

NADÉ
Alexandre
D2

TP 5
Pendou
05/10/2022

Université de Bretagne Sud
IUT de Vannes
BUT Info 1D

Exercice P1)

Écrire la méthode `creerDico()` qui crée un dictionnaire.

Code

```
1.  final String[] listLanguages = {"PYTHON", "JAVA", "RUST", "GO LANG", "JAVASCRIPT", "SWIFT"};
2.
3.  /**
4.   * Crée un dictionnaire de mots
5.   * @return le dictionnaire
6.   */
7.  String[] creerDico() {
8.      String[] dico = listLanguages;
9.      return dico;
10. }
11.
12. /**
13.  * teste la méthode creerDico
14.  */
15. void testCreerDico() {
16.     System.out.println();
17.     System.out.println("*** testCreerDico()");
18.     testCasCreerDico(listLanguages);
19. }
20.
21. /**
22.  * teste un appel de creerDico
23.  * @param dico dictionnaire attendu
24.  */
25. void testCasCreerDico(String[] dico) {
26.     System.out.print("creerDico() \t= ");
27.
28.     if (Arrays.equals(creerDico(), dico)) {
29.         System.out.println("OK");
30.     } else {
31.         System.err.println("ERREUR");
32.     }
33. }
```

Réponse

```
*** testCreerDico()
creerDico()      = OK
```

Exercice P2)

Écrire une méthode choisirMot() qui rend un des mots pris au hasard dans le dictionnaire passé en paramètre.

Code

```
1.  /**
2.   * Choisit un mot au hasard dans le dictionnaire
3.   * @param dico le dictionnaire de mots
4.   */
5.  String choisirMot(String[] dico) {
6.      int random = (int) (Math.random() * creerDico().length);
7.      return dico[random];
8.  }
9.
10. /**
11.  * teste la méthode choisirMot
12.  */
13. void testChoisirMot() {
14.     System.out.println();
15.     System.out.println("*** testChoisirMot()");
16.     testCasChoisirMot(creerDico());
17. }
18.
19. /**
20.  * teste un appel de choisirMot
21.  * @param dico dictionnaire attendu
22.  */
23. void testCasChoisirMot(String[] dico) {
24.     System.out.print("choisirMot() \t= ");
25.
26.     if (choisirMot(dico) != null) {
27.         System.out.println("OK");
28.     } else {
29.         System.err.println("ERREUR");
30.     }
31. }
32.
```

Réponse

```
*** testChoisirMot()
choisirMot()      = OK
```

Exercice P3)

Soit la méthode afficherReponse() qui affiche sur une ligne la réponse de l'utilisateur passée en paramètre. Chaque lettre est séparée par un espace.

Code

```
1.  /**
2.   * Affiche le mot avec les lettres trouvées et les lettres non trouvées remplacées par des
   tirets
3.   * @param reponse le mot à afficher
4.   */
5.  void afficherReponse(char[] reponse) {
6.      for (int i = 0; i < reponse.length; i++) {
7.          System.out.print(reponse[i]);
8.      }
9.      System.out.println();
10. }
11.
```

```

12.    /**
13.     * teste la méthode afficherReponse
14.     */
15.    void testAffichierReponse() {
16.        System.out.println();
17.        System.out.println("*** testAffichierReponse()");
18.        char[] reponse1 = {'P', 'R', 'O', 'G', 'R', 'A', 'M', 'M', 'E'};
19.        testCasAfficherReponse (reponse1);
20.        char[] reponse2 = {};
21.        testCasAfficherReponse (reponse2);
22.    }
23.
24.    /**
25.     * teste un appel à afficherReponse()
26.     * @param reponse tableau des réponse à afficher
27.     */
28.    void testCasAfficherReponse (char[] reponse) {
29.        System.out.print ("afficherReponse (" + Arrays.toString(reponse) + ") : ");
30.        afficherReponse (reponse);
31.    }
32.

```

Réponse

```

*** testAffichierReponse()
afficherReponse ([P, R, O, G, R, A, M, M, E]) : PROGRAMME
afficherReponse ([]) :

```

Exercice P4)

Écrire une méthode `creerReponse()` qui crée un tableau pour la réponse contenant uniquement des `_`.

Code

```

1.    /**
2.     * Affiche le mot a avec des '_'
3.     * @param lg la longueur du mot
4.     * @return le mot avec des '_'
5.     */
6.    char[] creerReponse(int lg) {
7.        char[] reponse = new char[lg];
8.        for (int i = 0; i < reponse.length; i++) {
9.            reponse[i] = '_';
10.        }
11.        return reponse;
12.    }
13.
14.    /**
15.     * teste la méthode creerReponse
16.     */
17.    void testCreerReponse() {
18.        System.out.println();
19.        System.out.println("*** testCreerReponse()");
20.        testCasCreerReponse(8, new char[] {'_', '_', '_', '_', '_', '_', '_', '_'});
21.        testCasCreerReponse(7, new char[] {'_', '_', '_', '_', '_', '_', '_'});
22.        testCasCreerReponse(6, new char[] {'_', '_', '_', '_', '_', '_'});
23.    }
24.
25.    /** teste un appel à creerReponse()
26.     * @param lg la longueur du mot
27.     * @param reponse tableau des réponse à afficher
28.     */
29.    void testCasCreerReponse(int lg, char[] reponse) {
30.        System.out.print("creerReponse(" + lg + ") \t= ");
31.
32.        if (Arrays.equals(creerReponse(lg), reponse)) {

```

```

33.         System.out.println("OK");
34.     } else {
35.         System.err.println("ERREUR");
36.     }
37. }

```

Réponse

```

*** testCreerReponse()
creerReponse(8)      = OK
creerReponse(7)      = OK
creerReponse(6)      = OK

```

Exercice P5)

Écrire une méthode tester() qui prend en paramètre le mot à deviner, le tableau des réponses de l'utilisateur, le caractère saisi par l'utilisateur et qui rend vrai si la lettre est présente dans le mot, faux sinon. Enfin, la méthode remplace dans le tableau reponse le signe _ par le caractère testé.

Code

```

1.  /**
2.   * Teste si la lettre est dans le mot
3.   * @param mot le mot
4.   * @param reponse le mot avec les lettres trouvées
5.   * @param car le caractère à tester
6.   * @return vrai ssi la lettre est dans le mot
7.   */
8.  boolean tester (String mot, char[] reponse, char car) {
9.      boolean result = false;
10.     for (int i = 0; i < mot.length(); i++) {
11.         if (mot.charAt(i) == car) {
12.             reponse[i] = car;
13.             result = true;
14.         }
15.     }
16.     return result;
17. }
18.
19. /**
20.  * teste la méthode tester
21.  */
22. void testTester () {
23.     System.out.println();
24.     System.out.println("*** testTester()");
25.     testCasTester("PYTHON", creerReponse(6), 'Y', true);
26.     testCasTester("JAVA", creerReponse(4), 'B', false);
27.     testCasTester("PYTHON", creerReponse(4), ' ', false);
28. }
29.
30. /**
31.  * teste un appel de tester
32.  * @param mot mot à tester
33.  * @param reponse réponse attendue
34.  * @param car caractère à tester
35.  * @param result résultat attendu
36.  */
37. void testCasTester (String mot, char[] reponse, char car, boolean result) {
38.     System.out.print("tester(" + mot + ", " + reponse + ", " + car + ") \t= ");
39.
40.     if (tester(mot, reponse, car) == result) {
41.         System.out.println("OK");
42.     } else {
43.         System.err.println("ERREUR");
44.     }
45. }
46.

```

Réponse

```
*** testTester()
tester(PYTHON, [C@449b2d27, Y) = OK
tester(JAVA, [C@5479e3f, B) = OK
tester(PYTHON, [C@27082746, ) = OK
```

Exercice P6)

Écrire la méthode `estComplet()` qui rend vrai si la réponse de l'utilisateur correspond lettre à lettre au mot cherché, faux sinon.

Code

```
1.  /**
2.   * estComplet
3.   * @param mot le mot
4.   * @param reponse le mot avec les lettres trouvées
5.   * @return vrai ssi le mot est complet
6.   */
7.  boolean estComplet (String mot, char[] reponse) {
8.      boolean result = true;
9.      for (int i = 0; i < mot.length(); i++) {
10.         if (mot.charAt(i) != reponse[i]) {
11.             result = false;
12.         }
13.     }
14.     return result;
15. }
16.
17. /**
18.  * teste la méthode estComplet
19.  */
20. void testEstComplet () {
21.     System.out.println();
22.     System.out.println("*** testEstComplet()");
23.     testCasEstComplet("PYTHON", new char[] {'P', 'Y', 'T', 'H', 'O', 'N'}, true);
24.     testCasEstComplet("JAVA", new char[] {'J', 'A', 'V', 'A'}, true);
25.     testCasEstComplet("PYTHON", new char[] {'P', 'Y', 'T', 'H', 'O', '_'}, false);
26.     testCasEstComplet("PYTHON", new char[] {'_', '_', '_', '_', '_', '_'}, false);
27. }
28.
29. /**
30.  * teste un appel de estComplet
31.  * @param mot mot à tester
32.  * @param reponse réponse attendue
33.  * @param result résultat attendu
34.  */
35. void testCasEstComplet (String mot, char[] reponse, boolean result) {
36.     System.out.print("estComplet(" + mot + ", ");
37.     afficherTabChar(reponse);
38.     System.out.println(") \t= ");
39.
40.     if (estComplet(mot, reponse) == result) {
41.         System.out.println("OK");
42.     } else {
43.         System.err.println("ERREUR");
44.     }
45. }
46.
```

Réponse

```
*** testEstComplet()
estComplet(PYTHON, [P, Y, T, H, O, N]) =
OK
estComplet(JAVA, [J, A, V, A]) =
OK
estComplet(PYTHON, [P, Y, T, H, O, _]) =
OK
estComplet(PYTHON, [_ , _ , _ , _ , _ , _]) =
OK
```

Exercice P7)

Écrire une méthode partie() qui permet de jouer une partie. Le dictionnaire est passé en paramètre.

Code

```
1. void partie(String[] dico) {
2.     String mot = choisirMot(dico);
3.     char[] reponse = creerReponse(mot.length());
4.     afficherReponse(reponse);
5.     int nbEssais = 9;
6.     while (nbEssais > 0 && !estComplet(mot, reponse)) {
7.         char car = SimpleInput.getChar("Proposez une lettre : ");
8.         if (tester(mot, reponse, car)) {
9.             System.out.println("Bravo !");
10.        } else {
11.            nbEssais = nbEssais - 1;
12.            System.out.println("Dommage ! Il reste " + nbEssais + " essais restants");
13.        }
14.        afficherReponse(reponse);
15.    }
16.    if (estComplet(mot, reponse)) {
17.        System.out.println("Bravo, vous avez gagné !");
18.    } else {
19.        System.out.println("Dommage, vous avez perdu !");
20.    }
21. }
22.
```

Exercice P8)

Écrire la méthode principal() du jeu.

Code

```
1. void principal() {
2.     // testCreerDico();
3.     // testChoisirMot();
4.     // testAffichierReponse();
5.     // testCreerReponse();
6.     // testTester();
7.     // testEstComplet();
8.
9.     partie(creerDico());
10. }
11.
```

Réponse

Proposez une lettre : J
Dommage ! Il reste 8 essais restants

Proposez une lettre : A
Dommage ! Il reste 7 essais restants

Proposez une lettre : P
Dommage ! Il reste 6 essais restants

Proposez une lettre : R
Bravo !

R____
Proposez une lettre : U
Bravo !

RU____
Proposez une lettre : U
Bravo !

RU____
Proposez une lettre : S
Bravo !

RUS____
Proposez une lettre : T
Bravo !

RUST____
Bravo, vous avez gagné !

Proposez une lettre : J
Bravo !

J_____
Proposez une lettre : A
Bravo !

JA_A_____
Proposez une lettre : V
Bravo !

JAVA_____
Proposez une lettre : P
Bravo !

JAVA____P____
Proposez une lettre : P
Bravo !

JAVA____P____
Proposez une lettre : X
Dommage ! Il reste 8 essais restants

JAVA____P____
Proposez une lettre : X
Dommage ! Il reste 7 essais restants

JAVA____P____
Proposez une lettre : X
Dommage ! Il reste 6 essais restants

JAVA____P____
Proposez une lettre : X
Dommage ! Il reste 5 essais restants

JAVA____P____
Proposez une lettre : X
Dommage ! Il reste 4 essais restants

JAVA____P____
Proposez une lettre : X
Dommage ! Il reste 3 essais restants

JAVA____P____
Proposez une lettre : X
Dommage ! Il reste 2 essais restants

JAVA____P____
Proposez une lettre : X
Dommage ! Il reste 1 essai restant

JAVA____P____
Proposez une lettre : X
Bravo, vous avez gagné !

Proposez une lettre : X
Dommage ! Il reste 3 essais restants
JAVA__P_
Proposez une lettre : X
Dommage ! Il reste 2 essais restants
JAVA__P_
Proposez une lettre : X
Dommage ! Il reste 1 essais restants
JAVA__P_
Proposez une lettre : X
Dommage ! Il reste 0 essais restants
JAVA__P_
Dommage, vous avez perdu !