

POO L3 Interfaces - 2023-2024

Nombre de participants : 15



1. Qu'affiche ce programme à l'exécution ?

5 bonnes réponses
sur 11 répondants

```

J Employee.java X
Q1 > J Employee.java
1 public abstract class Employee {
2     private String nom;
3     private int id;
4     private double rem;
5     private double prime;
6
7     Employee(String n, int id, double r) {
8         this.nom = n;
9         this.id = id;
10        this.rem = r;
11        this.prime = 100;
12    }
13
14    /* Augmentation de la primeAnnuelle
15       calcul: primeAnnuelle * (1+p) */
16    public abstract void bonusPrime(float p);
17
18    @Override
19    public String toString() {
20        return nom + "\n" +
21            "Identifiant: " + id + "\n" +
22            "Remunération: " + rem + "\n" +
23            "Prime annuelle: " + prime + "\n";
24    }
25
26 }

J Main.java X
Q1 > J Main.java
1 public class Main {
2     public static void main(String[] args) {
3         Employee e = new Employee("Albert", 154, 1854);
4         e.bonusPrime(0.10f); // augmentation de 10%
5         System.out.println(e);
6
7     }
8 }
9 }

```

Albert Identifiant:
154 remunération:
1854.0
primeAnnuelle:
110.0

Le programme
affiche une erreur à
l'exécution

Rien: le programme
ne compile pas car
on ne peut pas
instancier une
classe abstraite

Albert Identifiant:
154 remunération:
1854.0
primeAnnuelle:
100.0

Rien: le programme
ne compile pas car
la méthode

bonusPrime n'est
pas définie



2. Qu'affiche ce programme à l'exécution ?

1 bonne réponse
sur 13 répondants

```

J Employee.java
1 public abstract class Employee {
2     private String nom;
3     private int id;
4     private double rem;
5     private double prime;
6
7     Employee(String n, int id, double r) {
8         this.nom = n;
9         this.id = id;
10        this.rem = r;
11        this.prime = 100;
12    }
13
14    /* Augmentation de la primeAnnuelle
15       calcul: primeAnnuelle * (1+p) */
16    public abstract void bonusPrime(float p);
17
18    @Override
19    public String toString() {
20        return nom + "\n" +
21            "Identifiant: " + id + "\n" +
22            "Remunération: " + rem + "\n" +
23            "Prime annuelle: " + prime + "\n";
24    }
25 }

J Informaticien.java
1 public class Informaticien extends Employee {
2
3     Informaticien(String n, int id, double r) {
4         super(n, id, r);
5     }
6
7     @Override
8     public void bonusPrime(float p) {
9         this.prime = this.prime * (1+p);
10    }
11
12    @Override
13    public String toString() {
14        return "Informaticien: " + super.toString();
15    }
16 }

J Main.java
1 public class Main {
2     public static void main(String[] args) {
3         Employee e = new Informaticien("Albert", 154, 1854);
4         e.bonusPrime(0.10f); // augmentation de 10%
5         System.out.println(e);
6     }
7 }

```

Le programme
affiche une erreur à
l'exécution

Informaticien:
Albert Identifiant:
154 Remunération:
1854.0 Prime
annuelle: 110.0

Rien: le programme
ne compile pas car
une variable
Employee ne peut
référencer un objet
Informaticien

Rien: le programme
ne compile pas car
l'attribut prime
n'est pas visible
depuis la classe
Informaticien

Informaticien:
Albert Identifiant:

154 Remunération:
1854.0 Prime
annuelle: 100.0

Le mot-clef this pour accéder à l'attribut prime a été remplacée par



3. le mot-clef super. Après cette modification, qu'affiche ce programme à l'exécution ?

6 bonnes réponses
sur 15 répondants

```

Employee.java
1 public abstract class Employee {
2     private String nom;
3     private int id;
4     private double rem;
5     private double prime;
6
7     Employee(String n, int id, double r) {
8         this.nom = n;
9         this.id = id;
10        this.rem = r;
11        this.prime = 100;
12    }
13
14    /* Augmentation de la primeAnnuelle
15       calcul: primeAnnuelle * (1+p) */
16    public abstract void bonusPrime(float p);
17
18    @Override
19    public String toString() {
20        return nom + "\n" +
21            "Identifiant: " + id + "\n" +
22            "Remunération: " + rem + "\n" +
23            "Prime annuelle: " + prime + "\n";
24    }
25
26 }

Informaticien.java
1 public class Informaticien extends Employee {
2
3     Informaticien(String n, int id, double r) {
4         super(n, id, r);
5     }
6
7     @Override
8     public void bonusPrime(float p) {
9         super.prime = super.prime * (1+p);
10    }
11
12    @Override
13    public String toString() {
14        return "Informaticien: " + super.toString();
15    }
16 }

Main.java
1 public class Main {
2     public static void main(String[] args) {
3         Employee e = new Informaticien("Albert", 154, 1854);
4         e.bonusPrime(0.10f); // augmentation de 10%
5         System.out.println(e);
6     }
7 }

```

Le programme
affiche une erreur à
l'exécution

Informaticien:
Albert Identifiant:
154 Remunération:
1854.0 Prime
annuelle: 110.0

Rien: le programme
ne compile pas car
une variable
Employee ne peut
référencer un objet
Informaticien

Rien: le programme
ne compile pas car
l'attribut prime
n'est pas visible
depuis la classe
Informaticien

Informaticien:
Albert Identifiant:
154 Remunération:
1854.0 Prime
annuelle: 100.0



4. Qu'affiche ce programme à l'exécution ?

7 bonnes réponses
sur 13 répondants

```

J Employee.java
1 public abstract class Employee {
2     private String nom;
3     private int id;
4     private double rem;
5     private double prime;
6
7     Employee(String n, int id, double r) {
8         this.nom = n;
9         this.id = id;
10        this.rem = r;
11        this.prime = 100;
12    }
13
14    public double getPrime() {
15        return prime;
16    }
17
18    public void setPrime(double p) {
19        if (p > 0 && p < 10000) prime = p;
20    }
21
22    /* Augmentation de la primeAnnuelle
23       calcul: primeAnnuelle * (1+p) */
24    public abstract void bonusPrime(float p);
25
26    @Override
27    public String toString() {
28        return nom + "\n" +
29            "Identifiant: " + id + "\n" +
30            "Remunération: " + rem + "\n" +
31            "Prime annuelle: " + prime + "\n";
32    }
33
J Informaticien.java
1 public class Informaticien extends Employee {
2
3     Informaticien(String n, int id, double r) {
4         super(n, id, r);
5     }
6
7     @Override
8     public void bonusPrime(float p) {
9         setPrime(getPrime() * (1.0+p));
10    }
11
12    @Override
13    public String toString() {
14        return "Informaticien: " + super.toString();
15    }
16 }
J Main.java
1 public class Main {
2     public static void main(String[] args) {
3         Employee e = new Informaticien("Albert", 154, 1854);
4         e.bonusPrime(0.10f); // augmentation de 10%
5         System.out.println(e);
6     }
7 }

```

Le programme
affiche une erreur à
l'exécution

Informaticien:
Albert Identifiant:
154 Remunération:
1854.0 Prime
annuelle: 100.0

Rien: le programme
ne compile pas car
la méthode
bonusPrime n'a pas
été définie dans la
classe Employee

Rien: le programme
ne compile pas car
la méthode
getPrime n'est pas
définie

Informaticien:
Albert Identifiant:
154 Remunération:

1854.0 Prime
annuelle: 110.0



5. Qu'affiche le programme suivant à l'exécution ?

4 bonnes réponses
sur 12 répondants

```

J Vehicule.java X
1 import java.util.Date;
2
3 interface Vehicule {
4     public double getVitesseMax();
5     public String getImmatricule();
6     public String getConstructeur();
7     public Date getMiseEnCirculation();
8     public double getKiloMetrage();
9 }

J Informations.java
1 import java.util.Date;
2
3 public class Voiture implements Vehicule {
4     private double vitesseMax;
5     private String immatricule;
6     private String constructeur;
7     private Date miseEnCirculation;
8     private double kiloMetrage;
9
10    public Voiture(String im, String c, Date d, double km) {
11        this.vitesseMax = 130;
12        this.immatricule = im;
13        this.constructeur = c;
14        this.miseEnCirculation = d;
15        this.kiloMetrage = km;
16    }
17
18    public double getVitesseMax() {
19        return this.vitesseMax;
20    }
21
22    public String getImmatricule() {
23        return this.immatricule;
24    }
25
26    public String getConstructeur() {
27        return this.constructeur;
28    }
29
30    public double getKiloMetrage() {
31        return this.kiloMetrage;
32    }
33 }

J Main.java G2
1 import java.util.Date;
2
3 public class Main {
4     public static void main(String[] args) {
5         Date date = new Date(2015, 12, 31);
6         Vehicule v = new Voiture(
7             "AA-123-BB", "Peugeot", date, 45015
8         );
9         System.out.println(v);
10    }
11 }

```

Rien: le programme ne compile car un objet Voiture ne peut être instancié dans une référence qui est une interface

Voiture@8bcc55f

Le programme affiche une erreur à l'exécution car on a oublié de redéfinir la méthode toString()

Rien: le programme ne compile pas car Voiture n'est pas déclarée comme une classe abstraite et doit donc redéfinir toutes les méthodes de Vehicule

Voiture: AA-123-
BB, Peugeot,
31/12/2015, 45015