

✓ Terminé : Recevoir une note

Description

Remise

Modifier

Vue de la remise

Liste des remises antérieures

Fichiers requis: main/Main.java, univ/Personne.java (Télécharger)

Nombre maximal de fichiers: 20

Type de travail: Travail individuel

Personne

Une personne a un nom, un prénom et un genre (homme ou femme).

Classe Personne

Attributs

Une classe Personne qui modélise une personne doit encapsuler ces données

- nom et prenom de type String et
- homme de type boolean

(en respectant exactement cette orthographe).

Constructeur

Le constructeur de cette classe doit initialiser tous les attributs.

Getters

Les attributs sont encapsulés. Ils doivent être accessibles par des getters

- `getPrenom()`
- `getNom()`
- `isHomme()`

Présentation

Une personne aime bien se présenter. Elle le fait de deux manières, soit familière soit soutenue.

- La façon soutenue de se présenter d'une personne consiste à dire "M." ou "Mme ", selon son genre, suivi de son nom puis son prénom.
- La façon familière de se présenter d'une personne consiste à dire simplement son prénom puis son nom.

Ce sera le rôle de la méthode `String presentation(boolean soutenue)`. Dans cette méthode, pour accéder aux attributs, il est conseillé d'utiliser les getters.

Discussion

Une personne est polie. Quand elle rencontre une autre personne, elle dit bonjour et se présente. Il s'ensuit un petit dialogue, du style :

```
<< Bonjour, je suis M.Luke Lucky >> dit Lucky Luke.  
<< Bonjour M.Luke Lucky.  
    Moi c'est Mme Dalton Ma >> répond Ma Dalton.  
<< Comment allez-vous ? >> continue Ma Dalton.  
<< Ca va bien. Merci. >> termine Lucky Luke.
```

Ce dialogue est établi à l'aide des quatre méthodes suivantes :

Résumé de l'état du travail

Remises: 13

Dernière soumission:

Remis le mercredi 19 novembre 2025, 09:11 (Télécharger)

Relu le mercredi 19 novembre 2025, 09:11 par Note automatique

Note: 20,00 / 20,00

Détails

?

1. `public final void ditBonjourA(Personne personne)`
2. `private void repondAuBonjourDe(Personne personne)`
3. `private void demandeSiCaVaA(Personne personne)`
4. `private void demandeEtVousA(Personne personne)`

ELLES N'ONT PAS À ÊTRE MODIFIÉES EN QUOIQU'ELLE CE SOIT.

Les 4 précédentes méthodes invoquent 4 autres méthodes dont vous aurez à écrire le corps.

1. `protected String bonjour()` qui doit retourner `"Bonjour, je suis "` suivi de sa présentation soutenue.
Exemple : `"Bonjour, je suis M.Luke Lucky"`

2. `protected String reponseAuBonjourDe(Personne personne)` d'une personne qui retourne
 - `"Bonjour "`
 - suivi de la présentation soutenue de personne (du paramètre `personne`)
 - suivi d'un saut de ligne
 - suivi de `"Moi c'est "`
 - suivi de la présentation soutenue de cette personne.

Exemple

```
"Bonjour M.Luke Lucky.
Moi c'est Mme Dalton Ma"
```

3. `protected String caVa(Personne personne)` retourne `"Comment allez-vous ?"`
4. `protected String etVous(Personne personne)` retourne `"Ca va bien. Merci"`

L'exécution de la classe autonome `main.Main` devrait engendrer le petit dialogue entre Lucky et Ma Dalton déjà évoqué.

Fichiers requis

main/Main.java

```
1  /*
2  * @author : Yvan Maillot (yvan.maillot@uha.fr)
3  */
4  package main;
5
6  import java.io.IOException;
7
8  // 0. Regler les import si necessaire.
9
10 public class Main {
11
12     // Programme pour tester vos classes.
13     //
14     // 1. Enlever les lignes commentaires /* 1 et 1 */ quand vous aurez
15     // écrit le constructeur de la classe Personne. Bien sûr, ça ne suffit
16     // pas. Mais vous pourrez exécuter le programme et voir ce qu'il reste
17     // à faire.
18     //
19     public static void main(String[] args) throws IOException {
20         System.out.println("\n--- Quelque part dans le farwest ---- \n");
21         /* 1
22            Personne lucky = new Personne("Luke", "Lucky", true);
23            Personne ma = new Personne("Dalton", "Ma", false);
24
25            lucky.ditBonjourA(ma); // Ce simple salut doit entraîner la discussion suivante :
26            1 */
27         /*
28         << Bonjour, je suis M.Luke Lucky >> dit Lucky Luke.
29         << Bonjour M.Luke Lucky.
30         Moi c'est Mme Dalton Ma >> répond Ma Dalton.
31         << Comment allez-vous ? >> continue Ma Dalton.
32         << Ca va bien. Merci. >> termine Lucky Luke.          */
33     }
34 }
35
```

univ/Personne.java

```

1  /*
2   * @author : Yvan Maillot (yvan.maillot@uha.fr)
3   */
4   package univ;
5
6   public class Personne {
7       // TODO 1.01. Déclarer nom, prenom et homme (lettre pour lettre)
8       //
9
10      /**
11       * TODO 1.02
12       * Créer une personne en définissant son nom, son prenom et s'il est
13       * un homme ou une femme
14       * @param nom une String
15       * @param prenom une String
16       * @param homme un boolean vrai masculin, faux féminin
17       */
18      // TODO 1.02. Le constructeur
19      //
20
21      // TODO 1.03. Ne pas oublier les getters des attributs
22      //
23
24      /**
25       * TODO 1.04
26       * Retourne une chaîne de caractères qui constitue la présentation soutenue
27       * ou familière de cette personne.
28       *
29       * La présentation familière est tout simplement le prénom suivi du nom
30       * Par exemple, "Lucky Luke"
31       *
32       * La présentation soutenue est M. ou Mme (en fonction de isHomme()) suivi
33       * du nom et du prénom
34       * Par exemple, M. Luke Lucky
35       *
36       * ATTENTION : utilisez NÉCESSAIREMENT les getters et NON les attributs.
37       *
38       * @param soutenue pour déterminer le type de présentation
39       *
40       * @return la chaîne de présentation
41       */
42      protected String presentation(boolean soutenue) {
43          return null;
44      } // TODO 1.04. La présentation soutenue ou familière (lire la javadoc)
45
46
47      /**
48       * TODO 1.05
49       * @return la présentation soutenue
50       */
51      protected String presentation() {
52          return null;
53      } // TODO 1.05. Par défaut, la présentation est soutenue
54
55
56      /**
57       * TODO 1.06 String bonjour()
58       * @return "Bonjour, je suis " suivi de la présentation soutenue de cette personne
59       */
60      protected String bonjour() {
61          return null;
62      } // TODO 1.06. Lire la javadoc
63
64
65      /**
66       * TODO 1.07 String reponseAuBonjourDe(Personne personne)
67       * @param personne la personne à qui répond cette personne.
68       * @return "Bonjour "
69       *         suivi de la présentation soutenue de personne (du paramètre personne)
70       *         suivi d'un saut de ligne
71       *         suivi de "Moi c'est "
72       *         suivi de la présentation soutenue de cette personne.
73       *
74       *         Exemple : "Bonjour M.Luke Lucky.
75       *                   Moi c'est Mme Dalton Ma"
76       */
77      protected String reponseAuBonjourDe(Personne personne) {
78          return null;
79      } // TODO 1.07. Lire la javadoc
80
81
82      /**
83       * TODO 1.08 String caVa(Personne personne)
84       * @param personne la personne à s'adresser (ignorée ici).
85       * @return "Comment allez-vous ?"
86       */
87      protected String caVa(Personne personne) {
88          return null;
89      } // TODO 1.08. Comment allez-vous
90
91
92      /**
93       * TODO 1.09 String etVous(Personne personne)
94       * @param personne la personne à qui s'adresser (ignorée ici).
95       * @return "Ca va bien. Merci."
96       */
97      protected String etVous(Personne personne) {
98          return null;
99      } // TODO 1.09. Ca va bien. Merci.
100
101
102      /**
103       * Affiche sur la sortie standard "<< **B** >> dit **P**.\n" où
104       * 1. **B** est le résultat de l'invocation de bonjour()
105       * 2. **P** est le résultat de l'invocation de presentation(false)

```

?



[Webservice](#)

[VPL 4.2.4](#)