

L2 informatique, L2 mathématiques

Examen

Unité M.MIM3A1 : Introduction à la programmation orientée objet
2 h — Tous documents autorisés

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de la copie qu'il cachera par collage après avoir été pointé. Il devra en outre porter son numéro de place sur chacune des copies, intercalaires, ou pièces annexées.

1 Questions de cours (8 points)

Question 1 (2 points) *Décrire ce que sont une interface et une classe abstraite, en insistant sur les différences entre les deux.*

Pour la question 2, on considère les classes

```
class A {  
    private f() { /* définition de f() */ ...}  
    protected g() {...}  
    public h() {...}  
}  
  
class B extends A { public i() {...} }  
  
class C {  
    private A a;  
    public j() {...}  
}
```

Question 2 (3 points) *Parmi les appels suivants, lesquels sont valides ? Pour chaque appel invalide, justifier brièvement.*

1. `this.f()` dans la définition de `g()`
2. `this.f()` dans la définition de `h()`
3. `this.g()` dans la définition de `f()`
4. `this.g()` dans la définition de `h()`
5. `this.h()` dans la définition de `f()`
6. `this.h()` dans la définition de `g()`
7. `this.f()` dans la définition de `i()`
8. `this.g()` dans la définition de `i()`
9. `this.h()` dans la définition de `i()`
10. `a.f()` dans la définition de `j()`
11. `a.g()` dans la définition de `j()`
12. `a.h()` dans la définition de `j()`
13. `a.i()` dans la définition de `j()`

Pour la question 3, on considère la classe

```
abstract A {  
    f () {g(); h();}  
    abstract g();  
    abstract h();  
}
```

Question 3 (3 points) *En utilisant le plus possible la classe A définie ci-dessus, écrire une classe B comprenant deux attributs x et y , de sorte que l'appel `b.f()` sur une instance `b` de B affiche*

Mon premier attribut vaut X.

Mon second attribut vaut Y.

où X est la valeur de l'attribut x dans `b`, et Y celle de y . Avec cette proposition, sur une instance `b` de la classe B, donner l'appel de méthode permettant d'afficher

Mon premier attribut vaut X.

2 Conception orientée objet (12 points)

On souhaite créer un logiciel gérant les coûts d'une entreprise. Les coûts que l'on souhaite gérer sont ceux des salariés (en CDI, en CDD, en intérim), des loyers et des achats ponctuels ; par exemple :

- la salariée Marie est en CDI et son salaire coûte 3 000 € par période de 4 semaines à l'entreprise,
- Jean est en CDD de la semaine 7 à la semaine 24 de 2018, et son salaire coûte en tout 11 500 €,
- Patrick est en CDD de la semaine 11 à la semaine 24 de 2018, et son salaire coûte en tout 9 000 €,
- Léa est en intérim, envoyée par l'agence « Intérim 2018 », les semaines 10 et 11 de 2018, pour un coût total de 2 000 €,
- le loyer de la première agence de l'entreprise coûte 1 400 € par période de 4 semaines, et celui de la seconde agence, 9 000 € par période de 12 semaines,
- le 31 mars 2018 (en semaine 13), l'entreprise a acheté une imprimante pour 2 500 €,
- le 15 avril 2018 (en semaine 15), l'entreprise a changé un système d'alarme pour 9 000 €.

On remarque que ces coûts peuvent être limités à une certaine plage de dates (salaire d'un intérimaire, salaire d'un employé en CDD), récurrents (loyer, salaire d'un employé en CDI) ou ponctuels (achat de matériel par exemple). Dans la suite, on supposera que *toutes les plages de dates sont exprimées comme des ensembles continus de semaines* (donc, de façon générale, comme des intervalles $[i, j]$). On supposera également l'existence d'un type `Date`, avec un constructeur prenant les numéros de jour, mois et année en argument, et une méthode `semaine(Date d)` qui retourne le numéro de semaine d'une date donnée.

L'objectif principal du logiciel est de permettre de calculer des coûts sur des plages de dates données. Par exemple, si l'on considère la plage de date constituée des semaines 10 à 13 de 2018 (incluses, soit 4 semaines), les coûts de l'entreprise sont :

- 3 000 € pour le salaire de Marie,
- $(4 \times 11\,500 / 18)$ € pour le salaire de Jean (la plage demandée contenant 4 semaines, et le coût étant de 11 500 € sur 18 semaines),
- $(3 \times 9\,000 / 14)$ € pour le salaire de Patrick (le coût étant de 9 000 € sur 14 semaines, et Patrick travaillant 3 semaines dans la plage donnée),
- 2 000 € pour le salaire de Léa,
- $(1\,400 + 4 \times 9\,000 / 12)$ € pour les loyers de l'agence,
- 2 500 € pour l'achat de l'imprimante.

Question 4 (1 point) *Proposer une classe `PlageDates` permettant de représenter une plage de semaines. Donner les attributs ainsi que les définitions du constructeur, des accesseurs et d'une méthode permettant de calculer la durée de la plage, en nombre de semaines. Les définitions peuvent être en pseudo-code, en Java, Python, etc., au choix. Pour les méthodes, donner les types de retour et des arguments.*

Question 5 (2 points) *Proposer une interface `CoûtLimité` permettant de représenter les coûts limités à une certaine plage de dates, et fournissant des méthodes pour récupérer la plage de dates et le coût total sur cette plage. Pour cela, donner un diagramme présentant cette interface et ses méthodes, et ses relations avec des classes `Intérimaire` et `SalariéCDD` (sans donner le contenu de ces dernières classes).*

Question 6 (2 points) *Proposer de même une interface pour les coûts récurrents, donnant la périodicité de ces coûts (par exemple, 4 semaines pour le coût du CDI de Marie) et le montant sur une période (par exemple, 3 000€ pour le CDI de Marie).*

Question 7 (1 point) *Proposer de même une interface pour les coûts ponctuels, donnant la date de ces coûts (par exemple, le 31 mars 2018 pour l'achat de l'imprimante), le coût (2 500) et un libellé ("Achat imprimante").*

Question 8 (3 points) *Donner un diagramme de classes (pas de code) présentant au moins cinq classes (Intérimaire, SalariéCDD, SalariéCDI, Loyer, AchatPonctuel), et précisant quelles interfaces elles implémentent. En plus des informations permettant d'implémenter les interfaces, munir ces classes des informations suivantes : le nom de la personne (pour les intérimaires et les salariés en CDD ou CDI) et l'agence d'intérim (pour les intérimaires). En utilisant au besoin des classes additionnelles, faire en sorte que les attributs ne soient pas répétés.*

Question 9 (3 points) *Écrire le pseudo-code (ou le code Java, Python, etc., au choix) d'une classe Budget contenant*

- *des attributs permettant de représenter l'ensemble des coûts limités, des coûts récurrents et des coûts ponctuels de l'entreprise,*
- *une méthode permettant de calculer les coûts de l'entreprise sur une plage de dates donnée,*
- *une méthode permettant de calculer le coût moyen par jour sur une plage donnée, en considérant que les coûts non ponctuels se répartissent uniformément sur les 7 jours d'une semaine.*

Vous pouvez utiliser tous les types définis précédemment, et enrichir si besoin votre diagramme de classes avec de nouvelles classes, interfaces et/ou méthodes.