L01E\_EXERCICES – MODIFICATEURS D’ACCÈS

Table des matières

[1 Exercices – package module1 2](#_Toc137992684)

[1.1 Classe Employe (module1) 2](#_Toc137992685)

[1.2 Classe Cadre (module1) 2](#_Toc137992686)

[1.3 Classe Application (module1) 3](#_Toc137992687)

[2 Exercices – ajout d’un package module2 3](#_Toc137992688)

[2.1 Classe Afficher (module2) 3](#_Toc137992689)

[2.2 Classe ListeNoms (module2) 4](#_Toc137992690)

[2.3 Classe Application2 (module2) 4](#_Toc137992691)

[2.4 Questions 5](#_Toc137992692)

# Exercices – package module1

Ces exercices visent à pratiquer la compréhension des modificateurs d’accès.

Les exercices présentent un ensemble de packages ( paquets ), de classes et de méthodes pouvant servir à la classification d'employés. Déterminer si les appels aux méthodes peuvent être faits étant donné leur modificateur d'accès et expliquez pourquoi.

## Classe Employe (module1)

|  |
| --- |
| package module1;  public class Employe  {  private String nom;  int age;  public String getNom ()  {  Return this.nom;  }  protected void setNom ( String nom )  {  this.nom = nom;  }  } |

## Classe Cadre (module1)

|  |  |  |  |
| --- | --- | --- | --- |
| package module1;  public class GestionEmploye {  public void gererEmploye(){  Employe employe = new Employe();  employe.setNom("Bob");  employe.getNom( );  }  } | |  | | --- | | 1. Oui |  |  | | --- | | 1. Oui | |

## Classe Application (module1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| package module1;  class Application  {  public static void main ( String [ ] args )  {  Employe technicien = new Employe();  technicien.nom = "Éric";  technicien.age = 30;    technicien.setNom ("Patrick");  technicien.getNom ();    /\* Code … \*/  }  } | |  | | --- | | 3. non |  |  | | --- | | 4. oui |  |  | | --- | | 5. oui |  |  | | --- | | 6. oui | |

# Exercices – ajout d’un package module2

## Classe Afficher (module2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| package module2;  import module1.\*;  public class Afficher {  public void afficherEmploye1{  Employe infos = new Employe();  System.out.println(infos.nom);  System.out.println(infos.age);  System.out.println(infos.getNom());  }  public void getAfficherEmploye2{  System.out.println(getNom());  }  } | |  | | --- | | 7. non |  |  | | --- | | 8. non |  |  | | --- | | 9. oui |  |  | | --- | | 10. non | |

## Classe ListeNoms (module2)

|  |  |  |
| --- | --- | --- |
| package module2;  public class ListeNoms {  public void afficheNom(){  Employe employe = new Employe();  System.out.println(employe.getNom());  }  } | |  | | --- | | 11. non | |

## Classe Application2 (module2)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| package module2;  import module1.\*;  class Application2  {  public static void main ( String [ ] args )  {  Employe technicienne = new Employe();  Employe directrice = new Employe();  technicienne.nom = "Louise";  technicienne.age = 35;  directrice.setNom("Louise");  System.out.println(technicienne.nom);  System.out.println(directrice.getNom());  }  } | |  | | --- | | Non |  |  | | --- | | 13.non |  |  | | --- | | 14. non |  |  | | --- | | 15. non |  |  | | --- | | 16. oui | |

## Questions

1. Dans la vraie vie...

les variables d'instance (celles dont la valeur pouvant varier d'un objet à un autre qui sont des instances de la classe) ont toujours comme modificateur d'accès PRIVATE. Pourquoi ?

|  |
| --- |
|  |

|  |
| --- |
| **IMPORTANT :**   * **Un objet doit être créé pour pouvoir accéder à ses membres (on voit des exceptions à cette règle un peu plus loin).** * **Quel que soit le modificateur d'accès, les import nécessaires doivent toujours être faits pour avoir accès à une donnée ou à une méthode qui fait partie d’un autre package.** |