B1_BDP_M2_20231123_correction

Partie I

On rappelle que la **sortie** d'un programme consiste en ce qui va s'afficher à l'écran lors de son exécution *et rien d'autre*. **Donnez la sortie** du programme suivant :

JAVA

```java
int nbEntier;
nbEntier = 1;
String s = "hello";
int nb1 = 5;

if (nbEntier > 2) {
  System.out.print("ici");
  nbEntier = nbEntier + 2;
  System.out.println(nbEntier);
}

if (nbEntier <= 2) {
  System.out.print("là");
  nbEntier = nbEntier + 5;
  System.out.println(nbEntier);
} else {
  System.out.println("encore");
  nbEntier--;
  System.out.println(nbEntier);
}
System.out.println(nbEntier);
int nb2 = 0;
for (int i = 1; i <= nb1; i++) {
  System.out.println(s + i + " : " + nb2);
  nb2 = nb2 + 10;
}
System.out.println(nb2);
```

CORRIGÉ

Le premier bloc `if` n'est pas exécuté car la condition est fausse. La branche VRAI du second bloc `if` est exécutée car la condition est vraie. La boucle `for` est exécutée 5 fois. À chaque itération, on affiche un texte du style `hello 1 : 0` où les deux valeurs numériques sont modifiées à chaque itération (`+1` pour `i` et `+10` pour `nb2`). Attention à la différence entre `print` et `println` .

La sortie est la suivante :

```
là6
6
hello1 : 0
hello2 : 10
hello3 : 20
hello4 : 30
hello5 : 40
50
```

Partie II

On considère la version 3 du *Jeu de l'oie* vue en cours et dont une implémentation correcte est reproduite ci-dessous :

JAVA

```java
int caseObjectif = 20;
Random generateur = new Random();
int caseCourante = 0;
int compteurLancers = 0;
boolean gagne = false;

while (!gagne) {
  int lancer = generateur.nextInt(6) + 1;
  compteurLancers++;
  caseCourante = caseCourante + lancer;
  if (caseCourante > caseObjectif) {
    int depassement = caseCourante - caseObjectif;
    caseCourante = caseObjectif - depassement;
  }
  System.out.println(String.format("Lancer %d : vous avez fait %d. Vous êtes sur la case %d.",
      compteurLancers, lancer, caseCourante));
  if (caseCourante == caseObjectif) {
    System.out.println("Vous avez gagné !");
    gagne = true;
  } else {
    System.out.println("Il vous reste " + (caseObjectif - caseCourante) + " cases.");
  }
}
```

---

01 Expliquez la **différence entre les opérateurs** `=` **et** `==`. Vous pouvez vous appuyer sur le code pour illustrer votre propos.

CORRIGÉ: `=` est l'opérateur d'affectation. Il permet d'assigner une valeur (à droite) à une variable (à gauche). Par exemple, ici, `caseCourante = caseCourante + lancer` signifie qu'on va calculer la valeur de `caseCourante + lancer` et l'affecter comme nouvelle valeur à la variable `caseCourante`.

```
`==` est l'opérateur de comparaison. Il permet de tester l'égalité entre deux valeurs. Par exemple, ici,
`caseCourante == caseObjectif` signifie qu'on va tester si la valeur de `caseCourante` est égale à la valeur de
`caseObjectif`. L'expression sera évaluée à `true` si les deux valeurs sont égales, et à `false` sinon.
```

---

02 Expliquez le rôle de la variable `gagne` **en vous appuyant sur toutes ses apparitions** dans le programme.

CORRIGÉ: la variable `gagne` est utilisée pour savoir si le joueur a gagné ou non. Elle est initialisée à `false` au début du programme. Elle est mise à `true` à la ligne 19 si le joueur a gagné. Elle est testée à la ligne 7 dans le TANT QUE pour savoir si on continue ou non la partie (TANT QUE on n'a pas gagné, on continue).

On vous demande d'implémenter de **nouveaux besoins** pour ce programme. Vous considérerez que chaque question s'appuie sur la version de base ci-dessus (indépendamment des autres questions). Pour chaque implémentation d'un besoin, vous ne réécrirez pas l'ensemble du programme: vous n'écrirez que ce qui est modifié/ajouté, en indiquant à chaque fois les numéros de lignes de code concernées (par exemple: «*modification ligne 17: ...*» ou bien «*ajout entre 16 et 17: ...*»). Les besoins demandés sont exprimés dans les questions qui suivent.

---

03 L'utilisateur indique en début de programme sur quelle case il veut démarrer la partie.

CORRIGÉ: On remplace la ligne 3 par:

JAVA

```java
Scanner clavier = new Scanner(System.in);
System.out.print("Sur quelle case voulez-vous démarrer ? ");
int caseCourante = clavier.nextInt();
```

04 On lance à chaque fois **deux dés** au lieu d'un seul pour savoir de combien on avance.

CORRIG€: On remplace la ligne 8 par:

JAVA

```
int lancer1 = generateur.nextInt(6) + 1;
int lancer2 = generateur.nextInt(6) + 1;
int lancer = lancer1 + lancer2;
```

---

05 On veut maintenant pouvoir jouer à **deux joueurs**, chacun son tour. Le premier joueur à atteindre la case objectif a gagné.

CORRIGÉ: Il faut pouvoir suivre les pions des 2 joueurs (`caseCouranteJ1` et `caseCouranteJ2`), et savoir qui est en train de jouer (`joueurCourant` vaudra alternativement 1 et 2).

On remplace la ligne 3 par:

JAVA

```
int caseCouranteJ1 = 0;
int caseCouranteJ2 = 0;
int joueurCourant = 1;   // 1 ou 2
```

Le contenu du TANT QUE doit alors prendre en compte qui joue. On remplace le contenu du bloc TQ (lignes 8 à 22) par:

JAVA

```
if (joueurCourant == 1) {
  System.out.println("Joueur 1, à vous de jouer !");
  caseCouranteJ1 = caseCouranteJ1 + lancer;
  if (caseCouranteJ1 > caseObjectif) {
    int depassement = caseCouranteJ1 - caseObjectif;
    caseCouranteJ1 = caseObjectif - depassement;
  }
  System.out.println(String.format("Vous avez fait %d. Vous êtes sur la case %d.",
      lancer, caseCouranteJ1));
  if (caseCouranteJ1 == caseObjectif) {
    System.out.println("Vous avez gagné !");
    gagne = true;
  }
  joueurCourant = 2;
} else {
  System.out.println("Joueur 2, à vous de jouer !");
  // Exactement la même chose que précédemment, mais en remplaçant caseCouranteJ1 par caseCouranteJ2
  joueurCourant = 1;
}
```

---

06 Un score est attribué à chaque partie. Ce score est déterminé par **le nombre total de cases parcourues** lors de la partie (y compris quand on «rebondit» sur la case 20). Le score de la partie doit être affiché à la fin.

CORRIGÉ: Il faut comprendre que le score correspond simplement au total fait par les dés. Il faut donc une variable pour retenir le score au fur et à mesure (même principe que la variable `compteurLancers`).

Ajout entre 04 et 05: `int score = 0;`

Ajout entre 08 et 09: `score = score + lancer; // on ajoute le résultat de chaque lancer`

Remplacement ligne 18: `System.out.println("Vous avez gagné ! Votre score : " + score);`

Exceptionnellement, le besoin suivant s'appuie sur le précédent dont on supposera qu'il a été correctement implémenté. On veut maintenant effectuer un calcul statistique pour estimer le **nombre moyen de cases à parcourir pour gagner** (c'est-à-dire le score moyen). Pour cela, on va lancer 10 000 simulations et calculer la moyenne des scores. On rappelle que la moyenne sera obtenue en divisant la somme des scores par le nombre de simulations effectuées. La moyenne sera affichée à la fin du programme.

07 Faites une **analyse** très succincte de ce nouveau besoin. Tranchez sur les éventuels **choix d'implémentation** qui s'imposent.

La partie principale du programme (lignes 7 à 23) doit être répétée. Il nous faut donc une structure répétitive (boucle). On connaît le nombre exact d'itérations (de répétitions). On peut donc encadrer cette partie par une boucle `for` qui va s'exécuter 10 000 fois. Il nous faudra aussi une variable `scoreTotal` pour retenir la somme des scores. Finalement, il faudra attendre la sortie de boucle pour calculer et afficher la moyenne.

---

08 Implémentez ce nouveau besoin.

JAVA

```java
// Initialisation de variables supplémentaires
int score = 0;      // score de la partie en cours
int scoreTotal = 0; // score total des 10 000 parties
// Encadrement de la partie principale par une boucle for
for (int i = 1; i <= 10000; i++) {
  scoreTotal = scoreTotal + score; // On accumule le score de chaque partie
  score = 0;                       // On réinitialise ensuite le score pour la partie suivante
  // le reste : bloc while (ligne 7 à 23) est inchangé
}
// En sortie de boucle, on calcule et affiche la moyenne
double moyenneScores = scoreTotal / 10000.0;
System.out.println("Moyenne des scores : " + moyennesScores);
```

Annexe - Documentation

JAVA

```java
// Afficher « Bonjour » à l'écran puis passer à la ligne
System.out.println("Bonjour");

// Afficher « Bonjour » à l'écran sans passer à la ligne
System.out.print("Bonjour");

// Récupérer une entrée utilisateur (de type entier) au clavier
Scanner clavier = new Scanner(System.in);
int entree = clavier.nextInt();

// Générer un entier aléatoire entre 0 et 99
Random generateur = new Random();
int auHasard = generateur.nextInt(100);

// Formater une chaîne de caractères - %d représente un entier
// Chaque %d sera remplacé par la valeur de la variable correspondante dans la liste qui suit la chaîne
String.format("J'ai %d pommes et %d poires.", 6, 3); // produira la chaîne : "J'ai 6 pommes et 3 poires."
```

Last updated 2023-11-30 18:05:56 +0100