[](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 : 19 janvier 2016

Contrôle de : Module 4ETI de Programmation Objet / Java, 1ère partie (1ère 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 même si elle n'est pas tout à fait exacte.

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

**Question 1**

La définition de la classe Rectangle est la suivante:

public class Rectangle {

private double hauteur, largeur;

public double surface( ){

return hauteur\*largeur;

}

//...

}

Ecrire les instructions qui permettent de créer une instance r de la classe Rectangle et d'en appeler la méthode surface()?

|  |
| --- |
|  |

**Question 2**

La définition de la classe Point est la suivante:

public class Point{

private double x;

private double y;

public void setX(double newX){

x = newX;

}

public void setY(double newY){

y = newY;

}

public double getX(){

return x;}

public double getY(){

return y;}

}

Qu'affichera le programme suivant (cocher la case correspondante)?

Point p1 = new Point();

Point p2 = new Point();

p2.setX(2.5);

p1.setY(4.0);

System.out.println(p1.getX() + ", " + p1.getY());

System.out.println(p2.getX() + ", " + p2.getY());

|  |  |
| --- | --- |
| 2.5, 0.0  4.0, 0.0 |  |
| 2.5, 4.0  2.5, 4.0 |  |
| 0.0, 4.0  2.5, 0.0 |  |
| 2.5, 0.0  0.0, 4.0 |  |
| erreur de compilation |  |
| pas d'affichage |  |

**Question 3 :** Les affirmations ci-dessous sont-elles correctes?

|  |  |
| --- | --- |
| Les affirmations | Correcte : Vrai/Faux? |
| L'héritage définit une relation "A-UN/POSSEDE-UN" |  |
| Une sous-classe hérite du type de sa super-classe. |  |
| La classe parente a accès à tous les membres de ses sous-classes. |  |
| La classe parente connait toutes ses sous-classes. |  |

**Question 4**

Soient 3 classes A, B, C qui respectent les contraintes suivantes:

* La classe `C` est une sous-classe de la classe `B` ainsi que de la classe `A`.
* La classe `B` est une sous-classe de la classe `A`.

|  |
| --- |
| class B extends A { ... }  class C extends B extends A{ ... } |

Expliquez pourquoi la définition n'est pas correcte.

|  |
| --- |
|  |

**Question 5**

Qu'est-ce que la composition? (dites si les affirmations suivantes sont vraies/fausses?)

|  |  |
| --- | --- |
| Affirmations | Correcte : Vrai/Faux? |
| une structure de données |  |
| une méthode pour créer un objet |  |
| un lien du type "A-UN/POSSEDE-UN" |  |
| doter une classe d'un attribut faisant référence à une autre classe |  |
| rien de tout cela |  |

**Question 6**

La classe Livre est définie comme suit :

class Livre {

private int numPages;

public void setNumPages(int numPages){

numPages = this.numPages;

}

public void afficher(int numPages) {

if (numPages > 0) {

System.out.println(this.numPages + " pages");

}

else {

System.out.println("sans page");

}

}

}

Qu'affichera le programme suivant (choisir parmi les propositions données **celle** qui est correcte)?

Livre livre = new Livre();

livre.setNumPages(150);

livre.afficher(120);

|  |  |
| --- | --- |
| proposition | correcte? (oui/non) |
| 0 pages |  |
| 120 pages |  |
| sans page |  |
| erreur de compilation |  |
| pas d'affichage |  |
| 150 pages |  |

**Question 7**

class ClassA {

protected int number;

public ClassA() {

number = 10;

}

public void print() {

System.out.println(getPrefix() + ": " + number);

}

protected String getPrefix(){

return "A";

}

}

class ClassB extends ClassA{

protected int number = 20;

protected String getPrefix(){

return "B";

}

}

class ClassC extends ClassB{

public ClassC() {

super();

}

protected String getPrefix(){

return "C";

}

}

Qu'affichera le programme suivant (cocher parmi les propositions données **celle** qui est correcte)?

ClassC c = new ClassC();

c.print();

ClassB b = new ClassB();

b.print();

ClassA ab = new ClassB();

ab.print();

|  |  |  |  |
| --- | --- | --- | --- |
| proposition | correcte? | proposition | correcte? |
| C: 10  B: 10  B: 10 |  | A: 10  A: 10  A: 10 |  |
| C: 20  B: 20  B: 20 |  | C: 10  B: 10  A: 10 |  |

**Question 8**

Laquelle de ces définitions est valide? Cocher la case correspondante.

|  |  |  |  |
| --- | --- | --- | --- |
|  | valide? |  | valide? |
| class A {  private int attr = 1;  }  class B extends A {  private int c;  public B(int c){  this.c = c;  }  private doSth(){  super();  }  } |  | class A {  private int attr = 1;  }  class B extends A {  private int c;  public B(int c){  this.c = c;  super(c);  }  } |  |
| class A {  private int attr = 1;  }  class B extends A {  private int c;  public B(int c){  this.c = c;  }  } |  | class A {  private int attr = 1;  }  class B extends A {  private A attrA;  public B(int c){  attrA = super();  this.c = c;  }  } |  |

**Question 9**

Dites si les affirmations suivantes sont correctes ou non?

|  |  |
| --- | --- |
| Affirmations | correcte : Vrai/Faux? |
| Une collection permet de stocker un nombre variable d'objets. |  |
| LinkedList et ArrayList sont des classes qui héritent de la classe List. |  |
| Les Collection se redimensionnent dynamiquement au fur et à mesure des ajouts/suppressions d'objets. |  |
| Si vous insérez fréquemment des données en milieu de liste, choisir de préférence une LinkedList. |  |
| Les ensembles sont les seules collections d’éléments qui n’admettent pas les doublons. |  |
| Dans une Map, la clé et la valeur doivent être uniques. |  |

**Question 10:** Qu'affichera le programme suivant (cocher parmi les propositions données **celle** qui est correcte)?

List<Object> al = **new** ArrayList<Object>();

al.add(12);

al.add("Une chaîne de caractères !");

al.add(12.20f);

al.add('d');

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

{

System.out.println("donnée à l'indice " + i + " = " + al.get(i));

}

|  |  |
| --- | --- |
| proposition | correcte? (oui/non) |
| donnée à l'indice 0 = 12.0  donnée à l'indice 1 = U  donnée à l'indice 2 = 12.2  donnée à l'indice 3 = d |  |
| donnée à l'indice 0 = 12  donnée à l'indice 1 = Une chaîne de caractères !  donnée à l'indice 2 = 12.2  donnée à l'indice 3 = d |  |
| une erreur de compilation |  |
| pas d'affichage |  |
| donnée à l'indice 0 = 12  donnée à l'indice 2 = 12.2 |  |

**Question 11:**

List liste = new ArrayList();

for(int i = 0; i < 10; i++) {

liste.add(i);

}

Combien d'objets sont créés dans cette séquence ?

Justifiez votre réponse.

|  |
| --- |
|  |

**Question 12**

Qu'est-ce qu'une variable de classe? Quelles sont ses caractéristiques?

|  |
| --- |
|  |

**Question 13**

La classe Circle est définie comme suit :

class Circle {

private double radius;

public void setRadius(double r){

radius = r;

}

public double getRadius(){

return radius;

}

}

Qu'affichera le programme suivant?

Circle c1 = new Circle();

Circle c2 = new Circle();

c1.setRadius(10.0);

c1 = c2;

c2.setRadius(c1.getRadius());

System.out.println(c1.getRadius() + ", " + c2.getRadius());

|  |  |
| --- | --- |
| proposition | correcte? (oui/non) |
| 0.0, 0.0 |  |
| 0.0, 10.0 |  |
| 10.0, 0.0 |  |
| 10.0, 10.0 |  |
| une erreur de compilation |  |

**Question 14**

Dans l'instructionString string = new String ("string");

|  |  |
| --- | --- |
| **proposition** | **vrai/faux** |
| String est le nom de la variable |  |
| String est le type de la variable |  |
| string est le nom de la variable |  |
| string est le type de la variable |  |
| string est le nom de l’objet créé |  |
| String est une classe |  |

**Question 15 :** Dites si les affirmations suivantes sont vraies ou fausses.

|  |  |
| --- | --- |
| Affirmation | Vrai/Faux ? |
| Une classe peut implémenter plusieurs classes mais ne peut étendre qu’une seule interface. |  |
| Une classe peut implémenter plusieurs classes et peut étendre plusieurs interfaces. |  |
| Une classe peut implémenter plusieurs interfaces mais ne peut étendre qu’une seule classe. |  |
| Une classe doit implémenter une seule interface et étendre une seule classe. |  |

Les questions 1 à 15 sont notées sur 1 point chacune.

**Question 16 (5 pts)**

Le programme ci-dessous affiche un bouton dans une fenêtre. Ce bouton devrait changer de couleur en fonction des événements souris. Lire soigneusement le code et répondre aux questions posées.

public static void main(String[] args) {

Fenetre fen = new Fenetre() ;

fen.setVisible(true) ;

}

class Fenetre extends JFrame{

private JPanel pan = new JPanel();

private JButton bouton= new Bouton("Mon bouton", Color.blue);

public Fenetre(){

this.setTitle("Animation");

this.setSize(300, 300);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setLocationRelativeTo(null);

pan.add(bouton);

this.setContentPane(pan);

}

}

class Bouton extends JButton implements MouseListener{

private Color color ;

public Bouton(String str, Color color){

super (str);

this.color= color;

}

public void paintComponent(Graphics g){

super.paintComponent(g);

this.setBackground(color);

}

public void mouseClicked(MouseEvent event) {

}

public void mouseEntered(MouseEvent event) {

// fond en jaune lors du survol

color= Color.yellow;

repaint();

}

public void mouseExited(MouseEvent event) {

// fond en vert lorsqu'on quitte le bouton

color = Color.green;

repaint();

}

public void mousePressed(MouseEvent event) {

// fond en rouge lors du clic gauche

color = Color.red;

repaint();

}

public void mouseReleased(MouseEvent event) {

// fond en bleu lorsqu'on relâche le clic

color = Color.blue;

repaint();

}

}

1. Pourquoi le résultat n’est pas celui attendu ?

|  |
| --- |
|  |

1. Quelles corrections sont nécessaires? A quel endroit du code?

|  |
| --- |
|  |

1. Pourquoi faut-il écrire à vide la méthode mouseClicked() ?

|  |
| --- |
|  |