



Durée : 1h30

Documents autorisés : Polycopié de cours, TDs et TPs.

Java vs C++

Exercice 1 : Instanciation de tableaux (2 points)

Nous disposons déjà d'une classe Point, voici une suite d'instructions censée effectuer la même chose en Java et en C++ :

En Java	En C++
<code>Point vecteurPoints[] = new Point[10];</code>	<code>Point * vecteurPoints = new Point[10];</code>

Expliquez ce que font ces instructions dans chacun des langages.

Exercice 2 : Constructeurs / Destructeurs (5 points)

- Qu'est ce qu'un constructeur ? (2 point)
 - A quel moment est-il appelé ?
 - Peut-on l'invoquer soi-même comme n'importe quelle autre méthode ?
 - Quel est son type de retour ?
- Pour quelle raison ne définit-on pas toujours de constructeur par défaut lorsqu'on a défini d'autres constructeurs ? (1 point)
- Expliquez les principes de construction / destruction d'instances en Java et C++ ainsi que les mécanismes de gestion mémoire associés. (2 point)

Accessibilité

Exercice 3 : Accessibilité interne & externe (5 points)

Dans cet exercice on se placera dans le contexte du C++ afin de profiter du mode d'héritage à la carte.

Soit les déclarations suivantes :

<pre>class A { public : int pub; protected : int prot; private : int priv; // constructeur par défaut public : A(void); }; class B : public A { // constructeur par défaut public : B(void); };</pre>	<pre>class C : protected A { // constructeur par défaut public : C(void); }; class D : B { // constructeur par défaut public : D(void); }; class E : public D { // constructeur par défaut public : E(void); };</pre>
--	---

La classe A possède 3 attributs respectivement publique, protégé et privé. La classe B hérite de la classe A en mode public. La classe C hérite de la classe A en mode protégé. La classe D

hérite de la classe B avec le mode par défaut. Et enfin, la classe E hérite de la classe D en mode public.

Résumez dans des tableaux (comme celui présenté ci-dessous) :

- Le mode d'accès des trois attributs (pub, prot & priv) **dans** chacune des classes A, B, C, D & E.
- L'accessibilité des trois attributs (pub, prot & priv) **en dehors** de chaque classe A, B, C, D & E. C'est-à-dire : précisez si ces attributs sont accessibles ou pas.

Exemple de tableau :

		Classes				
		A	B	C	D	E
Attributs	pub					
	prot					
	priv					

Torture Test

Exercice 4 : Hiérarchie de classes (5 points)

<pre> public class ClassA { protected int i1; public ClassA(int val) { i1 = val; } public void methode1(int val) { i1+=val; } public void methode2(int val) { i1+=val; methode1(val+1); } public void methode3(int val, ClassA a) { i1+=val; methode1(val+1); methode2(val+2); a.methode1(val+3); a.methode2(val+4); } } </pre>	<pre> public class ClassC extends ClassB { protected int i3; public ClassC() { i3 = 0; } public void methode2(int val) { i3 += val; super.methode2(val+1); } } </pre>
<pre> public class ClassB extends ClassA { protected int i2; public ClassB() { super(1); i2 = 0; } public void methode1(int val) { i2+=val+1; } } </pre>	<p>Indiquez pour chacune des lignes du programme principal ci-dessous :</p> <ul style="list-style-type: none"> • La ou les méthodes déclenchées par l'instruction (par exemple « X.fl(5) » indique que la méthode fl de la classe X a été déclenchée avec la valeur 5). • Le ou les mécanismes utilisés dans le contenu de la ligne. Parmi ceux-ci on peut citer par exemple : la hiérarchie des constructeurs, le polymorphisme d'héritage, le lien dynamique ou encore la surcharge.

Programme principal

<pre>public class TestClassABC { public static void main (String args[]) { ClassA iA = new ClassA(0); ClassB iB = new ClassB(); ClassC iC = new ClassC(); ClassA iAB = new ClassB(); ClassA iAC = new ClassC(); ClassB iBC = new ClassC(); iA.methode1(1); iA.methode2(2); iA.methode3(3,iB); iB.methode1(1); iB.methode2(2); iB.methode3(3,iC); } }</pre>	<pre> iC.methode1(1); iC.methode2(2); iC.methode3(3,iA); iAB.methode1(1); iAB.methode2(2); iAB.methode3(3,iAC); iAC.methode1(1); iAC.methode2(2); iAC.methode3(3,iBC); iBC.methode1(1); iBC.methode2(2); iBC.methode3(3,iAB); } }</pre>
--	---

Exercice 5 : Exceptions (3 points)

En vous basant sur l'exemple ci-dessous, complétez de deux manières différentes la méthode « **methode2** » afin de montrer quelles sont les deux manières de traiter une exception.

```
public class MaClasseExcept
{
    ...
    public void methode1() throws MonException
    {
        ...
    }

    public void methode2() ...
    {
        ...
        methode1();
        ...
    }
}

...
class MonException extends Exception
{
    ...
}
```