);

Journal j = new Journal("Le 18h", maud,

 paul, TypeRecurrence.quotidienne,

 new Personne("Jeanne","Chateau"), 20, true);

//2

System.out.println(p.dureeAnnuelleEmissionProducteur(maud));

//3

Fiction lpnv2 = new Fiction("Le petit Nicolas", maud,

 paul, TypeRecurrence.quotidienne, 6, 12);

//4

p.gestionEmission(lpnv2);