[](http://lavande.cpe.fr/intranet/images/logo_CPE/logo_CPE_2009_moyenne_def.jpg)

Année universitaire 2015/2016

NOM :…………………………………… PRENOM :……………………………

Consignes relatives au déroulement de l’épreuve

Date : 8 mars 2016

Contrôle de : Module 4ETI de Programmation Objet / Java, 1ère partie (2ème session)

Durée : 45 minutes.

Professeur responsable : Martine BREDA

Documents : non autorisés 🗷

LES **DOCUMENTS**, CALCULATRICES, TELEPHONES PORTABLES ET AUTRES APPAREILS DE STOCKAGE DE DONNEES NUMERIQUES SONT **INTERDITS**.

Les téléphones portables doivent être éteints pendant toute la durée de l’épreuve et rangés dans les cartables.

Les oreilles des candidats doivent être dégagées.

Rappels importants sur la discipline lors des examens

La présence à tous les examens est strictement obligatoire ; tout élève présent à une épreuve doit rendre une copie, même blanche, portant son nom, son prénom et la nature de l’épreuve.

Une absence non justifiée à un examen invalide automatiquement le module concerné.

Toute suspicion sur la régularité et le caractère équitable d’une épreuve est signalée à la direction des études qui pourra décider l’annulation de l’épreuve; tous les élèves concernés par l’épreuve sont alors convoqués à une épreuve de remplacement à une date fixée par le responsable d’année.

Toute fraude ou tentative de fraude est portée à la connaissance de la direction des études qui pourra réunir le Conseil de Discipline. Les sanctions prises peuvent aller jusqu’à l’exclusion définitive du (des) élève(s) mis en cause.

Consignes générales :

Répondre sur ce document.

Lire soigneusement l’énoncé avant de commencer l’exercice.

Pour les instructions Java, la syntaxe doit rester précise et compréhensible.

Pour les questions qui attendent des réponses en français, ne récitez pas le cours de manière théorique mais soyez **synthétique** et **précis**.

**Question 1**

Les situations suivantes posent-elles problème ? Si vous estimez qu'elles vont produire une erreur, expliquer quel sera son type (**erreur de compilation, d'exécution, de logique)**.

|  |  |  |
| --- | --- | --- |
| 1 | Mettre une parenthèse ) à la place d'une accolade } |  |
| 2 | Faire une division par zéro. |  |
| 3 | Ecrire [Système](javascript:void(0);).sortie.écrire( ) au lieu de System.out.println( ) |  |
| 4 | Multiplier deux nombres alors qu'on voulait les soustraire. |  |
| 5 | Dans une arborescence de classes, ne pas implémenter dans les classes dérivées les méthodes qui sont abstraites dans la classe parent. |  |

**Question 2**

Considérons la méthode public static void main(String argv[ ]).   
Dites si les affirmations suivantes sont correctes ou non?

|  |  |
| --- | --- |
| affirmation | correcte? OUI/NON |
| En Java, les arguments de la ligne de commande sont transmis sous la forme d'un tableau de chaînes de caractères. |  |
| La méthode main ne peut retourner qu'un résultat à la fois |  |
| Une classe sans méthode main est compilable mais n'est pas utilisable seule. |  |
| La méthode main ne peut être appelée que si on possède une instance de la classe à laquelle appartient la méthode. |  |

**Question 3 :**

Dites si oui ou non l'instruction Java Date today = new Date();

|  |  |
| --- | --- |
| proposition | oui/non? |
| Déclare une nouvelle variable |  |
| Crée un objet |  |
| Invoque une méthode |  |
| Réalise une opération d'affectation |  |

**Question 4**

On possède une variableString myString;  
La méthode length( ) de la classe String renvoie le nombre de caractères de l'objet.

Quelle est la syntaxe correcte pour l'appel de cette méthode?

|  |  |
| --- | --- |
| proposition | cocher la bonne proposition |
| int longueur = myString.length; |  |
| int longueur = myString.length(); |  |
| int longueur = myString::length; |  |
| int longueur = myString["length"]; |  |

**Question 5**

class ClasseA { double x ; }

class ClasseB extends ClasseA { int x ; }

class ClasseC extends ClasseB { char x ; }

Une méthode de la classe C fait appel à une variable x désignée de la façon suivante.   
Vous identifierez la variable appelée par la classe à laquelle elle appartient.

|  |  |
| --- | --- |
| nom de la variable | sa classe d'appartenance |
| x |  |
| this.x |  |
| super.x |  |
| ((ClasseB) this).x |  |
| ((ClasseA) this).x |  |
| super.super.x |  |

**Question 6**

Quels qualificateur d'accès peuvent s'appliquer à la méthode doSomething () de la classe B ?

class A {

void doSomething(){ }

}

class B extends A {

void doSomething(){}

}

|  |
| --- |
|  |

**Question 7**

class A {

. . .

}

class B extends A {

. . .

}

Dites si les propositions suivantes sont correctes ou non. Expliquez.

|  |  |
| --- | --- |
| A y =new A();  B x= (B)y;  A z=x; |  |
| B x= new B();  A y = x;  B z=(B)y; |  |
| B x= new B();  A y = x;  B z=y; |  |
| A y =new A();  B x= (B)y;  A z=(A)x; |  |

**Question 8**

Dites si les propositions suivantes sont correctes ou non. Expliquez.

|  |  |
| --- | --- |
| Il est possible d'invoquer la méthode toString() sur tous les objets. |  |
| La méthode toString() n'est pas redéfinie dans la classe X.  x une référence sur un objet de classe X. L'appel à System.out.println(x) renvoie une chaîne de caractères correspondant à la valeur de la référence x (soit une adresse mémoire). |  |
| Dans une même classe, les instructions suivantes sont équivalentes :  System.out.println(this) ; System.out.println(this.toString()); |  |
| Dans une même classe, les instructions suivantes sont équivalentes :  System.out.println(toString()) ;  System.out.println(this.toString()); |  |

**Question 9**

Dites si les propositions suivantes sont correctes ou non?

|  |  |
| --- | --- |
| En Java, la classe HashMap<K,V> définit une collection d'éléments, dont les éléments sont de type V  et l'indice d'accès est un entier K qui va de 0 à size()-1 . |  |
| La méthode Collections.sort(List<T>) de la classe Collections permet de trier les éléments de n'importe quelle classe qui implémente l'interface List<T> si T implémente l'interface Comparable |  |
| List<T> maListe = new ArrayList<T>();  T peut être une classe abstraite  T peut être une interface  T ne peut pas être de type primitif. |  |

**Question 10:**

public class E {

String s = "e";

public E(String s) {

this.s = s;

}

public boolean equals(Object o){

if(o == this) return true;

if(o == null) return false;

if(! (o instanceof E)) return false;

E e = (E)o;

if(e.s.equals(s)) return true;

return false;

}

}

public class F extends E {

public F(String s) {

super(s);

}

}

public class testEquals {

public static void main(String[] args) {

String s = "t";

E e = new E("e");

F f = new F("e");

Object g = f;

System.out.println( e.equals(f) ); //(1)

System.out.println( e.equals(g) ); //(2)

System.out.println( e.equals(s) ); //(3)

System.out.println( f.equals(s) ); //(4)

System.out.println( g.equals(f) ); //(5)

}

}

|  |  |  |
| --- | --- | --- |
|  | Qu'affiche le programme ? | Justifier votre réponse |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

**Question 11:** dites si les affirmations suivantes sont correctes ou non?

|  |  |
| --- | --- |
| Si une classe C implémente plusieurs interfaces I1, I2, I3 la classe C ou les classes dérivées de C doivent implémenter toutes les méthodes de toutes les interfaces I1, I2, I3. |  |
| Soit une classe A qui hérite de B qui hérite de C,  C étant une classe abstraite qui implémente une interface I, et A et B étant des classes concrètes, si C n'implémente qu'une partie des méthodes de l'interface I, A et B doivent implémenter à elles deux toutes les méthodes de I qui ne l'ont pas été par C. |  |
| Des classes non liées hiérarchiquement peuvent implémenter la même interface. |  |
| Une interface permet de créer des classes abstraites. |  |

**Question 12**

public class E {

int a;

public E(int a) {

this.a = a;

}

}

public class F extends E {

int b = 1;

public F(int a,int b) {

super(a);

this.b = b;

}

public F(int a) {

super(a);

b = 10;

}

public F() {

super(5);

}

}

Dites si les affirmations sont vraies.

|  |  |
| --- | --- |
| Les deux classes ne compilent pas. |  |
| L'instruction F f = new F( ) ; crée un objet de classe F avec les attributs a et b valant respectivement 5 et 10. |  |
| L'instruction F f = new F(3,5) ; crée un objet de classe F avec les attributs a et b valant respectivement 3 et 5. |  |

**Question 13**

class Animal {

private boolean enVie = true;

private String nom;

public void affronte(Animal other) {

if (mange(other))

other.setNotEnVie();

else if (other.mange(this))

this.setNotEnVie();

}

public void setNotEnVie() {

enVie = false;

}

public boolean mange(Animal other) {

return false;

}

public boolean isEnVie() {

return enVie;

}

public String getNom() {

return nom;

}

protected void setNom(String nom) {

this.nom = nom;

}

}

class Souris extends Animal {

public Souris(String nom) {

setNom(nom);

}

}

class Chat extends Animal {

public Chat(String nom) {

setNom(nom);

}

public boolean mange(Animal animal) {

return isEnVie() && animal instanceof Souris;

}

}

class Tigre extends Chat{

public Tigre(String nom) {

super(nom);

}

public boolean mange(Animal animal) {

return isEnVie() && (super.mange(animal) || !(animal instanceof Tigre));

}

}

Quels messages seront affichés sur l'écran après l'exécution des instructions suivantes ?

Animal s = new Souris("Souris1");

Animal chat1= new Chat("Chat1");

Animal chat2= new Chat("Chat2");

Animal Tigre1 = new Tigre("Tigre1");

Animal Tigre2 = new Tigre("Tigre2");

//on déclare et on initialise un tableau d'"Animal"

Animal[] animals = new Animal[] { m, chat1, chat2, Tigre1, Tigre2 };

Tigre1.affronte(m);

Tigre1.affronte(chat1);

chat2.affronte(Tigre2);

Tigre1.affronte(Tigre2);

for (int i = 0; i < animals.length; i++){

if(animals[i].isEnVie()){

System.out.println(animals[i].getNom() + " est bien vivant.");

}

else{

System.out.println(animals[i].getNom() + " est completement mort");

}

|  |  |
| --- | --- |
| Souris1 est completement mort  Chat1 est bien vivant  Chat2 est bien vivant  Tigre1 est bien vivant  Tigre2 est bien vivant |  |
| Souris1 est completement mort  Chat1 est completement mort  Chat2 est completement mort  Tigre1 est bien vivant  Tigre2 est bien vivant |  |
| Souris1 est completement mort  Chat1 est bien vivant  Chat2 est completement mort  Tigre1 est completement mort  Tigre2 est completement mort |  |
| Souris1 est bien vivant  Chat1 est completement mort  Chat2 est completement mort  Tigre1 est completement mort  Tigre2 est bien vivant |  |