

## HAI7171 - Programmation par objets (Examen de substitution)

17 février 2021

17h-19h

Université de Montpellier – Faculté Des Sciences

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

Dans ce sujet, nous élaborons autour des notions d'UE (Unité d'Enseignement dans l'enseignement supérieur). Une UE est décrite par un code (par ex. HAI7171), le semestre où elle a lieu, pair ou impair, et le niveau, licence ou master. Certaines UEs sont validées lors d'examens (contrôle continu, écrit, Travaux pratiques) qui ont lieu lors de deux sessions. Trois énumérations décrivant les types de niveaux, les types de semestres et les types de session et une classe **FormatSession** décrivant le format d'une session d'examen (avec trois coefficients respectivement pour le CC, l'écrit et le TP) vous sont donnés ci-dessous.

```
public enum TypeNiveau {L1, L2, L3, M1, M2}
public enum TypeSemestre {pair,impair}
public enum TypeSession {session1, session2}

public class FormatSession {
    private int coeffCC;
    private int coeffEcrit;
    private int coeffTP;
    public FormatSession(int coeffCC, int coeffEcrit, int coeffTP) {
        this.coeffCC = coeffCC ; this.coeffEcrit = coeffEcrit; this.coeffTP = coeffTP;
    }
    public int getCoeffCC() {return coeffCC;}
    public int getCoeffEcrit() {return coeffEcrit;}
    public int getCoeffTP() {return coeffTP;}
    public void setCoeff(int cc, int ecrit, int tp) {
        if (cc+ecrit+tp==100)
            {this.coeffCC=cc; this.coeffEcrit=ecrit; this.coeffTP=tp;}
        else System.out.println("la somme des coefficients n'est pas égale à 100");
    }
    public void saisie(Scanner clavier) {
        System.out.println("Saisie du coeff de CC");
        int cc = clavier.nextInt();
        System.out.println("Saisie du coeff d'écrit");
        int ecrit = clavier.nextInt();
        System.out.println("Saisie du coeff de TP");
        int tp = clavier.nextInt();
        this.setCoeff(cc, ecrit, tp);
    }
}
```

**Q1 (1 point).** Ecrivez les deux instructions d'un **main** permettant de créer (1) un format de session avec les coefficients 30, 70 et 0 resp. pour le CC, l'écrit et le TP ; (2) un format de session avec les coefficients 0, 100 et 0 resp. pour le CC, l'écrit et le TP.

Réponse Q1.

```
FormatSession fs1 = new FormatSession(30,70,0);
FormatSession fs2 = new FormatSession(0,100,0);
```

Un début de classe **UE** vous est donné ci-dessous.

```
public class UE {
    private String code = "non renseigné";
    private TypeSemestre semestre;
    private TypeNiveau niveau;

    public UE(String code, TypeSemestre semestre, TypeNiveau niveau) {
```

```

        this.setCode(code);
        this.setSemestre(semester);
        this.setNiveau(niveau);
    }
    public String getCode() {return this.code;}
    public void setCode(String code) {this.code = code;}
    public TypeSemestre getSemestre() {return this.semestre;}
    public void setSemestre(TypeSemestre semestre) {this.semestre = semestre;}
    public TypeNiveau getNiveau() {return this.niveau;}
    public void setNiveau(TypeNiveau niveau) {this.niveau = niveau;}
}

```

**Q2 (3 points).** Deux sous-classes d'UE seront considérées : les UE de type stage et les UE théoriques.

Les *UE de type stage* sont des UEs décrites en plus par une durée en mois.

Les *UE théoriques* sont des UEs décrites en plus par un nombre d'heures en présentiel et deux formats de session, pour la session 1 et pour la session 2.

Ecrire pour les sous-classes **UEstage** et **UEtheorique** : leur entête, leur(s) attribut(s), leurs constructeurs avec 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 et que les trois classes disposent d'une méthode **toString**. N'écrivez pas ces méthodes.

Réponse Q2.

```

public class UEstage extends UE{
    private int duree;
    public UEstage(String code, TypeSemestre semestre, TypeNiveau niveau, int duree) {
        super(code, semestre, niveau);
        this.setDuree(duree);
    }
}
public class UEtheorique extends UE{
    private int nbHeuresPresentiel;
    private FormatSession formatSession1;
    private FormatSession formatSession2;
    public UEtheorique(String code, TypeSemestre semestre, TypeNiveau niveau, int nbHeuresPresentiel,
        FormatSession formatSession1, FormatSession formatSession2) {
        super(code, semestre, niveau);
        this.nbHeuresPresentiel = nbHeuresPresentiel;
        this.formatSession1 = formatSession1;
        this.formatSession2 = formatSession2;
    }
}

```

**Q3 (3 points).** Ecrire dans les trois classes une méthode de signature **public void saisie(Scanner clavier)** qui permette de saisir des valeurs pour tous les attributs d'un objet.

Réponse Q3.

```

// Dans UE
public void saisie(Scanner clavier) {
    System.out.println("Saisie du code de l'UE");
    this.setCode(clavier.next());
    System.out.println("Saisie du semestre");
    this.setSemestre(TypeSemestre.valueOf(clavier.next()));
    System.out.println("Saisie du niveau");
    this.setNiveau(TypeNiveau.valueOf(clavier.next()));
}

```

```

// Dans UEstage
    public void saisie(Scanner clavier) {
        super.saisie(clavier);
        System.out.println("Saisie de la durée");
        this.setDuree(clavier.nextInt());
    }

// Dans UE theorique
    public void saisie(Scanner clavier) {
        super.saisie(clavier);
        System.out.println("Saisie du nombre d'heures présentiel");
        this.setNbHeuresPresentiel(clavier.nextInt());
        System.out.println("Saisie du format de la session 1");
        this.formatSession1.saisie(clavier);
        System.out.println("Saisie du format de la session 2");
        this.formatSession2.saisie(clavier);
    }

```

**Q4 (3 points).** Ecrire dans les trois classes une méthode de signature **public int nbCreditsECTS()** qui retourne le nombre de crédits ECTS de l'UE. Cette méthode se comporte différemment dans les deux sous-classes : chaque mois de stage donne droit à 5 crédits ECTS ; chaque tranche complète de 10h d'heures présentielle correspond à 1 crédit ECTS.

Après l'avoir écrite, indiquez si vous avez dû modifier quelque chose dans l'entête de la classe **UE** (expliquer pourquoi cela a été ou non nécessaire). Il y a plusieurs solutions, expliquez simplement votre choix.

Réponse Q4.

```

// Dans UE
    public abstract int nbCreditsECTS(); // et rendre la classe abstraite

// Dans UEstage
    public int nbCreditsECTS() {
        return this.duree*5;
    }

// Dans UE theorique
    public int nbCreditsECTS() {
        return nbHeuresPresentiel/10;
    }

```

**Q5 (0,5 point).** Ecrire la suite de la méthode **main** en créant l'UE théorique HAI717I, qui compte 45h d'enseignement présentiel, avec les deux formats de session créés à la question 1.

Réponse Q5.

```

UE hai717i = new UEtheorique("HAI717I", TypeSemestre.impair, TypeNiveau.M1, 45, fs1, fs2);

```

**Q6 (2 points).** Ecrire l'entête, les attributs et le constructeur avec paramètres d'une classe **Formation** avec les indications suivantes. Une formation est décrite par une identification (chaîne de caractères qui est invariable). Elle contient une liste d'UEs que vous devez créer lors de sa déclaration.

Réponse Q6.

```

public class Formation {
    private final String identification;
    private ArrayList<UE> listeUEs = new ArrayList<>();
    public Formation(String identification) {
        this.identification = identification;
    }
}

```

**Q7 (1,5 points).** Ecrire, dans la classe **Formation**, une méthode **contientCode**, qui prend en paramètre une chaîne de caractères représentant un code et retourne vrai si la formation contient une UE ayant ce code.

Réponse Q7.

```
public boolean contientCode(String code) {
    for (UE c : listeUEs)
        if (c.getCode().equals(code))
            return true;
    return false;
}
```

**Q8 (1,5 points).** Ecrire, dans la classe **Formation**, une méthode **ajouteUE**, qui prend en paramètre une UE, l'ajoute à la liste des UEs si la formation ne contient pas déjà une UE ayant ce code, et affiche un message d'erreur sinon.

Réponse Q8.

```
public void ajouteUE(UE ue) {
    if (this.contientCode(ue.getCode()))
        System.out.println("ce code est déjà présent");
    else
        listeUEs.add(ue);
}
```

**Q9 (1,5 points).** Ecrire, dans la classe **Formation**, une méthode **nbCreditsECTS**, qui retourne la somme des crédits ECTS apportés par la formation.

Réponse Q9.

```
public int nbCreditsECTS() {
    int somme = 0;
    for (UE c : listeUEs)
        somme += c.nbCreditsECTS();
    return somme;
}
```

**Q10 (1,5 points).** Ecrire, dans la classe **Formation**, une méthode **listeUEsemestre**, qui retourne les UEs de la formation ayant lieu un certain semestre passé en paramètre.

Réponse Q10.

```
public ArrayList<UE> listeUEsemestre(TypeSemestre ts){
    ArrayList<UE> res = new ArrayList<>();
    for (UE c : listeUEs)
        if (c.getSemestre()==ts)
            res.add(c);
    return res;
}
```

**Q11 (1,5 points).** Ecrire les instructions permettant de compléter le **main** par : (1) la création d'une formation d'identification M1-ICo, (2) l'ajout à cette formation de l'UE créée en question Q4 puis (3) l'affichage du nombre de crédits ECTS de la formation.

Réponse Q11.

```
Formation f = new Formation("M1-ICo");
f.ajouteUE(hai717i);
System.out.println(hai717i.nbCreditsECTS());
```